R. Ramanujam
Srini Ramaswamy (Eds.)

LNCS 7154

Distributed Computing and Internet Technology

8th International Conference, ICDCIT 2012
Bhubaneswar, India, February 2012
Proceedings

Springer

Lecture Notes in Computer Science 7154

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

R. Ramanujam   Srini Ramaswamy (Eds.)

Distributed Computing and Internet Technology

8th International Conference, ICDCIT 2012
Bhubaneswar, India, February 2-4, 2012
Proceedings

Springer

Volume Editors

R. Ramanujam
Institute of Mathematical Sciences
C.I.T. Campus
Taramani, Chennai 600113, India
E-mail: jam@imsc.res.in

Srini Ramaswamy
Industrial Software Systems
ABB India Corporate Research Center
Bangalore, India
Email: srini@ieee.org

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-28072-6 e-ISBN 978-3-642-28073-3
DOI 10.1007/978-3-642-28073-3
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011945543

CR Subject Classification (1998): C.2.4, C.2, D.4, H.2.4, E.3, H.3-5, K.6.5

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains the papers presented at ICDCIT 2012: The 8th International Conference on Distributed Computing and Internet Technologies held during February 2–4, 2012 at KIIT University, Bhubaneswar, India. The papers range over a spectrum of issues related to the theme of the conference, covering theoretical foundations, computational tools, and societal applications. State-of-the-art techniques like game theoretic ones are used by authors to analyze conceptual problems.

As in the previous years, we were fortunate to have highly eminent researchers giving plenary talks. It gives us great pleasure to thank N. Asokan (Nokia Research Centre, Helsinki, Finland), Catherine Meadows (Naval Research laboratory, Washington DC, USA), Yoram Moses (Technion, Haifa, Israel) and Lynne Parker (University of Tennessee, USA) for agreeing to give talks and for contributing to this volume.

There were 89 submissions. Each submission was reviewed by at least 2, and on the average 2.7, Program Committee members or their subreviewers. The committee decided to accept 17 regular papers and 15 submissions for short presentation (for some of which a short abstract is included in the proceedings). We thank all the reviewers for their invaluable help. We also express our gratitude to all members of the Program Committee for doing an excellent job of discussion and selection.

As is customary in ICDCIT, there were satellite events of interest to the community: an Industry Syposium, a Research Student Symposium and a Student Poster Exhibition.

The conference was co-organized by KIIT University, Bhubaneswar, and was held on their pleasant campus. We thank the Organizing Committee, KIIT University and its founder Achyuta Samanta for taking on the responsibility and doing a fine job.

We thank Springer for their continued support of ICDCIT and the EasyChair system for making the Program Committee's job as well as that of the proceedings editors easy indeed.

February 2012

R. Ramanujam
Srini Ramaswamy

Organization

Program Committee

Shivali Agarwal	IBM, India Research Lab, India
Amy Apon	University of Arkansas, USA
Kai Chen	Google Inc., USA
Venkatesh Choppella	International Institute of Information Technology Hyderabad, India
Meenakshi D'Souza	International Institute of Information Technology, India
Anupam Datta	CMU, USA
Elsa Estevez	United Nations University – International Institute for Software Technology, Japan
Veena Goswami	KIIT University, India
Maurice Herlihy	Brown University, USA
Prasad Jayanti	Dartmouth College, USA
S.N. Krishna	IIT Bombay, India
Rupak Majumdar	UCLA, USA
Tulika Mitra	National University of Singapore
Neeraj Mittal	The University of Texas at Dallas, USA
Hrushikesha Mohanty	University of Hyderabad, India
G.B. Mund	KIIT University, India
R. Ramanujam	Institute of Mathematical Sciences, Chennai (Co-chair), India
Srini Ramaswamy	Industrial Software Systems, ABB Corporate Research (Co-chair), India
Pradip Srimani	Clemson University, USA
Maria Wimmer	Universität Koblenz-Landau, Germany
Nobuko Yoshida	Imperial College London, UK

Additional Reviewers

Al-Shukri, Shaymaa
Anguraj, Baskar
Anil, R.
Arumugam, Sivabalan
Barjis, Joseph
Bhattacharya, Debojyoti
Bhavani S., Durga
Chebrolu, Kameswari
Chenthati, Deepak
D'Souza, Deepak
Damani, Om
Das, Debabrata
Datta, Subhajit
Fossati, Luca
Gabale, Vijay
Garg, Vikas K.

Ghosh, Sanjay
Hota, Chittaranjan
Hu, Raymond
Hurst, William
Jetley, Raoul
Jithendrian, S.
Joshi, Hemant
Kanaskar, Nitin
Karunakaran, Deepak
Kulkarni, Purushottam
Kulkarni, Sandeep
Kumar, Atul
Lal, Rajendra Prasad
Lodaya, Kamal
Manoj, Kranthi
Martha, Venkata Swamy
Mohanty, Hrushikesha
Muralidhara, V.N.
Narasimhan, Lakshmi
Nelabhotla, Narendra Kumar
Ngo, Linh
Paul, Soumya
Ponnalagu, Karthikeyan
Raghavan, Rama
Ramanathan, Chandrashekar
Rao, Shrisha
Rath, Anand
Rathinasamy, Bhavanandan
Sahoo, Anirudha
Sapna, P.G.
Sarac, Kamil
Sen, Rijurekha
Shanthi, V.
Sharma, Divyasheel
Sivakumar, G.
Sobha Rani, T.
Sreevalsan Nair, Jaya
Srinivasa, Srinath
Subramanian, Vimalathithan
Sudarsan, Sithu
Suresh, S.P.
Tudoreanu, M. Eduard
Vaidyanathan, Aparajithan
Vu, Hai
Yadav, Dharmendra Kumar
Yoshigoe, Kenji
Yu, Liguo

Table of Contents

Usable Mobile Security

N. Asokan and Cynthia Kuo

Nokia Research Center
{n.asokan,cynthia.kuo}@nokia.com

Abstract. We make the case for usable mobile security by outlining why usable security in mobile devices is important and why it is hard to achieve. We describe a number of current problems in mobile devices that need usable and secure solutions. Finally, we discuss the characteristics of mobile devices that can actually help in designing usable solutions to mobile security problems.

1 Introduction

Over the last decade or so, the security research community has come to recognize the importance of simultaneously achieving usability and security goals when designing new protocols, applications and systems. Although balancing security, usability (and other constraints like deployment cost) continues to be a hard challenge, it is no longer assumed that usability and security are mutually contradictory goals[15]. In this paper, we discuss the particular challenges in designing usable and secure solutions for mobile devices. Mobile devices have displays of limited size and limited means of user interaction. Energy and power efficiency is a critical concern on mobile devices. The convenience of portability of mobile devices comes with an increased risk of loss or theft. In Section 2 we discuss some concrete problems. Often the primary motivation (from the perspective of designers) for usable security arises when lack thereof will lead to a definite cost. The source of such costs can be surprising as we see in the next section.

On the positive side, as we discuss in Section 3, mobile devices posses certain attributes that can in fact help in the design of effective security and privacy solutions.

2 Example Problems Needing Usable Mobile Security Solutions

As a first concrete example, let us consider the problem of "first connect": setting up communication and security contexts between two (or more) end-user devices. A typical user of a smartphone encounters various instances of first connect, such as pairing a wireless headset with the phone over Bluetooth or enrolling the phone to a home Wi-Fi network. During first connect, the devices store information necessary for setting up and securing future communication sessions. The security-related setup needed is a key establishment scheme.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 1–6, 2012.

By the mid-2000s, the installed base of Bluetooth- and Wi-Fi-capable devices was growing fast. Ordinary users with little technical expertise needed to perform first connect procedures like Bluetooth pairing and found it daunting. When an attempted pairing is unsuccessful, hapless users returned the devices or called the help lines of the manufacturers. In fact, around 2004, roughly one in ten Wi-Fi products sold generated a technical support call. Most calls concerned basic setup issues. In addition, up to 30% of all consumer Wi-Fi purchases were returned. The vast majority of these returned products an estimated 90% were not defective, suggesting a major problem with first connect procedures [8]. This was a source of significant cost for the manufacturers, who were motivated to design more intuitive first connect procedures.

The underlying key establishment in Wi-Fi and Bluetooth were key agreement protocols using symmetric key cryptography. Authentication of key agreement relied on the user-supplied password. Because the passwords were entered by human users, they were necessarily shorter (∼12-20 bits) than standard cryptographic keys considered acceptable (∼128 bits). As a result, the first connect key agreement protocols could be completely broken by an eavesdropper performing a dictionary attack [6]. When the standardization bodies set about redesigning the first connect protocols, their objectives were to make the first connect process more *intuitive*, the key agreement protocol *secure* enough against passive eavesdroppers as well as active man-in-the-middle attacks, as well as ensuring that the new scheme remained *inexpensive.* Several schemes aimed at meeting these objectives were developed and have been deployed. For a comparative survey see [13,2]. The main lesson from this effort is that improving the usability of an existing security procedure could not be achieved simply by tweaking the user interface but rather required fundamental changes to the underlying protocols.

There are a number of current problems in need of usable and secure solutions. We list them below. In each case, we identify the source of the cost that motivates the need for usable security and we outline currently available solutions.

Local User Authentication: Today's mobile devices are small and highly portable. The downside of this convenience is that they are more susceptible to loss or theft. The standard protection against loss or theft of mobile devices is to have a *device lock* with local user authentication: after a period of idleness the device locks itself; to unlock the user has to authenticate himself locally on the device, for example, by typing in a PIN code. Many users find it inconvenient to unlock their device several times a day. As a result, an estimated 38% to 70% of users do not lock their mobile phones and tablets[11,10,12]. This leaves any valuable data on their devices vulnerable to theft or loss.

Increasingly, people use their mobile devices to access their work e-mail or their work intranet. Such users are not allowed to turn off the device lock at their discretion since they will be risking valuable corporate data on their devices. For a business user who does not want to suffer through cumbersome local authentication, the only option is to forego the possibility of accessing work e-mail and other enterprise data on their mobile devices. This turns out to be a source of concern and cost for the enterprise system administrators because their original objective of giving their employees access to their work e-mail is undermined by the lack of usable security in local authentication.

Several alternatives for local user authentication have been proposed [3,14,4], and in some cases deployed in limited scale. None has been a clear success yet. Devising a usable, sufficiently secure and fast local authentication mechanism for mobile devices is still an open problem.

Credential Recovery: While it is important to prevent an attacker from accessing the data in a lost or stolen device, it is equally important to restore the legitimate user's ability to access the data. This is particularly important when the mobile device acts as a repository of credentials that are used to access other resources. Several initiatives have worked towards the goal of turning a personal mobile device into a *personal trusted device* which can hold the keys and credentials for a variety of services like on-line banking, payment, and even for physical access control like opening cars or doors. Many services provide credential recovery mechanisms which rely on the user having physical possession of his personal trusted device. The challenge is to devise a credential recovery mechanism that can be used when the personal trusted device itself is lost.

Mobile CAPTCHA: CAPTCHAs are commonly used by online services as a way to resist automated software robots. For example, sites may require users to solve CAPTCHAs to create new identities for an online community or to purchase tickets online for popular concerts. Without such a deterrent software robots can create a large number of accounts and abuse them, for example, to send out spam. However, CAPTCHAs on mobile devices have two problems: first, the limited user input and output capabilities of mobile devices make it much harder for users to solve CAPTCHAs on mobile devices than on personal computers; second, the history of CAPTCHA technology has been an arms race between the designers and attackers, making standard text-based CAPTCHAs increasingly difficult to solve for legitimate users. When a user trying access a service and is presented a CAPTCHA, there is a certain probability that he will drop off at that point. Anecdotal evidence from service providers suggests that the drop-off rate can be as high as 15%. In some scenarios, the use of a CAPTCHA can be avoided by using other techniques, such as hardware-based device authentication[7] with a limit on the number of accounts that can be created from the same device within a reasonable time period. But such workarounds are not applicable in all scenarios where CAPTCHAs are used.

Installation of Applications and Content: Mobile device manufacturers and platform vendors have developed various means to encourage more and more people to create applications and content for their respective device platforms. While this increases overall choice and variety, it raises the issue of malicious or inappropriate applications. How can users decide whether there are any adverse effects of installing a certain application or a certain piece of content on their devices? Some platforms, like Android, resort to asking the user to explicitly decide whether an application being installed can be granted the privileges it requests. Others resort to centralized vetting. Apple has the sole prerogative to decide which applications are made available for installation on its devices. Both approaches are unsatisfactory: the former puts the burden on the user who may not have the knowledge required to make an informed decision; whereas the latter, while removing the usability problem, restricts end consumers' choices of applications and developers' ability to incorporate certain features and architectures.

3 Mobility Can Help Security/Privacy

Finally we turn our attention to the possibility the the very same characteristics that make the data collected from mobile devices attractive to various parties (and thus open the door for possible privacy and security concerns) can in fact help in developing novel solutions for the management of security and privacy. Mobile devices today come equipped with a variety of sensors for location, movement, ambient light and sound, and a variety of radio technologies. A major challenge in deploying security and privacy frameworks is the difficulty that ordinary users face in configuring appropriate policies for themselves. The context information inferred by the various sensors of the mobile device and the history of this information can ease this problem by suggesting and possibly enforcing appropriate policies for a given user and context. As an example, consider the device lock. Users are typically asked to choose a single device unlock mechanism (e.g., entering a PIN) and a single configuration (e.g., the device will lock after five minutes of inactivity). In practice, users become irritated by having to type in a PIN several times a day. They end up either configuring very long timeouts or disabling the lock altogether. As a result, the original objective of theft protection is not met. Now consider a device that continuously monitors its environment. Over time, it can recognize places of interest where it finds itself frequently and profile those places in terms of the information it can sense, for example the set of other devices that are present. Gupta et al [5] describe how such *context profiling* can be used to select the locking mechanism and timeout that is appropriate for the context: for example, the locking timeout can be very long in a safe place like the user's home but short in a public place in the presence of many unfamiliar devices.

As a second example, consider the problem of sharing photos and videos. By default, when a user decides to share a photo he will have to manually decide who to share it with. But if a device that can sense the presence of other people present in the vicinity at the time the photo was taken, then presenting the list of such persons as possible sharing targets may be a reasonable assumption [9].

Note that both of these examples make use of exactly the type of data that caused the privacy furor over Apple's data collection on iPhones [1]. In the Apple case, the collected data was stored unencrypted and contained over 10 months of precise GPS coordinates. With the proper safeguards, such as storing coarse-grained or aggregated data in encrypted form, sensor data should be available for benign applications.

4 Conclusions

Mobile devices are in dire need of more usable security solutions. Compared to traditional PCs, mobile devices are characterized by more diverse hardware and user interfaces. Screens are smaller, and input mechanisms range from miniaturized qwerty keyboards to capacitive touch screens to basic numeric keypads. These differences make developing security mechanisms challenging. A few of the most pressing security challenges for mobile devices include local user authentication, credential recovery, CAPTCHAs, and helping users with application installation decisions. However, mobile devices also offer opportunities for security researchers that traditional PCs do not. Additional sensors and rich context information may prove useful for creating new usable security mechanisms.

References

1. Allan, A., Warden, P.: iPhone Tracking: "What Your iPhone Knows About You". O'Reilly Where 2.0 Conference (April 2011), http://where2conf.com/where2011/public/schedule/detail/20340
2. Asokan, N., Nyberg, K.: Security associations for wireless device. In: Gritzalis, S., Karygiannis, T., Skianis, C. (eds.) Security and Privacy in Mobile and Wireless Networking, pp. 23–62. Troubador Publishing Ltd., Leicester (2009), http://research.ics.tkk.fi/publications/knyberg/secass.pdf
3. Clarke, N.L., Furnell, S.: Advanced user authentication for mobile devices. Computers & Security 26(2), 109–119 (2007), http://dx.doi.org/10.1016/j.cose.2006.08.008
4. Dunphy, P., Heiner, A.P., Asokan, N.: A closer look at recognition-based graphical passwords on mobile devices. In: Cranor, L.F. (ed.) SOUPS. ACM International Conference Proceeding Series, vol. 485. ACM (2010), http://doi.acm.org/10.1145/1837110.1837114
5. Gupta, A., Miettinen, M., Asokan, N.: Using context-profiling to aid access control decisions in mobile devices. In: PerCom Workshops, pp. 310–312. IEEE (2011), http://dx.doi.org/10.1109/PERCOMW.2011.5766891
6. Jakobsson, M., Wetzel, S.: Security Weaknesses in Bluetooth. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 176–191. Springer, Heidelberg (2001)
7. Kostiainen, K., Reshetova, E., Ekberg, J.-E., Asokan, N.: Old, new, borrowed, blue –: a perspective on the evolution of mobile platform security architectures. In: Sandhu, R.S., Bertino, E. (eds.) CODASPY, pp. 13–24. ACM (2011), http://doi.acm.org/10.1145/1943513.1943517
8. Kuo, C., Goh, V., Tang, A., Perrig, A., Walker, J.: Empowering ordinary consumers to securely configure their mobile devices and wireless networks. Technical Report CMU-CyLab-05-005, Carnegie Mellon University (2005), http://repository.cmu.edu/cylab/65/
9. Miettinen, M., Asokan, N.: Towards security policy decisions based on context profiling. In: Greenstadt, R. (ed.) AISec, pp. 19–23. ACM (2010), http://doi.acm.org/10.1145/1866423.1866428
10. Norton. Norton survey reveals one in three experience cell phone loss, theft (February 8, 2011), http://www.symantec.com/about/news/release/article.jsp?prid=20110208_01

11. Retrevo Blog. iPhones, backups and toilets, what's the connection? (August 2, 2011), http://www.retrevo.com/content/blog/2011/08/iphones-backups-and-toilets-connection
12. Sophos Naked Security blog. Survey says 70% don't password-protect mobiles: download free Mobile Toolkit (August 9, 2011), http://nakedsecurity.sophos.com/2011/08/09/free-sophos-mobile-security-toolkit/
13. Suomalainen, J., Valkonen, J., Asokan, N.: Standards for security associations in personal networks: a comparative analysis. IJSN 4(1/2), 87–100 (2009)
14. van Oorschot, P.C., Thorpe, J.: On predictive models and user-drawn graphical passwords. ACM Trans. Inf. Syst. Secur. 10(4) (2008), http://doi.acm.org/10.1145/1284680.1284685
15. Yee, K.-P.: Aligning security and usability. IEEE Security and Privacy 2, 48–55 (2004)

Actor-Network Procedures
(Extended Abstract)

Dusko Pavlovic[1] and Catherine Meadows[2,⋆]

[1] Royal Holloway, University of London, and Universiteit Twente, EWI/DIES
dusko.pavlovic@rhul.ac.uk
[2] Naval Research Laboratory, Code 5543, Washington, DC 20375
catherine.meadows@nrl.navy.mil

Abstract. In this paper we propose *actor-networks* as a formal model of computation in heterogenous networks of computers, humans and their devices, where these new procedures run; and we introduce *Procedure Derivation Logic* (PDL) as a framework for reasoning about security in actor-networks, as an extension of our previous *Protocol Derivation Logic*. Both formalisms are geared towards graphic reasoning. We illustrate its workings by analysing a popular form of two-factor authentication.

1 Introduction

Over the last few years, almost without being aware of it, we have been seeing a marked change in our view computation and networking is. We are moving beyond networks of computers communicating on channels constructed out of wires and routers to networks of human beings and various types of devices communicating over multiple channels: wired, wireless, cellular, as well as human-usable channels based on voice and vision.

This change has been particularly relevant to security, and to the development and analysis of security protocols. Over the past twenty years or so, there has been extensive, and often influential work on the development of formal methods for the analysis of security protocols. One of the secrets of the success of this work is that it has been based on a simple but powerful model, first introduced in the late thirty years ago by Dolev and Yao [14], in which abstract principals communicate across a network controlled by a hostile intruder. This model has made it possible to develop both model checkers for determining whether or not attacks are possible, and logical systems for determining what a principal can conclude as a result of participating in a protocol. However, this network model is harder to apply in a heterogeneous networks using multiple types of channels.

Our goal is to contribute towards a formal framework for for *reliable* and *practical* reasoning about security of computation and communication in network engineering. Towards this goal, we draw our formal models from the informal reasoning practices, and attempt to make them mathematically precise, while

⋆ Supported by ONR.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 7–26, 2012.

trying to keep them as succinct and intuitive as possible. The main feature of our formalism is that it provides support for *diagrammatically* based security proofs, which we illustrate by examples in this paper. Our approach is intended to capture what we consider the two most salient concepts for security in this new paradigm for computing: the coalitions that are formed between humans and devices in order to enable secure computation and communication in these emerging networks, which we describe via the use of *actor-networks*, a concept borrowed from sociology, and the orchestration of different types of communication along different types of channels, which we describe by generalizing the notion of protocol to that of a *procedure*. These ideas are described in more detail below.

Actor-Networks. Networks have become an immensely popular model of computation across sciences, from physics and biology, to sociology and computer science [15,33,31]. Actor-networks [25] are a particularly influential paradigm in sociology, emphasizing and analyzing the ways in which the interactions between people and objects, as equal factors, drive social processes, in the sense that most people cannot fly without an airplane; but that most airplanes also cannot fly without people. Our goal in the present paper is to formalize and analyze some security processes in networks of people, computers, and the ever expanding range of devices and objects used for communication and networking, blurring many boundaries. The idea that people, computers, and objects are equal actors in such networks imposed itself on us, through the need for a usable formal model, even before we had heard of the sociological actor-network theory. After we heard of it, we took the liberty of adopting the name actor-network for a crucial component of our mathematical model, since it conveniently captures many relevant ideas. While the originators of actor-network theory never proposed a formal model, we believe that the tasks, methods and logics that we propose are not alien to the spirit of their theory. In fact, we contend that computation and society have pervaded each other to the point where computer science and social sciences already share their subject.

Procedures. In computer programs, frequently used sequences of operations are encapsulated into *procedures*, also called *routines*. A procedure can be called from any point in the program, and thus supports reuse of code.

In computer networks, frequently used sequences of operations are specified and implemented as *network protocols*, or as *cryptographic protocols*. So protocols are, in a sense, network procedures. Beyond computer networks, there are now hybrid networks, where besides computers with their end-to-end links, there may be diverse devices, with their heterogenous communication channels, cellular, short range etc. Online banking and other services are nowadays usually secured by two-factor and multi-factor authentication, combining passwords with smart cards, or cell phones. A vast area of *multi-channel* and *out-of-band* protocols opens up, together with the web service *choreographies* and *orchestrations*;

and we have only scratched its surface. And then there are of course also social networks, where people congregate with their phones, their cameras and their smiling faces, and overlay the wide spectrum of their social channels over computer networks and hybrid networks. Many sequences of frequently used operations within these mixed communication structures have evolved. This is what we call *actor-network procedures.*

Outline of the Paper. The remainder of the paper is organized as followed. In Sec. 2, we give an overview of related work, in particular the work that has most contributed to our own. Sec. 3 introduces the formal model of actor-networks. Sec 4 explains how actor-networks compute, and introduces the formalisms to represent that computation, all the way to actor-network procedures. Sec. 5 presents Procedure Derivation Logic (PDL) as a method for reasoning about actor-network procedures. In Sec. 6 we provide the first case studies using PDL: we analyze the two-factor authentication in online banking. Sec. 7 contains a discussion of the results and the future work.

2 Related Work

In social and computational networks, procedures come in many flavors, and have been studied from many angles. Besides cryptographic protocols, used to secure end-to-end networks, in hybrid networks we increasingly rely on multi-channel protocols [44], including device pairing [24]. In web services, standard procedures come in two flavors: choreographies and orchestrations [41]. There are, of course, also social protocols and social procedures, which were developed and studied first, although not formally modeled. As social networks are increasingly supported by electronic networks, and on the Web, social protocols and cryptographic protocols often blend together. Some researchers have suggested that the notion of protocol should be extended to study such combinations [6,19,23]. On the other side, the advent of ubiquitous computing has led to extensive, careful, but largely informal analyses of the problems, e.g., of device pairing, and of security interactions of using multiple channel types [44,22,32]. One family of device pairing proposals has been systematically analyzed in the computational model in [45,34,26,27].

There is a substantial and extremely successful body of research on the formal specification and verification of security protocols. These can be roughly be divided into work in the cryptographic model, which directly formalizes cryptographic reasoning, and the symbolic model, which represents data and computations symbolically as terms and operators in a term algebra. The symbolic approaches, in turn, can be divided into two approaches. The first relies on model checking, which is used to implement exhaustive search for attacks. This often comes together with proofs that exploration of a certain finite search space without finding an attack guarantees security. See [4] for a history and survey of model checking security protocols. The second builds on logical systems that are used to derive what a participant in a protocol can conclude after the completion of a run. The earliest logical system for cryptographic protocol analysis was the Burrows-Abadi-Needham logic [7]. A number of successful tools and logics followed. More recent work that has concentrated applying logical reconstructions includes the Cryptographic Protocol Shape Analyzer (CPSA) [13], the Protocol Composition Logic (described in more detail below), and our own Protocol Derivation Logic, which we use as the basis for logical framework developed in this paper.

It is well understood that model checking, by producing explicit attacks, can be very useful in aiding understanding of a protocol and where it has gone wrong. What seems to be less well appreciated is that the use of logical reconstructions can also give insight, but in a different way, since they give a clearer picture of the assumptions that are necessary for the security of a protocol, as well as the ways in which various pieces of the protocol contribute to its security. When combined with a diagrammatic reasoning, this can give considerable insight into the structure and applicability of a protocol. This is very important, since protocols are often reused and redesigned for different environments, with different security assumptions and types of communication channels. This approach of combining logical reconstruction with diagrammatic reasoning is the one we take in this paper.

Our formal model, as well as the basis for its diagrammatic support, is derived from the *strand space* model [20]. Among its many salient features, the convenient diagrammatic protocol descriptions of strand spaces has been an important reason for their wide acceptance and popularity. It is important to note that the strand space diagrams are not just an intuitive illustration, but that they are formal objects, corresponding to precisely defined components of the theory, while on the other hand closely resembling the informal "arrows-and-messages" protocol depictions, found in almost every research paper and on almost every white board where a protocol is discussed.

Protocol Composition Logic (PCL) was, at least in its early versions [18,12,10,17,11], an attempt to enrich the strand model with a variable binding and scoping mechanism, making it into a process calculus with a formal handle on data flows, which would thus allow attaching Floyd-Hoare-style annotations to protocol executions, along the lines of [39,40]. This was necessary for incremental refinement of protocol specifications, and for truly compositional, and thus scalable protocol analyses, which were the ultimate goal of the project. However, less attention was paid to the purely diagrammatic contribution of the strand space model.

Protocol Derivation Logic (PDL) has been an ongoing effort [29,8,36,2,30,37] towards a scalable, i.e. incremental protocol formalism, allowing composition and refinement like PCL, but equipped with an intuitive and succinct diagrammatic notation, like strand spaces. The belief that these two requirements can be reconciled is based on the observation that the reasoning of protocol participants is concerned mostly with the order of events in protocol executions. It follows that the protocol executions and their logical annotations both actually describe the same structures, which can be viewed as partially ordered multisets [43], and manipulated within the same diagrammatic language. This has been the guiding idea of PDL. Several case studies of standard protocols, and the taxonomies of the corresponding protocol suites, have been presented in [29,8,36,2]. An application to a family of distance bounding protocols has been presented in [30]; and an extension supporting the probabilistic reasoning necessary for another such family has been proposed in [37]. In the present paper, we propose the broadest view of PDL so far — which should here be read as *Procedure* Derivation Logic. Since cryptographic protocols are usually construed as the tools of computer security, we use the term *procedure* here to denote a frequently used pattern of operations in a modern network, which may include computers and software agents, but also humans, as well as various kinds of communication devices. Procedure Derivation Logic is thus our attempt to address the need for formal reasoning about the *pervasive* security problems, that arise at the interfaces of cyber space with physical and social spaces, as discussed in [38].

It should be mentioned that the mosaic of protocol logics, which we are thus attempting to expand beyond protocols, is, in a certain sense, conterbalanced by the more homogenous (if not entirely monolithic) world of computational modeling. Following the seminal work in [1], it has become customary to verify that every symbolic model, underlying a protocol logic, is sound when interpreted computationally. Such interpretations have led to many interesting results and useful tools [5,3,9]. It should be clear, however, that the standard computational model does not allow modeling many of the physical or social features that we are trying to capture in procedures. Since our execution model includes network nodes that represent, e.g., humans, and the visual channels through which these humans can look at each other, it does not seem reasonable to try to interpret it computationally. And a model can, of course, only be computationally sound, or unsound, relative to a specific computational interpretation. Without such an interpretation, the question of computational soundness cannot be stated.

3 Actor-Network Model

We model network computation in terms of (1) computational agents, some of them controlled by various parties, others available as resources, and (2) communication channels between the agents, supporting different types of information flows.

3.1 Formalizing Actor-Networks

Definition 3.1. *An* actor-network *consists of the following sets: (1)* identities*, or* principals $\mathcal{J} = \{A, B, \ldots\}$*, (2)* nodes $\mathcal{N} = \{M, N, \ldots\}$*, (3)* configurations $\mathcal{P} = \{P, Q, \ldots\}$*, where a configuration can be a finite set of nodes, or a finite set of configurations; (4)* channels $\mathcal{C} = \{f, g, \ldots\}$*, and (5)* channel types $\Theta = \{\tau, \varsigma, \ldots\}$ *given with the structure* $\Theta \xleftarrow{\vartheta} \mathcal{C} \overset{\delta}{\underset{\varrho}{\rightrightarrows}} \mathcal{P} \overset{©}{\rightharpoonup} \mathcal{J}$ *where the partial map*

© $: \mathcal{P} \rightharpoonup \mathcal{J}$ *tells which principals control which configurations, the pair of maps* $\delta, \varrho : \mathcal{C} \to \mathcal{P}$ *assign to each channel* f *an* entry δf *and an* exit ϱf*, and the map* $\vartheta : \mathcal{C} \to \Theta$ *assigns to each channel a type.*

An actor *is an element of a configuration.*

Notation. We denote by N_B a node N controlled by the principal ©$N = B$. We write $g = (P \xrightarrow{\tau} N_B)$ for a channel g of type $\vartheta g = \tau$, with the entry $\delta g = P$, and with the exit $\varrho g = N$ controlled by ©$N = B$. Since there is usually at most one channel of a given type between two given configurations, we usually omit the label g, and write just $P \xrightarrow{\tau} N_B$ to denote this channel.

3.2 Example: An Actor-Network for Two Factor Authentication

To mitigate phishing attacks, some online banks have rolled out two factor authentication. This means that they do not just verify that the user knows a password, but also something else — which is the second authentication factor. This second factor often requires some additional network resources, besides the internet link between the customer and the bank. This is the first, quite familiar step beyond simple cyber networks.

Some banks authenticate that the user is in possession of her smart card. The underlying actor-network is on Fig. 1. The user Alice controls her computer C_A and her smart card S_A. She is also given a portable smart card reader R. She inserts the card in the reader to form the configuration Q. The reader is available to Alice, but any other reader would do as well. Configured into Q, the smart card and the reader verify that Alice knows the PIN, and then generates the login credentials, which Alice copies from R's screen to her computer C_A's keyboard, which forwards it to bank Bob's computer C_B. The details of the authentication procedure will be analyzed later.

In summary, the network thus consists of principals $\mathcal{J} = \{A, B\}$, $\mathcal{N} = \{I_A, C_A, S_A, R, C_B\}$; configurations $\mathcal{P} = \mathcal{N} \cup \{Q\}$, where $Q = \{S_A, R\}$, and the following six channels: (1 and 2) cyber channels $C_A \leftrightarrows C_B$ between Alice's and Bank's computers, (3) a visual channel $C_A \to I_A$ from Alice's computer to her human I_A, (4) a keyboard $I_A \to C_A$ from Alice's human to her computer, (5) a visual channel $R \to I_A$ from the smart card reader to Alice's human, and (6) a keyboard $I_A \to R$ from Alice's human to the card reader.

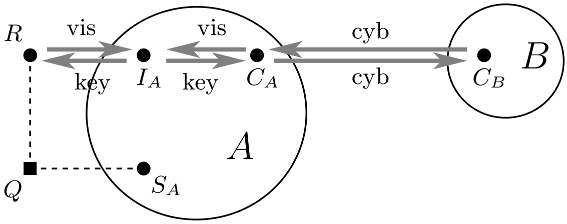


Fig. 1. A pervasive network: Online banking with a smart card reader

4 Actor-Network Processes

4.1 Computation and Communication

Computation in a network consists of *events*, which are localized at nodes or configurations. An event that is controlled by a principal is an *action.*

Communication in a network consists of *information flows* along the channels. Each flow corresponds to a pair of events: (1) a *write* event at the entry of the channel, and (2) a *read* event at the exit of the channel. There are two kinds of flows: (1) *messages*, which consist of a *send action* at the entry of the channel, and a *receive coaction* at the exit, and (2) *sources*, which consist of an *sample action* at the exit, and a *emit coaction* at the entry.

Besides transferring information from one configuration to another, the flows also synchronize the events that take place at different localities, because: every receive coaction must be preceded by a corresponding send action, and every sample action must be preceded by a corresponding emit coaction.

If a source has not been emitted to anywhere, then there is nothing to sample, and no sampling of that source can occur. If a message has not been sent, then the corresponding receive event cannot occur. So when I receive a message, then I know that it must have been sent previously by someone; and when I sample a source, then I know that someone must have emited to this source. That is how I draw conclusions about non-local events from the observations of my own local actions. This is formalized in Sec. 5.2.

4.2 Formalizing Data as Terms

Each flow carries some data, which contain information. As is standard in the symbolic protocol model, we represent this as terms in an algebra. Recall that an algebraic theory is a pair (O, E), where O is a set of finitary operations (given as symbols with arities), and E a set of well-formed equations (i.e. where each operation has a correct number of arguments) [21].

Definition 4.1. *An algebraic theory* $\mathbb{T} = (O, E)$ *is called a* data theory *if* O *includes a binary pairing* $(-,-)$ *operation, and the unary operations* π_1 *and* π_2 *such that* E *contains the equations* $\pi_1(u,v) = u$*,* $\pi_2(u,v) = v$*, and* $((x,y),z) = (x,(y,z))$*. A* data algebra *is a polynomial extension* $\mathcal{T}[\mathcal{X}]$ *of a* $\mathbb{T}$*-algebra* $\mathcal{T}$*.*

Function notation. When no confusion seems likely, we elide the function applications to concatenation, and write $f.x$ instead of $f(x)$.When no confusion is likely, we even elide the dot from the concatenation and simply write fx instead of $f.x$, or $f(x)$.

Random values are represented by indeterminates. A polynomial extension $\mathcal{T}[\mathcal{X}]$ is the free $\mathbb{T}$-algebra generated by adjoining a set of *indeterminates* $\mathcal{X}$ to a $\mathbb{T}$-algebra $\mathcal{T}$ [21, §8]. The elements $x, y, z \ldots$ of $\mathcal{X}$ are used to represent nonces and other randomly generated values.

Easy subterms. We assume that every data algebra comes equipped with the *easy subterm relation* $\sqsubseteq$. The idea is that that $s \sqsubseteq t$ implies that s is a subterm of t such that every principal who knows t also knows s. In other words, the views $\varGamma_A$ are lower closed under $\sqsubseteq$, as explained in [36]. This is in contrast with hard subterms, which cannot be extracted: e.g., the plaintext m and the key k are hard subterms of the encryption $E.k.m$.

4.3 Formalizing Events and Processes

In this section we define processes, the events that processes engage in, and the ordering of events within a process.

An event or action is generally written in the form $a[t]$ where a is the event identifier, and t is the term on which the event may depend. When an event does not depend on data, the term t is taken to be a fixed constant $t = \checkmark$, and we often abbreviate $a[\checkmark]$ to a.

The most important events for our analyses are the action-coaction couples send-receive, and sample-emit, for which we introduce special notations: send $\langle\cdot\, t \,\cdot\rangle$, receive $(\cdot\, t \,\cdot)$, and emit $\langle:t:\rangle$, sample $(:t:)$. Generically, we write $\langle\, t \,\rangle$ for a write action, which can be either $\langle\cdot\, t \,\cdot\rangle$ or $\langle:t:\rangle$, and $(\, t \,)$ for a read action, which can be either $(\cdot\, t \,\cdot)$ or $(:t:)$. Another often used action is $\nu[x]$ for the generation of a random value. It could also be implemented as sampling a source of randomness represented as a devoted node. In addition, the nodes are capable of performing various local operations, which are specified in the definition of the procedure. For actions, such as $\langle\cdot\, t \,\cdot\rangle$ and $(:t:)$, the configuration P must be controlled, i.e. the partial function $\copyright : \mathcal{N} \rightharpoonup \mathcal{J}$ must have a definite value $\copyright P$.

Definition 4.2. *A* process $\mathcal{F}$ *is a partially ordered multiset of localized events, i.e. a mapping*

$$\mathcal{F} \;=\; \langle \mathcal{F}_{\mathbb{E}}, \mathcal{F}_{\mathcal{P}} \rangle : \mathbb{F} \to \mathbb{E} \times \mathcal{P}$$

where $(\mathbb{F}, \to)$ *is a well-founded partial order, representing the structure time,* $\mathbb{E}$ *is a family of events, and* $(\mathcal{P}, \subseteq)$ *the partial order of configurations, and they satisfy the requirements that*

(a) if $\mathcal{F}_{\mathbb{E}}\phi$ *is an action, then* $\copyright(\mathcal{F}_{\mathcal{P}}\phi)$ *is well defined, and*
(b) if $\phi \to \psi$ *in* $\mathbb{F}$ *then* $\mathcal{F}_{\mathcal{P}}\phi \subseteq \mathcal{F}_{\mathcal{P}}\psi$ *or* $\mathcal{F}_{\mathcal{P}}\phi \supseteq \mathcal{F}_{\mathcal{P}}\psi$ *in* $\mathcal{P}$.

Notation: The points in time are denoted by events. By abuse of notation, we usually write $a[t]_P$ for $\phi \in \mathcal{F}$ where $\mathcal{F}_{\mathbb{E}}\phi = a[t]$ and $\mathcal{F}_{\mathcal{P}} = P$.

(a) if an action takes place at a configuration P, then P is controlled, i.e. $\copyright P$ must be well defined, and
(b) if $a[t]_P \to b[s]_Q$ then $P \subseteq Q$ or $P \supseteq Q$.

Definition 4.3. *We say that the term* t originates *at the point* $\phi \in \mathcal{F}$ *if* ϕ *is the earliest write of a term containing* t*. Formally,* ϕ *thus satisfies* $\mathcal{F}_{\mathbb{E}}\phi = \langle\, s \,\rangle$ *where* $t \sqsubseteq s$*, and* $\mathcal{F}_{\mathbb{E}}\xi = \langle\, s \,\rangle \wedge t \sqsubseteq s \Longrightarrow \phi \to \xi$ *holds for all events* ξ*.*

Notation: Origination. We extend the notational conventions described above by denoting by $\sqrt{\langle\langle\, t \,\rangle\rangle_P}$ the event ϕ where the term t originates. The configuration P is the *originator* of t.

4.4 Formalizing Flows, Runs and Procedures

We now extend our discussion to the definition of communication between processes, and extend our ordering to events occurring within a procedure as well as individual processes.

We begin by defining a more general version of channel between two configurations, called a flow channel. A flow channel exists between any two configurations if a channel exists between any two nodes on the configuration trees. It is called a flow channel because the information passed along the channel flows upwards to the configuration as a whole. It is defined formally below.

Definition 4.4. *For configurations $P, Q \in \mathcal{P}$, a* flow channel $P \xrightarrow{\tau} Q$ *can be either (1) a channel $P \overset{\tau}{\to} Q$, (2) a flow channel $P \xrightarrow{\tau} Q'$, where $Q' \in Q$, (3) a flow channel $P' \xrightarrow{\tau} Q$, where $P' \in P$, or (4) a flow channel $P' \xrightarrow{\tau} Q'$, where $P' \in P$ and $Q' \in Q$.*

A flow $a[t]_P \xrightarrow{\tau} b[s]_Q$ *is given by a flow channel $P \xrightarrow{\tau} Q$, and an interaction pair $a[t], b[s]$, i.e. a pair where either $a[t] = \langle \cdot\, t \,\cdot \rangle$ and $b[s] = (\cdot\, s \,\cdot)$, or $a[t] = \langle : t : \rangle$, and if $b[s] = (: s :)$.*

A flow $a[t]_P \xrightarrow{\tau} b[s]_Q$ is complete *if $s = t$.*

Definition 4.5. *Let $\mathcal{F}$ be a process. A* run, *or* execution $\mathcal{E}^{\mathcal{F}}$ *of $\mathcal{F}$ is an assignment for each coaction $b[s]_Q$ of a unique flow $a[t]_P \xrightarrow{\tau} b[s]_Q$, which is required to be* sound, *in the sense that $b[s]_Q \not\to a[t]_P$ in $\mathcal{F}$.*

A run is complete *if all of the flows that it assigns are complete: the terms that are received are just those that were sent, and the inspections find just those terms that were submitted.*

A run is a pomset extending its process. Setting $a[t]_P \to b[s]_Q$ whenever there is a flow $a[t]_P \xrightarrow{\tau} b[s]_Q$ of some type τ makes a run $\mathcal{E}^{\mathcal{F}}$ into an extension of the ordering of the process $\mathcal{E}$, as a partially ordered multiset. The pomset $\mathcal{E}^{\mathcal{F}}$ does not have to satisfy condition (b) of Def. 4.2 any more. Indeed, the whole point of running a process is to extend in $\mathcal{E}^{\mathcal{F}}$ the internal synchronizations, given by the ordering of $\mathcal{F}$, with the additional external synchronizations.

Definition 4.6. *A* network procedure $\mathcal{L}$ *is a pair $\mathcal{L} = \langle \mathcal{F}_{\mathcal{L}}, E_{\mathcal{L}} \rangle$ where $\mathcal{F}_{\mathcal{L}}$ is a process, and $E_{\mathcal{L}} = \{\mathcal{E}_1^{\mathcal{F}_{\mathcal{L}}}, \mathcal{E}_2^{\mathcal{F}_{\mathcal{L}}}, \mathcal{E}_3^{\mathcal{F}_{\mathcal{L}}} \ldots\}$ is a set of runs of $\mathcal{F}_{\mathcal{L}}$. The elements of $E_{\mathcal{L}}$ are called* secure *runs. All other runs are* insecure. *A procedure is said to be secure if every insecure run can be detected by a given logical derivation from the observations of a specified set of participants.*

Graphic presentations of procedures. To specify a procedure $\mathcal{L}$, we draw a picture of the pomset $\mathcal{F} = \mathcal{F}_{\mathcal{L}}$, and then each of its extensions $\mathcal{E} = \mathcal{E}_i^{\mathcal{F}_{\mathcal{L}}}$. Because of condition (b) of Def.4.2, the events comparable within the ordering of a process $\mathcal{F}$ must happen within a maximal configuration. Therefore, if the diagram of the partially ordered multiset $\mathcal{F}$ is drawn together with the underlying network, then each component of the comparable events can all be depicted under the corresponding configuration. We can thus draw the network above the process, and place the events occurring at each configuration along the imaginary vertical lines flowing, say, downwards from it, like in Fig. 3. The additional ordering, imposed when in a run $\mathcal{E}$ the messages get sent and the facts get observed, usually run across, from configuration to configuration. This ordering can thus be drawn along the imaginary horizontal lines between the events, or parallel with the channels of the network. Such message flows can also be seen in Fig. 3. The dashed lines represent the data sharing within a configuration.

4.5 Examples of Procedures

Challenge Response Authentication Protocols. We begin a familiar special case of a procedure: a protocol. A large family of challenge-response authentication protocols is subsumed under the template depicted on Fig. 2. Bob wants to make sure that Alice is online. It is assumed that Alice and Bob share a secret k^{AB}, which allows them to define functions c^{AB} and r^{AB} such that

- $r^{AB}x$ can be computed from $c^{AB}x$ using s^{AB}, but
- $r^{AB}x$ can*not* be computed from $c^{AB}x$ alone, without s^{AB}.

So Bob generates a fresh value x, sends the challenge $c^{AB}x$, and if he receives the response $r^{AB}x$ back, he knows that Alice must have been online, because she must have originated the response. The idea behind this template has been discussed, e.g., in [29,8,36,37]. The template instantiates the concrete protocol components by refining the abstract functions c^{AB} and r^{AB} to concrete implementations, which satisfy the above requirements: e.g., c^{AB} may be the encryption by Alice's public key, and r^{AB} may be the encryption by Bob's public key, perhaps with Alice's identity.

Two-Factor Authentication Procedure. Next we describe the first nontrivial procedure, over the actor-network described in Sec. 3.2. It can be viewed as an

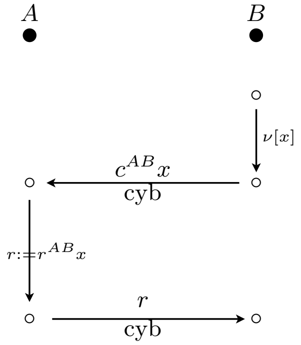


Fig. 2. Challenge-Response (CR) protocol template

extension of the simple challenge-response authentication. There, Bob authenticates Alice using her knowledge of a secret s^{AB}, which they both know. Here Bob authenticates that that knows a secret p^A that Bob does not know, and that she has a security token S_A, in this case a smart card. The secret and the smart card are the "two factors". This is the idea of the procedure standardized under the name *Chip Authentication Programme (CAP)*, analyzed in [16]. The desired run of the challenge-response option of this procedure is depicted on Fig. 3.

We assume that, prior to the displayed run, Alice the customer identified herself to Bob the bank, and requested to be authenticated. Bob's computer C_B then extracts a secret s^{AB} that he shares with Alice. This time, though, the shared secret is too long for Alice's human I_A to memorize, so it is is stored in the smart card S_A. Just like in CR protocol above, Bob issues a challenge, such that the response can only be formed using the secret. So Bob in fact authenticates the smart card S_A. He entrusts the smart card S_A with authenticating Alice's human I_A. This is done using the secret p^A shared by I_A and S_A. The secret is stored in both nodes. To form the response to Bob's challenge, Alice forms the configuration Q by inserting her card S_A into the reader R. The configuration Q requests that I_A enters the secret PIN (Personal Identification Number) p^A before it forms the response for Bob. There is no challenge from Q to I_A, and thus no freshness guarantees in this authentication: anyone who sees I_A's response can replay it at any time. Indeed, the human I_A cannot be expected to perform computations to authenticate herself: most of us have trouble even submitting just the static PIN. The solution is thus to have the card-reader configuration Q computes the response, which Alice relays it to Bob. The old PIN authentication is left to just convince Q that Alice's human I_A is there: Q tests p^A, sent through the keybord channel from I_A to the reader R, coincides with $\overline{p}^A$ stored in the card S_A, and then generates a keyed hash $Hs^{AB}x$ using the shared secret s^{AB} and the challenge x. This hash is displayed for Alice on the card reader R as the response r, which Alice then sends to her computer C_A by the keyboard channel, and further to C_B by the cyber channel.

5 Procedure Derivation Logic

5.1 The Language of PDL

A statement of PDL is in the form $A : \Phi$, where $A \in \mathcal{J}$ is a principal, and Φ is a predicate asserted by A. The predicate Φ is formed by applying logical connectives to the atomic predicates, which can be (1) $a[t]_P$ — meaning "the event $a[t]_P$ happened", or (2) $a[t]_P \to b[s]_Q$ — meaning "the event $a[t]_P$ happened before $b[s]_Q$".

5.2 Communication Axioms

The statements of PDL describe the events that happen in a run of a process, and their order. The basic PDL statements are its axioms, which we describe next. They are taken to be valid in all runs of all processes. The other valid statements are derived from them.

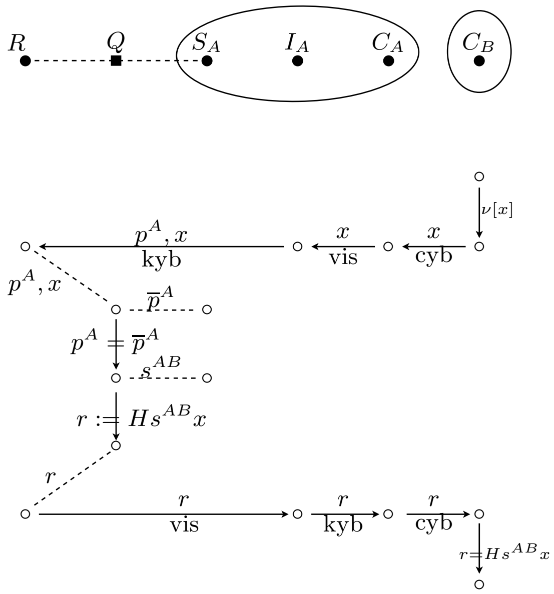


Fig. 3. Chip Authentication Program (CAP) procedure

Origination. The origination axioms say that any message that is received must have been sent, and that any source that is sampleed must have been emitted to. This has been explained early in Sec.4. More precisely, any principal that controls a configuration P where a message is received knows that it must have been sent by someone, no later than it was received; and similarly for a source that is sampleed. Formally

$$\copyright P \;:\; (\cdot\, t\,\cdot)_P \Longrightarrow \exists X.\ \langle\cdot\, t\,\cdot\rangle_X \to (\cdot\, t\,\cdot)_P \qquad \textsf{(orig.m)}$$

$$\copyright P \;:\; (:t:)_P \Longrightarrow \exists X.\ \langle: t :\rangle_X \to (: t :)_P \qquad \textsf{(orig.s)}$$

Freshness. In Sec. 4.2 we explained the idea of modeling random values as the indeterminates in polynomial algebras of messages. The freshness axiom extends this idea to processes, by requiring that each indeterminate x must be (1) *freshly generated* by an action $\nu[x]$ before it is used anywhere, and (2) that it can only be used elsewhere after it has passed in a message or a source. which formally becomes

$$\copyright P \;:\; a[t.x]_P \Longrightarrow \exists X.\ \nu[x]_X \to a[t.x]_P \qquad \textsf{(fresh.1)}$$

$$\begin{aligned}\copyright P \;:\; \neg\nu[x]_P \wedge a[t.x]_P \Longrightarrow \exists X.\ &\big(\nu[x]_X \to \langle\langle\cdot\ x\ \cdot\rangle\rangle_X \to ((\cdot\ x\ \cdot))_P \to a[t.x]_P\big)\\ &\vee \big(\nu[x]_X \to \langle\langle: x :\rangle\rangle_X \to ((: x :))_P \to a[t.x]_P\big) \qquad \textsf{(fresh.2)}\end{aligned}$$

where, using the easy subterm order $\sqsubseteq$ from Sec. 4.2, $\langle\langle\cdot\ x\ \cdot\rangle\rangle_X$ abbreviates $\exists t.\ x \sqsubseteq t\ \wedge\ \langle\cdot\, t\,\cdot\rangle_X$, $((\cdot\ x\ \cdot))_X$ abbreviates $\exists t.\ x \sqsubseteq t\ \wedge\ (\cdot\, t\,\cdot)_X$, etc.

5.3 Authentication Axioms

In our model, there are two forms of authentication: interactions along authentic channels, and challenge-response authentication.

Interactions along Authentic Channels. An authentic channel allows at least one of the participants to observe not only the events on their own end of the channel, but also on the other end. So there are four types of authentic channels, supporting the following assertions:

$$\begin{array}{lll@{\qquad}lll}
\copyright P & : & \langle\cdot\, t\,\cdot\rangle_P \to (\cdot\, t\,\cdot)_Q \;\; (\mathsf{auch.m.1}) & \copyright P & : & \langle :t:\rangle_P \to (:t:)_Q \;\; (\mathsf{auch.p.1}) \\
\copyright Q & : & \langle\cdot\, t\,\cdot\rangle_P \to (\cdot\, t\,\cdot)_Q \;\; (\mathsf{auch.m.2}) & \copyright Q & : & \langle :t:\rangle_P \to (:t:)_Q \;\; (\mathsf{auch.p.2})
\end{array}$$

Channels that satisfy auch.m.1 or auch.p.1 are called *write*-authentic; channels that satisfy auch.m.2 or auch.p.2 are called *read*-authentic. Here are some examples from each family:

- (auch.m.1): A keyboard channel guarantees to the sender that the device at which she is typing is receiving the message
- (auch.m.2): A visual channel used for sending a message allows the receiver to see the sender.
- (auch.p.1): When my fingerprints are taken, I observe that they are taken, and can see who is taking them.
- (auch.p.2): Moreover, the person taking my fingerprints also observes that they are taking my fingerprints.

Besides these assertions about the order of events, some authentic channels support other assertions. They are usually application specific, and we impose them as procedure specific axioms.

Challenge-Response Authentication. The challenge-response axiom is in the form

$$\copyright P \;:\; \mathsf{Local}_P \Longrightarrow \mathsf{Global}_{PQ} \qquad (\mathsf{cr})$$

where, using the notation from Sec. 5.2

$$\begin{array}{rcl}
\mathsf{Local}_P & = & \nu[x]_P \to \left\langle \cdot\, c^{PQ}x \,\cdot \right\rangle_P \qquad\qquad \to \qquad\qquad \left(\cdot\, r^{PQ}x \,\cdot \right)_P \\
\mathsf{Global}_{PQ} & = & \nu[x]_P \to \left\langle \cdot\, c^{PQ}x \,\cdot \right\rangle_P \to \left(\left(\cdot\; c^{PQ}x \;\cdot \right)\right)_Q \to \sqrt{\langle\langle \cdot\; r^{PQ}x \;\cdot \rangle\rangle_Q} \to \left(\cdot\, r^{PQ}x \,\cdot \right)_P
\end{array}$$

Translated into words, (cr) says that the owner $\copyright P$ of the configuration P knows that (1) if he generates a fresh x, sends the challenge $c^{PQ}x$, and receives the response $r^{PQ}x$, then (2) Q must have received a message containing $c^{PQ}x$ after he sent it, and then she must have sent a message containing $r^{PQ}x$ before he received it.

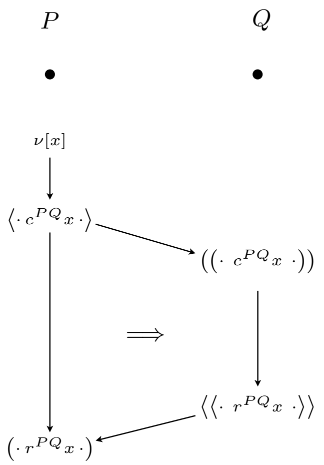

P
Q
$\nu[x]$
$\langle\cdot\, c^{PQ}x\,\cdot\rangle$
$((\cdot\; c^{PQ}x\;\cdot))$
$\Longrightarrow$
$\langle\langle\cdot\; r^{PQ}x\;\cdot\rangle\rangle$
$(\cdot\, r^{PQ}x\,\cdot)$

Fig. 4. The graphic view of (cr) axiom

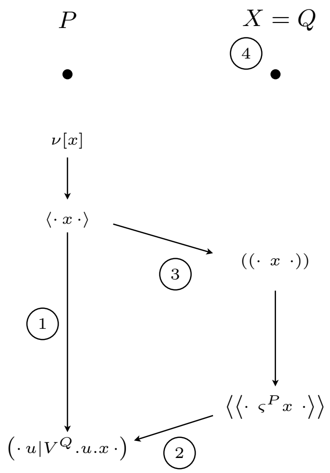

P
$X = Q$
4
$\nu[x]$
$\langle\cdot\; x\;\cdot\rangle$
3
$((\cdot\; x\;\cdot))$
1
$\langle\langle\cdot\; \varsigma^{P}x\;\cdot\rangle\rangle$
$(\cdot\; u|V^{Q}.u.x\;\cdot)$
2

Fig. 5. Challenge-response using signatures

Using (cr) and certain observations of the local events at P, the principal ©P can thus draw the conclusions about certain non-local events at Q, which he cannot directly observe. Fig. 4 depicts this reasoning diagrammatically.

Remark. The (cr) axiom, and the corresponding protocol template, displayed on Fig. 5, has been one of the crucial tools of the Protocol Derivation Logic, all the way since [29,8], through to [37].

6 Examples of Reasoning in PDL

6.1 On the Diagrammatic Method

In its diagrammatic form depicted on Fig. 5, axiom (cr) says that the verifier P, observing the local path on the left, can derive the path around the non-local actions on the right. This pattern of reasoning resembles the categorical practice of *diagram chasing* [28,35]. Categorical diagrams are succinct encodings of lengthy sequences of equations. Just like the two sides of the implication in (cr) correspond to two paths around Fig. 5, the two sides of an equation are represented in a categorical diagram as two paths around a face of that diagram. The components of the terms in the equations correspond to the individual arrows in the paths. The equations can be formally reconstructed from the diagrams. Moreover, the diagrams can be formally combined into new proofs. The algebraic structures are thus formally transformed into geometric patterns. After some practice, the geometric intuitions begin to guide algebraic constructions in the formal language of diagrams. We apply a similar strategy to PDL.

6.2 Cryptographic (Single-Factor) Authentication

We begin with a simple example of diagrammatic reasoning, present already in [29].

Theorem 6.1. *The functions $c^{PQ}x = x$ and $r^{PQ}x = \varsigma^Q x$ implement* (cr)*, provided that the abstract signature function ς satisfies the following axioms:*

- *(a) $\varsigma^Q u = \varsigma^Q v \Longrightarrow u = v$, i.e., ς^Q is injective,*
- *(b) $\sqrt{\langle\!\langle \varsigma^Q t \rangle\!\rangle_X} \Longrightarrow X = Q$, i.e., $\varsigma^Q t$ must originate from Q,*
- *(c) $V^Q.u.t \iff u = \varsigma^Q t$, i.e., the predicate V^Q is satisfied just for the pairs u, t where $u = \varsigma^Q t$,*

and that these axioms are known to the principal Bob $= \copyright P$.

Proof. To prove the claim, we chase the diagram on Fig. 10. The numbered arrows arise from the following steps:

1. Bob $= \copyright P$ observes $\nu[x]_P \to \langle \cdot\, x \cdot \rangle_P \to \big(\cdot\, r | V^Q r x \cdot\big)$, i.e. after sending a fresh value x, he receives a response u which passes the verification $V^Q r x$.
2. Using the axioms (c) and (orig.m), he concludes that there is some X such that $\big\langle \cdot\, V^Q x \cdot \big\rangle_X \to \big(\cdot\, r | V^Q r x \cdot\big)_P$.
3. Using (fresh.2) he further derives that for the same X holds $\langle \cdot\, x \cdot \rangle_X \to ((\cdot\, x \cdot))_X \to \big\langle \cdot\, V^Q x \cdot \big\rangle_X$.
4. Using (a) and (b), Bob concludes that $V^Q x$ must have originated from Q.

□

6.3 Pervasive (Two-Factor) Authentication

Next we describe how Bob the bank authenticates Alice the customer in the CAP procedure.

Theorem 6.2. *The procedure on Fig. 3 implements authentication, i.e. satisfies (cr), provided that the following assumptions are true, and known to Bob:*

- *(a) $Hu = Hv \Longrightarrow u = v$, i.e., H is injective;*
- *(b) $\sqrt{\langle\!\langle s^{AB} \rangle\!\rangle_X} \Longrightarrow X = S_A \vee X = C_B$, i.e., s^{AB} must originate from S_A or C_B;*
- *(c) $\sqrt{\langle\!\langle p^A \rangle\!\rangle_X} \Longrightarrow X = I_A \vee X = S_A$, i.e., p^A must originate from I_A or S_A;*
- *(d) $\big\langle \cdot\, H s^{AB} x \cdot \big\rangle_Q \Longrightarrow \Big(\big(\cdot\, p^A, x \cdot\big)_Q \to \big\langle \cdot\, H s^{AB} x \cdot \big\rangle_Q \Big) \wedge p^A = \overline{p}^A$, i.e., S_A and R are honest.*

Proof. Prior to the displayed execution, Alice is assumed to have sent to Bob her identity, and a request to be authenticated. Following this request, Bob's computer C_B has extracted the secret s^{AB} from a store, which he will use to verify that S_A has generated the response.

To prove the claim, we chase the diagram on Fig. 6. The enumerated steps in the diagram chase correspond to the following steps in Bob's reasoning:

1. Bob observes $\nu[x]_{C_B} \to \langle \cdot\, x \cdot \rangle_{C_B} \to \big(\cdot\, H s^{AB} x \cdot\big)_{C_B}$.
2. Using (orig.m) he concludes that there is some X such that $\langle \cdot\, H s_A x \cdot \rangle_X \to \big(\cdot\, H s^{AB} x \cdot\big)_{C_B}$.

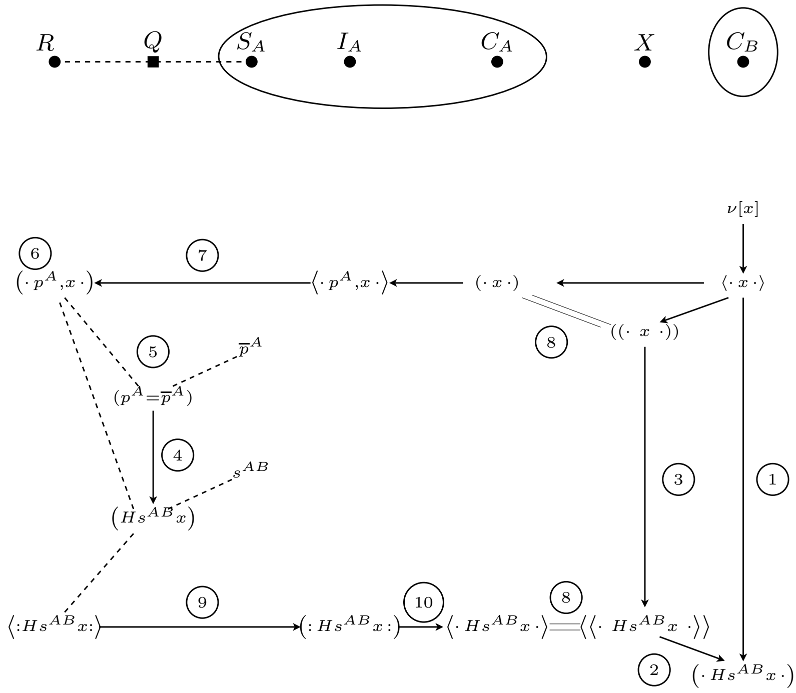

R
Q
S_A
I_A
C_A
X
C_B
$\nu[x]$
6
7
$(\cdot\, p^A, x\, \cdot)$
$\langle\cdot\, p^A, x\, \cdot\rangle$
$(\cdot\ x\ \cdot)$
$\langle\cdot\ x\ \cdot\rangle$
$((\cdot\ x\ \cdot))$
8
5
$\overline{p}^A$
$(p^A = \overline{p}^A)$
4
s^{AB}
$(Hs^{AB}x)$
3
1
9
10
8
$\langle : Hs^{AB}x : \rangle$
$(: Hs^{AB}x :)$
$\langle\cdot\ Hs^{AB}x\ \cdot\rangle$
$\langle\langle\cdot\ Hs^{AB}x\ \cdot\rangle\rangle$
2
$(\cdot\ Hs^{AB}x\ \cdot)$

Fig. 6. B's reasoning in CAP

3. Using (fresh.2) he further derives that for the same X holds $\langle\cdot\, x\,\cdot\rangle_{C_B} \to ((\cdot\ x\ \cdot))_X \to \langle\cdot\, Hs^{AB}x\,\cdot\rangle_X$.
4. By (a) and (b), from the observation that he did not use s^{AB}, Bob concludes that $Hs^{AB}x$ must have originated in a configuration Q containing S_A.
5. By (c), $\langle\langle\cdot\ p^A\ \cdot\rangle\rangle_{I_A} \to ((\cdot\ p^A\ \cdot))_Q \to (p^A = \overline{p}^A) \to (Hs^{AB}x)$, where the last action abbreviates $(r := Hs^{AB}x)$, and we write out r as $Hs^{AB}x$ in the rest of the diagram.
6. Since Q had to also receive x before computing the response in $(\cdot\, p^A, x\,\cdot)_R \to (Hs^{AB}x)$ follows by (d). So $((\cdot\ p^A\ \cdot))_Q$ from 5 is $(\cdot\, p^A, x\,\cdot)_R$.
7. By (orig-m), there is Y with $\langle\cdot\, p^A, x\,\cdot\rangle_Y \to (\cdot\, p^A, x\,\cdot)_R$. By (e), $\langle\langle\cdot\ p^A\ \cdot\rangle\rangle_{I_A}$ from 5 must be $\langle\cdot\, p^A, x\,\cdot\rangle_{I_A}$.
8. The fresh value x has thus been sent to Q by I_A. It follows that in 2 and 3 above must be $X = I_A$.
9. Since A controls S_A and I_A, and $S_A \in Q$ generated the response $Hs^{AB}x$, only I_A could have sampled $Hs^{AB}x$ along the visual channel.
10. Since A controls I_A and C_A, only I_A could have sent $Hs^{AB}x$ to C_A along the keyboard channel.

These logical steps suffice to assure Bob that if he observes the local flow on the right in Fig. 6, then the non-local flow along the external boundary, all the way to the left side of the diagram and back must have taken place. Comparing this diagrammatic conclusion with the pattern of (cr) on Fig. 5, we see that Bob has proven an instance of authentication. □

We have provided a security proof of the CAP protocol discussed in [16]. However, the discussion in that paper is devoted to pointing out the security risks inherent in relying upon that very protocol. How can this happen? As it turns out, we can reproduce the situations that the authors in [16] warn against by relaxing the assumptions that our proof relies upon. This is one advantage of combining logical reconstruction with explicit specifications of the configurations and channels involved.

Relaxing the assumption that Alice is honest: In this case $\sqrt{\langle\langle p^A \rangle\rangle_X}$ fails to hold, because if Alice is not honest she could turn PIN over to a third party. This is discussed in [16] in terms of the card being stolen and Alice being intimidated into revealing her PIN.

Relaxing the assumption that R is honest: in this case $\sqrt{\langle\langle p^A \rangle\rangle_X}$ fails to hold again, because S_A or R could reveal the PIN to a third party. This is discussed in [16] in two places. First, the reader could inadvertently reveal the PIN because the keys used to enter it become visibly worn. Secondly, for practical reasons users are often required to use untrusted readers provided by third parties, which could steal the PIN.

Relaxing the assumption that Alice controls C_A: if C_A has been infiltrated by malware then Alice no longer controls it. As the reader can verify, this has absolutely no effect on the proof of security of the CAP and PIN protocol in isolation. Indeed, C_A could be replaced by a cyber channel where ever it is used without affecting the protocol's security. The problem is when the hash computed by the smart card is used to authenticate a bank transaction. The most straightforward way of doing this is for the hash to be passed to C_A, which computes, for example, a Message Authentication Code on the transaction. If C_A is not controlled by Alice, it could substitute a different transaction.

7 Conclusion

We have presented a logical framework for reasoning about security of protocols that make use of a heterogeneous mixture of humans, devices, and channels. We have shown how different properties of channels and configurations can be expressed and reasoned about within this framework. A key feature of this framework is that it supports explicit reasoning about both the structure of a protocol and the contributions made by its various components, using a combination of diagrammatic and logical methods. Because of this, we believe that our approach can be particularly useful in giving a more rigorous foundation for white-board discussions, in which protocols are usually displayed graphically. By annotating the diagram with the proof using the methods described in this paper, formal reasoning could be brought to bear at the very earliest stages of the design process.

Acknowledgement. The first author would like to thank Wolter Pieters [42] for introducing him to Bruno Latour's ideas about actor-networks.

References

1. Abadi, M., Rogaway, P.: Reconciling two views of cryptography (the computational soundness of formal encryption). J. of Cryptology 15(2), 103–127 (2002)
2. Anlauff, M., Pavlovic, D., Waldinger, R., Westfold, S.: Proving authentication properties in the Protocol Derivation Assistant. In: Proc. FCS-ARSPA 2006. ACM (2006)
3. Barthe, G., Hedin, D.L., Béguelin, S.Z., Grégoire, B., Heraud, S.: A machine-checked formalization of sigma-protocols. In: Proc. CSF 2010, pp. 246–260. IEEE Computer Society (2010)
4. Basin, D., Cremers, C., Meadows, C.: Model checking security protocols. In: Clarke, E., Henzinger, T., Veith, H. (eds.) Handbook of Model Checking. Springer, Heidelberg (to appear, 2011), `http://people.inf.ethz.ch/cremersc/publications/index.html`
5. Blanchet, B.: A computationally sound mechanized prover for security protocols. IEEE Trans. Dependable Sec. Comput. 5(4), 193–207 (2008)
6. Blaze, M.: Toward a Broader View of Security Protocols. In: Christianson, B., Crispo, B., Malcolm, J.A., Roe, M. (eds.) Security Protocols 2004. LNCS, vol. 3957, pp. 106–120. Springer, Heidelberg (2006)
7. Burrows, M.L., Abadi, M., Needham, R.: A Logic of Authentication. ACM Trans. Computer Systems 8(1), 18–36 (1990)
8. Cervesato, I., Meadows, C., Pavlovic, D.: An encapsulated authentication logic for reasoning about key distribution protocols. In: Proc. CSFW 2005, pp. 48–61. IEEE (2005)
9. Cortier, V., Kremer, S., Warinschi, B.: A survey of symbolic methods in computational analysis of cryptographic systems. J. Autom. Reasoning 46(3-4), 225–259 (2011)
10. Datta, A., Derek, A., Mitchell, J., Pavlovic, D.: Secure protocol composition. E. Notes in Theor. Comp. Sci., 87–114 (2003)
11. Datta, A., Derek, A., Mitchell, J., Pavlovic, D.: A derivation system and compositional logic for security protocols. J. of Comp. Security 13, 423–482 (2005)
12. Datta, A., Derek, A., Mitchell, J.C., Pavlovic, D.: A derivation system for security protocols and its logical formalization. In: Proc. of CSFW 2003, pp. 109–125. IEEE (2003)
13. Doghmi, S.F., Guttman, J.D., Thayer, F.J.: Searching for Shapes in Cryptographic Protocols. In: Grumberg, O., Huth, M. (eds.) TACAS 2007. LNCS, vol. 4424, pp. 523–537. Springer, Heidelberg (2007)
14. Dolev, D., Yao, A.C.: On the security of public key protocols. IEEE Transactions on Information Theory 29(2), 198–208 (1983)
15. Dorogovtsev, S.N., Mendes, J.F.F.: Evolution of Networks: From Biological Nets to the Internet and WWW. Oxford University Press (2003)

16. Drimer, S., Murdoch, S.J., Anderson, R.: Optimised to Fail: Card Readers for Online Banking. In: Dingledine, R., Golle, P. (eds.) FC 2009. LNCS, vol. 5628, pp. 184–200. Springer, Heidelberg (2009)
17. Durgin, N., Mitchell, J., Pavlovic, D.: A compositional logic for proving security properties of protocols. J. of Comp. Security 11(4), 677–721 (2004)
18. Durgin, N., Mitchell, J.C., Pavlovic, D.: A compositional logic for protocol correctness. In: Proc. of CSFW 2001, pp. 241–255. IEEE (2001)
19. Ellison, C.: Ceremony design and analysis. Cryptology ePrint Archive, Report 2007/399 (October 2007)
20. Thayer Fabrega, J., Herzog, J., Guttman, J.: Strand spaces: What makes a security protocol correct? Journal of Computer Security 7, 191–230 (1999)
21. Gratzer, G.A.: Universal Algebra. Van Nostrand, Princeton (1968)
22. Hoepman, J.-H.: Ephemeral Pairing on Anonymous Networks. In: Hutter, D., Ullmann, M. (eds.) SPC 2005. LNCS, vol. 3450, pp. 101–116. Springer, Heidelberg (2005)
23. Karlof, C.S., Tygar, J.D., Wagner, D.: Conditioned-safe ceremonies and a user study of an application to web authentication. In: Proceedings of the 5th Symposium on Usable Privacy and Security, SOUPS 2009, p. 38:1 ACM (2009)
24. Kumar, A., Saxena, N., Tsudik, G., Uzun, E.: A comparative study of secure device pairing methods. Pervasive Mob. Comput. 5, 734–749 (2009)
25. Latour, B.: Reassembling the Social: An Introduction to Actor-Network Theory. Oxford University Press (2005)
26. Laur, S., Nyberg, K.: Efficient Mutual Data Authentication Using Manually Authenticated Strings. In: Pointcheval, D., Mu, Y., Chen, K. (eds.) CANS 2006. LNCS, vol. 4301, pp. 90–107. Springer, Heidelberg (2006)
27. Laur, S., Pasini, S.: User-aided data authentication. IJSN 4(1/2), 69–86 (2009)
28. Mac Lane, S.: Categories for the Working Mathematician. Graduate Texts in Mathematics, vol. 5. Springer, Heidelberg (1971)
29. Meadows, C., Pavlovic, D.: Deriving, Attacking and Defending the GDOI Protocol. In: Samarati, P., Ryan, P.Y.A., Gollmann, D., Molva, R. (eds.) ESORICS 2004. LNCS, vol. 3193, pp. 53–72. Springer, Heidelberg (2004)
30. Meadows, C., Poovendran, R., Pavlovic, D., Chang, L., Syverson, P.: Distance bounding protocols: authentication logic analysis and collusion attacks. In: Poovendran, R., Wang, C., Roy, S. (eds.) Secure Localization and Time Synchronization in Wireless Ad Hoc and Sensor Networks. Springer, Heidelberg (2006)
31. Newman, M.: Networks: An Introduction. Oxford University Press (2010)
32. Nguyen, L.H., Roscoe, A.W.: Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey. Journal of Computer Security (to appear, 2011)
33. Palsson, B.O.: Systems Biology: Properties of Reconstructed Networks. Cambridge University Press (2006)
34. Pasini, S., Vaudenay, S.: SAS-Based Authenticated Key Agreement. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 395–409. Springer, Heidelberg (2006)
35. Pavlovic, D.: Maps II: Chasing diagrams in categorical proof theory. J. of the IGPL 4(2), 1–36 (1996)
36. Pavlovic, D., Meadows, C.: Deriving Secrecy in Key Establishment Protocols. In: Gollmann, D., Meier, J., Sabelfeld, A. (eds.) ESORICS 2006. LNCS, vol. 4189, pp. 384–403. Springer, Heidelberg (2006)
37. Pavlovic, D., Meadows, C.: Bayesian authentication: Quantifying security of the Hancke-Kuhn protocol. E. Notes in Theor. Comp. Sci. 265, 97–122 (2010)

38. Pavlovic, D., Meadows, C.: Deriving ephemeral authentication using channel axioms. In: Proceedings of the Cambridge Workshop on Security Protocols 2009. Springer, Heidelberg (2010) (to appear)
39. Pavlovic, D., Smith, D.R.: Composition and refinement of behavioral specifications. In: Proc. Automated Software Engineering 2001. IEEE (2001)
40. Pavlovic, D., Smith, D.R.: Guarded Transitions in Evolving Specifications. In: Kirchner, H., Ringeissen, C. (eds.) AMAST 2002. LNCS, vol. 2422, pp. 411–425. Springer, Heidelberg (2002)
41. Peltz, C.: Web services orchestration and choreography. Computer 36, 46–52 (2003)
42. Pieters, W.: Representing humans in system security models: An actor-network approach. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications 2(1), 75–92 (2011)
43. Pratt, V.: Modelling concurrency with partial orders. Internat. J. Parallel Programming 15, 33–71 (1987)
44. Stajano, F., Wong, F.-L., Christianson, B.: Multichannel Protocols to Prevent Relay Attacks. In: Sion, R. (ed.) FC 2010. LNCS, vol. 6052, pp. 4–19. Springer, Heidelberg (2010)
45. Vaudenay, S.: Secure Communications over Insecure Channels Based on Short Authenticated Strings. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 309–326. Springer, Heidelberg (2005)

Knowledge as a Window into Distributed Coordination[*]

Yoram Moses

Department of Electrical Engineering
Technion—Israel Institute of Engineering
moses@ee.technion.ac.il

1 Introduction

Distributed and multi-agent systems come in many forms, serving various purposes and spanning a large variety of properties. They include networked processors communicating via message-passing, shared-memory systems in which processes interact by reading from and writing to shared variables, and even systems of robots that coordinate their actions by viewing each others' actions and locations, and perform no explicit communication actions. While distinct systems may differ completely in their detailed structure and operation, fundamental to all distributed systems is the fact that decisions are performed based on a local, partial, view of the the state of the system. Proper coordination among different sites of such a system requires information flow among them, to ensure that decisions are taken based on appropriate knowledge. Reasoning about when elements of the system do or do not know relevant facts is therefore a central aspect of the design and construction of distributed systems and distributed protocols.

Let us begin with an example. Consider nodes in a computer network that start out with individual initial values (their ID's, say) and need to compute the maximal such value in the system. For simplicity assume that node number 1 needs to print the maximal value. If, by chance, this node holds the maximal value at the outset, then printing this value would satisfy the requirements. But it would typically be wrong for node 1 to do so without interacting with other nodes first: Before outputting a value, node 1 must *know* that this value is maximal. And this is different from seeing or holding the maximal value. But how can it *know* what is the maximal value? There are many ways by which such knowledge may be obtained, and these depend on the protocol being executed, on the system's properties, and on the setting in which the protocol operates. One possibility would be for node 1 to collect all values in the system and choose the maximum. But this can be rather costly, requiring many redundant values to be sent in the system. After all, there is no need to send information about a value that is smaller than one previously reported on. In some cases, it would suffice for node 1 to know that it has received a message chain starting from

[*] The title follows that of an inspiring *RSA Animate* talk on `Language as a Window into Human Nature` by Steven Pinker viewable on YouTube.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 27–34, 2012.

every node in the network. But even this need not always be required. Under some protocols the node can use the passage of time to determine that it holds the maximal value, even without message chains from all nodes to it. In other cases, the ID values are bounded, and if the maximal possible value is seen, it can be printed immediately. These are but a few of the many different types of possible solutions to a simple problem. What is shared by all different solutions, however, is that node 1 can act only once it *knows* that a given value is maximal.

The dependence of actions on knowledge properties is a general phenomenon. Some knowledge requirements follow directly from a problem's specification (computing the maximum, dispensing cash by an ATM only if the account has sufficient credit), while others can arise from the structure of the protocol, or from coordination requirements. For example, suppose that Bob is allowed to perform an action a_2 only after Alice has performed a related action a_1. Then whenever he performs a_2, Bob must know that a_1 was performed. Moreover, if Alice can only perform a_1 under some condition φ—say her account on an online shop's site must be approved—then Bob must know that Alice knows φ before he can perform a_2. If Susan can determine (i.e., knows) that Bob does not have such knowledge, she will know that a_2 has not been performed yet.

While the need for knowing the maximal value in our original example is straightforward, making sense out of statements regarding knowledge about the knowledge of others can quickly become hard to follow and sometimes even rather suspect. Yet, in some cases such knowledge precisely captures the underlying structure of protocols solving particular problems of interest. Studying information flow and its implications on the attainability of states of knowledge can often provide useful insights into how to solve such problems. Fortunately, there is a rigorous framework that enables reasoning about fairly convenient and flexible way to model knowledge so that it can be applied to the analysis of almost every distributed protocol. A formal theory of knowledge in distributed systems has been developed over the last three decades, providing tools for analyzing this aspect of distributed action and interaction [6,5]. As we mentioned above, one of the chief facilitators of achieving various states of knowledge is communication. There are well-known results about how the evolution of knowledge depends on communication [2,6,10,7,11]. In the last couple of years, extensions have been made capturing the interaction between knowledge, communication and coordination. The combined results provide insights into the fundamental elements underlying coordination in multi-agent systems. The purpose of this invited talk is to illustrate the use of knowledge theory in the analysis of distributed coordination. This short article is intended to provide a simple guide to some of the basic material underlying the invited talk.

The article is structured as follows. The next section will sketch my favorite way of defining knowledge in a distributed system. Section 3 will illustrate how knowledge theory can be related to coordination by showing that linear coordination requires a state of nested knowledge (i.e., knowledge about knowledge). A classical theorem by Chandy and Misra will then be reviewed, which relates message chains and gaining nested knowledge in asynchronous contexts. Finally, Section 4 contains a discussion of the connection between knowledge and coordination beyond the scope presented in this article.

2 The Knowledge Framework

A flexible framework (called the *runs and systems*, or *interpreted systems* framework) for reasoning about knowledge in multi-agent systems is presented by Fagin, Halpern, Moses, and Vardi in [5]. The basic idea is that to every pair (P, γ) consisting of a *context* γ describing a model of distributed computation and a protocol P that can be executed in γ, corresponds a *system* $R = R(P, \gamma)$ consisting of the set of all possible runs of P in γ. Knowledge is defined with respect to such a system. Since there are many very different distributed contexts, this approach makes it possible to reason about the knowledge in a wide variety of settings.

2.1 Sketch of the Model

We shall simplify the exposition here, suppressing many details. The purpose is to review just enough of the details to support the analysis of coordination reviewed in this paper. For a more elaborate and detailed exposition, see [9] and chapters 4 and 5 of [5].

We view a multi-agent system as consisting of a set $\mathbb{P} = \{1, \ldots, n\}$ of agents. At any point in time the system is in some *global state*, which we think of as an instantaneous "snapshot" of the system. A *run* of the system is a function from time to global states. Intuitively, a run is a complete description of what happens over time in one possible execution of the system. For simplicity, time here is taken to range over the natural numbers rather than the reals (so that time is viewed as discrete, rather than dense or continuous). Thus, $r(t)$ denotes the system's global state at time t in run r. Essential to our approach is the assumption that in every global state, each agent i has a well-defined *local state*. We denote by $r_i(t)$ agent i's local state in the global state $r(t)$.

We are generally interested in facts whose truth value can change over time. For example,"$x = 0$" may be true at time 5 and false at time 10. So facts are evaluated at a *point* (r, t), corresponding to time t in the run r. Being able to reason about knowledge requires a bit more. Intuitively, we say that Alice knows that Bob received her message if this must be the case, in all possible instances in which Alice has the information that is currently at her disposal. In a system in which messages are guaranteed to be delivered in one step, Alice can know this one round after she sends the message to Bob. In a system where messages may be lost, for example, Alice might need to receive an acknowledgement. The set of possible instances referred to above is not the same in both cases. We formally capture this as follows. We identify an agent's local information at a point (r, t) with her local state there $r_i(t)$. Knowledge is defined at a point with respect to a set R of runs, which we call a *system*, and which is typically of the form $R(P, \gamma)$.

Fix a system R. We say that two points (r, t) and (r', t') are *indistinguishable* to agent i if $r_i(t) = r'_i(t')$. Intuitively, we identify agent i's information at a given point with its local state there.

We are now ready to describe how knowledge is formally defined.[1] We will use

$$(R, r, t) \models \varphi$$

to denote that the fact φ is true at the point (r, t) with respect to the system R. Suppose that we defined the set of points of R at which φ holds, and let us denote *agent i knows* φ by $K_i\varphi$. Then knowledge is defined by

$$(R, r, t) \models K_i\varphi \quad \text{iff} \quad (R, r', t') \models \varphi \ \text{ whenever } \ r'_i(t') = r_i(t) \text{ and } r' \in R.$$

The Alice and Bob scenario just described, a run r in which a message that Alice sends to Bob at time 0 arrives in one step can belong both to the reliable system R_1 and to the unreliable one R_2. Suppose that $\mathsf{rec'd}$ denotes the fact that Bob received Alice's message. In the first system, the fact $K_A\mathsf{rec'd}$ will hold at $(r, 1)$ with respect to the system R_1, and it will not hold with respect to R_2. Hence, $(R_1, r, 1) \models K_A\mathsf{rec'd}$ and $(R_2, r, 1) \not\models K_A\mathsf{rec'd}$. In this case, there is a run $r' \in R_2 \setminus R_1$ such that $r'_A(1) = r_A(1)$ and in r' Alice's message was not delivered by time 1, so that $(R_2, r', 1) \not\models \mathsf{rec'd}$.

Observe that the definition of knowledge can be applied repeatedly. Namely, if the truth of φ is at all points of R is well defined, then so is that of $K_i\varphi$. So the fact $\psi = K_i\varphi$ is also well defined, and we can apply the definition to it, to give precise formal meaning to $K_j\psi = K_jK_i\varphi$. Formally, $(R, r, t) \models K_jK_i\varphi$ means that agent j's local state at (r, t) provides ample evidence to support the claim that agent i's local state at (r, t) provides ample evidence that φ is true at r and t. In the same fashion, precise formal meaning can be given to facts involving arbitrarily deep nested knowledge.

3 Nested Knowledge and Linear Coordination

Recall from the introduction that statements involving nested knowledge statements—regarding what one agent knows about what another one knows about what a third does..., etc., may be hard to parse and make sense of. We have just seen how they can be given precise meaning in the runs and systems framework for knowledge. But are such properties of any practical use? We now show how they relate to sequential coordination. Following [1], we focus on actions that are coordinated in response to a spontaneous action. In many distributed applications, the system must be able to respond to events that are initiated by the environment. This can be an external input to the system such as a cash withdrawal request from an online account, a compute job sent to a cloud application, a link or process failure, or a fire alarm being set off. The distinguishing feature of a spontaneous event is that when and whether it will

[1] A full formalization is obtained by defining a logic for knowledge; we focus only on its core definition.

occur is unknown ahead of time, and is independent of any actions initiated by the agents in the system. In a system in which agent 1 will always begin with an initial value $v_1 = 0$, for example, a fact such as $K_3K_2K_1(v_1 = 0)$ can be true without any interaction or information flow taking place. Knowledge about the occurrence of a spontaneous event at a remote site, however, will necessarily involve information flow of some type.

Let $\mathcal{A} = \langle \mathsf{a}_1, \ldots, \mathsf{a}_k \rangle$ be a sequence of actions that, in the system R, can be performed only by processes $i_1, \ldots, i_k$, respectively. Moreover, assume for ease of exposition that i_j can perform a_j at most once in a given run. Finally, we assume that each agent i_j's local state records whether i_j has performed a_j. (Thus, agents remember whether they performed their $\mathcal{A}$ actions.) We say that $\mathcal{A}$ is a *conditional ordered response* (cOR) to a spontaneous event e_s in R if throughout all runs

1. a_1 can only take place after *trig* has occurred, and
2. a_{j+1} can only take place after a_j has been performed, for $1 \leq j < k$.

By definition, if $\mathcal{A}$ is a conditional ordered response to e_s, then some prefix of $\mathcal{A}$ is performed in every run in which e_s takes place. Importantly, the actions in this prefix take place in their stated sequential order. A particular instance of a cOR is a case in which *all* actions of $\mathcal{A}$ are guaranteed to occur whenever e_s takes place.

We can now state and prove a strong connection between cOR and nested knowledge:

Proposition 1. *Let e_s be a spontaneous event in the system R, and denote by $\mathsf{oc'd}(e_\mathsf{s})$ the fact that e_s has occurred already in the current run. Moreover, let $\mathcal{A} = \langle \mathsf{a}_1, \ldots, \mathsf{a}_k \rangle$ be a conditional ordered response to e_s in the system R. If the action $\mathsf{a}_j \in \mathcal{A}$ takes place at time t_j in the run $r \in R$, then*

$$(R, r, t_j) \vDash K_{i_j} K_{i_{j-1}} \cdots K_{i_1} \mathsf{oc'd}(e_\mathsf{s}).$$

Proof. We prove by induction on j that $(R, r, t^*) \vDash K_{i_j} K_{i_{j-1}} \cdots K_{i_1} \mathsf{oc'd}(e_\mathsf{s})$ holds for all times $t^* \geq t_j$. The claim will follow because $t_j \geq t_j$.

$j = 1$: For the base case, a_1 takes place at time t_1 in r. Let $t^* \geq t_1$. To show that $(R, r, t^*) \vDash K_{i_1} \mathsf{oc'd}(e_\mathsf{s})$ we need to show that $(R, r', t') \vDash \mathsf{oc'd}(e_\mathsf{s})$ holds at all points (r', t') such that $r'_{i_1}(t') = r_{i_1}(t^*)$. Fix such a run r'. Since the fact that i_1 has performed a_1 is by assumption recorded in i_1's local state $r_{i_1}(t^*) = r'_{i_1}(t')$, we have that a_1 has been performed by i_1 by time t' in r'. By definition of cOR we have that e_s has occurred by time t' in r', and so $(R, r', t') \vDash \mathsf{oc'd}(e_\mathsf{s})$, as required.

$j > 1$: For the inductive step, assume that the claim is true for $j - 1$ in all runs r', that a_j occurs at time t_j in r, and that $t^* \geq t_j$. The argument is now analogous to the previous case. Fix a point (r', t') such that $r'_{i_j}(t') = r_{i_j}(t^*)$. The local state $r'_{i_j}(t')$ thus records that a_j has been performed. Since $j > 1$ we have by definition of cOR that a_{j-1} is performed in r' before time t'. By the inductive hypothesis for $j - 1$

and r' we have that $(R, r', t') \vDash K_{i_{j-1}} \cdots K_{i_1} \mathsf{oc'd}(e_\mathsf{s})$, and it follows that $(R, r, t^*) \vDash K_{i_j} K_{i_{j-1}} \cdots K_{i_1} \mathsf{oc'd}(e_\mathsf{s})$, as desired. □

Proposition 1 states that sequential ordering of actions is guaranteed to yield states of nested knowledge. Perhaps more interestingly, however, it can be viewed as pinpointing a *precondition* for performing actions in a cOR: Before the stated nested knowledge formula for a_j is true, agent i_j *cannot* perform the action a_j! Indeed, by analyzing how nested knowledge can arise in different contexts, it is possible to establish what it must take to guarantee that a set of responses to a spontaneous event are performed in proper linear order. This leads naturally to a central result in the theory of knowledge in distributed systems, due to Chandy and Misra [2], which we now briefly sketch. They considered the *asynchronous message passing model*, which is a very popular and well studied model of distributed systems. In this setting, processes are connected via a network of links, and they communicate by sending each other messages. Processes are assumed to have no clock, and there are no guarantees about the relative rates at which processes operate or the time it takes messages to be delivered.

In asynchronous message-passing systems, a process has no guarantee of progress at other sites other than that obtained by explicit communication. In a seminal paper, Lamport defined the *happened-before* relation among events in an asynchronous system [8]. Essentially, it says in such a setting an event e_i at site i can causally affect another event e_j at site j if there is a message chain starting at i at or after e_i occurs, and reaching j no later than when e_j does. Chandy and Misra formalized this connection in terms of knowledge gain and nested knowledge. In our terminology it implies the following statement:

Proposition 2 (Chandy and Misra). *Let γ be an asynchronous context and let $R = R(P, \gamma)$. Moreover, let e_s be a spontaneous event occurring at site i_0 in the run $r \in R$. If*

$$(R, r, t) \vDash K_{i_j} K_{i_{j-1}} \cdots K_{i_1} \mathsf{oc'd}(e_\mathsf{s}),$$

then there is a message chain in r starting at site i_0 after e_s occurred, passing through i_1, i_2, …, i_j in this order, and reaching i_j by time t.

Combining the two propositions we can conclude that the only way in which actions can be linearly ordered in an asynchronous context is via explicit message chains. Clearly, the converse is also true: We can design protocols that ensure cOR based solely on constructing a message chain of the required form. Notice that slightly more than the existence of a message chain can be shown. Since in the asynchronous context every step of progress is in a precise sense a spontaneous event (there can be arbitrarily long pauses between steps), Proposition 2 can be used to conclude further constrain the segments of the message chain required in a cOR $\mathcal{A} = \langle \mathsf{a}_1, \ldots, \mathsf{a}_k \rangle$. In fact, if $h < j$ then there must be a message chain in the run r leaving i_h after a_h occurs, and reaching i_{h+1} before a_{h+1} does.

4 Discussion and Further Directions

This short note presented background and some basic results relating knowledge and distributed coordination. Beyond the scope here were additional aspects that will be discussed in the invited talk. The tight connection nested knowledge and ordered action captured by Proposition 1 is true in all models of distributed computing. In a precise sense, Proposition 1 provides a reduction of linear coordination to nested knowledge. Being able to prove such results is an illustration of the power of knowledge theory. By studying how nested knowledge can be obtained in particular models, we can gain insight into the underlying structure of coordination.

Proposition 2 characterizes how nested knowledge can be gained in asynchronous contexts. It demonstrates the very close connection between nested knowledge and message chains. Message chains of this form are not necessary in other distributed contexts. A general challenge is to prove analogues of Proposition 2 in other contexts. In the talk I will discuss recent work with Ido Ben Zvi, in which this has been done for *synchronous contexts*, in which processes share a global clock, and timing guarantees are available [1]. In this setting, nested knowledge can be gained without the type of message chain needed in the asynchronous context. Consequently, there is much broader range of solutions to the problem of linearly ordering responses to spontaneous events. In the talk, I will discuss how timing information combines with communication to give rise to new causal structures (called *centipedes*) that strictly generalize message chains.

Beyond the scope of this article is the connection between knowledge theory and simultaneous coordination. This connection was developed in [3,6,5,4]. Proposition 2 can be used to show that common knowledge of spontaneous events is unattainable in asynchronous contexts. In synchronous contexts common knowledge *is* attainable, and simultaneous coordination is possible. The interaction between timing guarantees and communication in this case provides new insights into the structure underlying simultaneous coordination [1].

In summary, there are interesting and nontrivial connections between forms of coordination and the states of knowledge that they require. We have only begun to discover these connections and to make use of the insights that they provide. The study of knowledge gain and information flow in different types of contexts, as well as the study of the states of knowledge underlying particular forms of coordination, promises to be exciting and fruitful.

References

1. Ben-Zvi, I., Moses, Y.: Beyond Lamport's *Happened-Before*: On the Role of Time Bounds in Synchronous Systems. In: Lynch, N.A., Shvartsman, A.A. (eds.) DISC 2010. LNCS, vol. 6343, pp. 421–436. Springer, Heidelberg (2010)
2. Chandy, K.M., Misra, J.: How processes learn. Distributed Computing 1(1), 40–52 (1986)
3. Dwork, C., Moses, Y.: Knowledge and common knowledge in a Byzantine environment: crash failures. Information and Computation 88(2), 156–186 (1990)

4. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Common knowledge revisited. In: Shoham, Y. (ed.) Proc. Sixth Conference on Theoretical Aspects of Rationality and Knowledge, pp. 283–298. Morgan Kaufmann, San Francisco (1996)
5. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about Knowledge. MIT Press, Cambridge (2003)
6. Halpern, J.Y., Moses, Y.: Knowledge and common knowledge in a distributed environment. Journal of the ACM 37(3), 549–587 (1990); A preliminary version appeared in Proc. 3rd ACM Symposium on Principles of Distributed Computing (1984)
7. Krasucki, P.J., Ramanujam, R.: Knowledge and the ordering of events in distributed systems (extended abstract). In: Proc. Theoretical Aspects of Reasoning About Knowledge, pp. 267–283. Morgan Kaufmann (1994)
8. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. Communications of the ACM 21(7), 558–565 (1978)
9. Moses, Y.: Reasoning about knowledge and belief. In: van Harmelen, F., Lifschitz, V., Porter, B. (eds.) Handbook of Knowledge Representation, ch. 15, pp. 621–648. Elsevier B.V. (2008)
10. Parikh, R., Krasucki, P.: Levels of knowledge in distributed computing. Sādhanā 17(1), 167–191 (1992)
11. Parikh, R., Ramanujam, R.: Distributed Processes and the Logic of Knowledge. In: Parikh, R. (ed.) Logic of Programs 1985. LNCS, vol. 193, pp. 256–268. Springer, Heidelberg (1985)

Decision Making as Optimization in Multi-robot Teams

Lynne E. Parker

University of Tennessee, Knoxville, TN 37996-3450, USA
LEParker@utk.edu
http://web.eecs.utk.edu/~parker

Abstract. A key challenge in multi-robot teaming research is determining how to properly enable robots to make decisions on actions they should take to contribute to the overall system objective. This article discusses how many forms of decision making in multi-robot teams can be formulated as optimization problems. In particular, we examine the common multi-robot capabilities of task allocation, path planning, formation generation, and target tracking/observation, showing how each can be represented as optimization problems. Of course, globally optimal solutions to such formulations are not possible, as it is well-known that such problems are intractable. However, many researchers have successfully built solutions that are approximations to the global problems, which work well in practice. While we do not argue that all decision making in multi-robot systems should be based on optimization formulations, it is instructive to study when this technique is appropriate. Future development of new approximation algorithms to well-known global optimization problems can therefore have an important positive impact for many applications in multi-robot systems.

Keywords: Distributed robots, multi-robot systems, decision making.

1 Introduction

The topic of multi-robot systems[1] has been extensively studied for the past two decades. The typical design objective in creating multi-robot teams is to enable the group of robots to solve physical tasks in a manner that is superior to single-robot systems. Many advances have been made in this field (e.g., see [32] for an overview), with some systems beginning to work their way into practical applications (e.g., for warehousing operations [45]).

These advances are possible because of the successful development of solutions to many challenges, including: (a) the design of sophisticated robot hardware that can physically achieve demanding tasks, (b) the development of advanced sensors and perception systems that can provide robots with detailed knowledge

[1] For our purposes, we define *multi-robot systems* to be groups of autonomous mobile robots that operate simultaneously in a shared workspace.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 35–49, 2012.

of their environment and the state of their mission, (c) the development of robust and reliable communications systems that allow robots to share information when distributed across a (potentially large) workspace, and (d) the design of intelligent robot control software that enables robots to achieve globally coherent results from individual local control actions. While all of these challenges are equally important, in this article we focus on the latter challenge — the development of intelligent software control for multi-robot teaming.

In a broad sense, robot control software could be considered equivalent to *decision making* in multi-robot teams. In some sense, every action that a robot takes is indeed based on a decision that the robot has made. However, not all multi-robot systems are typically characterized as decision making systems. The concept of decision making usually connotes a *cognitive* mental process, involving processes such as understanding, attention, reasoning, judgment, and so forth. Many multi-robot systems — especially the *swarm robotics* approaches (e.g., [20,26,27]) — do not make use of such cognitive processes, and instead incorporate relatively simple control laws that result in emergent group behavior. Since the type of control implemented in these systems does not involve cognitive processes, most robotics researchers would not call this control *decision making.*

On the other hand, a different class of multi-robot systems involves more direct and purposeful interaction (e.g., [30,34,38,40]). These systems consist of robots with higher-level reasoning capabilities, and with possibly varying sensory, computational, and physical skills. In these systems, robots must decide how to coordinate their actions in a more deliberative manner. Thus, these multi-robot systems could indeed be accurately described as decision making systems.

Interestingly, many forms of decision making in *intentionally cooperative* multi-robot systems can be formulated as optimization problems. While we do not claim that all decision making in these domains can, or should, be viewed in this manner, it is instructive to review some representative techniques that have been demonstrated to enable robots to successfully work together in a variety of applications. This article reviews some common optimization formulations that have been presented in the literature for this domain. However, we do not attempt a complete survey of the literature on this topic, since that is beyond the scope of this article.

We begin in Section 2 by providing additional background material for understanding decision making in multi-robot teams. Then, in Section 3, we look at some common multi-robot capabilities that are suitable for formulation as optimization problems. Sections 4 through 7 examine each of these capabilities in more detail. Since the globally optimal solution typically requires excessive computation (i.e., the global problem formulations are NP-complete), distributed approaches that approximate the global solution are typically pursued; example techniques are also discussed in these sections. We offer concluding remarks in Section 8.

2 Background: Types of Multi-robot Interaction

To place multi-robot decision making in context, we briefly introduce the most common forms of interaction in multi-robot teams (see [33] for a more detailed discussion of these types of systems). These types of interaction are characterized as follows:

- *Collective* multi-robot teams typically involve simple robots that are not aware of other robots on the team, even though they share common goals. In these systems, individual actions are typically beneficial to the team as a whole, and contribute to the team-level objective.
- *Cooperative* multi-robot teams involve robots that are aware of other robots on the team, and share goals with other teammates. Individual robot actions are usually beneficial to the objectives of the team as a whole.
- *Collaborative* multi-robot teams consist of robots that are aware of each other, and have individual goals. However, even though their individual goals may not be identical, they are willing to work with others when needed to help them achieve their individual goals.
- *Coordinative* multi-robot teams are composed of robots with individual goals, but which work together with other robots in a shared workspace to minimize inter-robot interference. In these systems, robots do not actively try to help other robots, but instead work to actively avoid interference.

Decision making most commonly occurs in the latter three types of systems, since the first (collective) type of multi-robot system does not typically include cognitive robot skills. Note that none of these types of interaction incorporates *adversarial* robots that work actively against each other. While this is a popular subject of study in multi-robot systems, especially for the domain of multi-robot soccer (e.g., [6,17,44]), we presume for the purposes of this article that all robots in the workspace share the objective of minimizing negative interference with other robots. In the sections that follow, we refer back to these types of interactions, noting which type of interaction is commonly studied in different multi-robot applications.

3 Decision Making as Optimization

To understand decision making in multi-robot teams, it is instructive to consider the types of applications and tasks to which multi-robot teams are applied. As outlined in [32] in more detail, some common applications for multi-robot teams include foraging, coverage, search, warehouse management, surveillance and security, construction and assembly, cooperative manipulation, search and rescue, and soccer. Each application area has its own unique challenges. However, many of these domains make use of solutions to some fundamental multi-robot interaction skills, including task allocation, path planning, formation control, and target tracking or observation. Thus, we examine these latter four areas in more detail in this article, since they have broad relevance to many multi-robot applications.

In all of these multi-robot interaction capabilities, the decision making process can be formulated as an optimization problem. Typically, these are formulated as combinatorial optimization [29] or as convex optimization [5] problems, in order to take advantage of the many tools available for these type of optimization.

Importantly, however, these problems are typically not treated as *global* optimization problems for multi-robot applications, since such problems are known to be NP-complete. Since most robotic applications require real-time robot response, there is insufficient time to calculate globally optimal solutions for most applications; such solutions are only possible for very small-scale problems. Instead, typical solutions use distributed methods that incorporate only local cost/utility metrics. While such approaches can only achieve approximations to the global solution, they often are sufficient for practical applications.

As previously mentioned, the four representative multi-robot interaction skills that are often framed as optimization problems are as follows:

– *Task allocation*: Optimize a combination of robot cost and task utility in mapping a set of robots to a set of tasks [13].
– *Path planning*: Generate paths for multiple robots that minimize a performance metric [35]. Typical performance metrics include combined robot path lengths, combined travel times for robots to reach their respective goals, and combined energy use.
– *Formations*: Enable robots to move into a desired formation, or to maintain a specified formation, while moving through the environment. A common quality metric is to minimize the error between each current robot position and that robot's assigned position in the formation [28].
– *Target tracking or observation*: Control cooperative robot motions to ensure that a group of targets remains under observation by the robots. The typical metric is to optimize a combination of the time targets are under observation and a robot cost function [31].

The following sections present formulations of these capabilities as optimization problems. Examples are also given of approximation approaches that have been proposed for these problems.

4 Optimization in Task Allocation

Simply put, the *task allocation* problem in multi-robot systems is determining the proper mapping of a set of robots to a set of tasks, so as to maximize the total utility of the system. The task allocation problem arises frequently in *cooperative*, *collaborative*, and *coordinative* types of multi-robot teams.

In [13], Gerkey and Mataric define a taxonomy that covers several variations of the multi-robot task allocation problem. This taxonomy is defined on three axes, each of which has two possible settings: (1) robots — single-task (ST) versus multi-task (MT), (2) tasks — single-robot (SR) versus multi-robot (MR), and (3) assignments — instantaneous (IA) versus time-extended (TA). A particular task allocation problem is denoted by one choice from each list, such as ST-MR-IA, which is the "easiest" of the task allocation problems.

Each variant requires a different formulation of the optimization problem. For example, if solved in a centralized manner, Gerkey notes that the ST-MR-IA problem can make use of the Hungarian method [21] to find an optimal solution in polynomial time. Distributed variants make use of auction algorithms (e.g., [4]), which are based on economics-inspired metaphors, in which tasks are put up for bid by robots, who then propose their cost (or utility) in performing the task. Robots are awarded tasks based on maximizing the utility of the system. Many implementations of this market-based approach have been developed; an overview of this literature is given in [9].

Other variants of the problem are related to well-known NP-complete problems such as the Set Covering Problem (SCP), and the Set Partitioning Problem (SPP). For example, as noted in [13], viewing the ST-MR-IA multi-robot task allocation problem as an instance of SPP can be stated as:

Definition 1. *Given a finite set of robots R, a family F of acceptable subsets of R that represent all feasible robot coalition-task pairs, and a utility function $u : F \rightarrow \mathbb{R}_+$, find a maximum-utility family X of elements in F such that X is a partition of R.*

Some proposed task allocation approaches (e.g., [43]) have adapted existing SCP and SPP approximation algorithms (e.g., [7,15]), making them relevant to the multi-robot domain. Example results from [47], which make use of market-based techniques to enable multi-robot teams to achieve an exploration task, are shown in Figure 1.

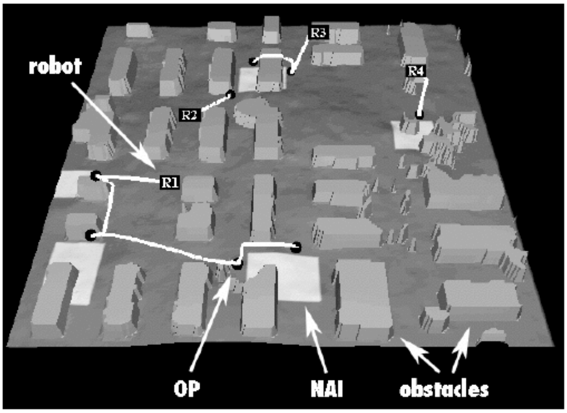


Fig. 1. Illustration of task allocation results in an exploration problem. From [47].

To provide a more detailed example of an approximation approach to task allocation, we briefly overview the ASyMTRe work of [34], which addresses the ST-MR-IA task allocation variant of the Gerkey taxonomy [13]. In this variant, the objective is to achieve task allocation for single-task robots (ST) performing multi-robot tasks (MR) using instantaneous assignment (IA). This problem variant is also called the *coalition formation* problem. While this problem has been addressed extensively in the multi-agent community (e.g., [18,36,37]), it has been noted by Vig [42] that most of the multi-agent approaches to coalition formation cannot be directly transferred to multi-robot applications, since robot capabilities and sensors are situated directly on the robots and are not transferable between robots.

The ASyMTRe approach is aimed at enabling sensor-sharing across robots for the purpose of forming coalitions to solve single multi-robot tasks. This method defines basic building blocks of robot capabilities to be collections of environmental sensors (ESs), perceptual schemas (PSs), motor schemas (MSs), and communication schemas (CSs). A robot, R_i, can be represented by $R_i = (ES^i, S^i)$, where ES^i is a set of environmental sensors installed on R_i, and S^i is the set of schemas that are pre-programmed into R_i. According to a set of rules, connections are created among the schemas on the robots to allow information to flow through the system. A set of information types $F = \{F_1, F_2, ...\}$ is introduced to label the inputs and outputs of each schema. (Information types differ from data types (e.g., the data type *integer*) in that they have semantic meanings (e.g., a robot's global position)). A schema can be activated if its inputs are satisfied either by sensors or the outputs of other schemas with the same information types. The ultimate goal is to activate the required MSs on the robot coalition team members to accomplish the task.

For reasoning about coalitions, ASyMTRe uses an anytime algorithm to search the entire solution space and return the best solution found so far according to predefined cost measures. One of the most important contributions of ASyMTRe is that it enables a finer resource sharing by dividing robot capabilities into smaller chunks (i.e., schemas), and reasons about how these schemas can be connected. Information can flow through the system to where it is required such that capability sharing is implicitly enabled through communication. ASyMTRe effectively manages the search space by reducing the solution space to an equivalence class that is smaller in practice, although in theory it can still be of exponential size. It also orders the search through the solution space via a *maximum cooperation size* constraint, preferring smaller-sized coalition solutions over larger ones. These techniques enable ASyMTRe to quickly find good (although not optimal) solutions that work well in practice.

ASyMTRe has been proven to be sound and complete, and has been shown to provide more flexibility for achieving tightly-coupled multi-robot tasks. Figure 2 illustrates ASyMTRe performing dynamic coalition formation for a cooperative navigation task. In this example, a coalition is formed with two robots to reach a goal position, but during task execution, a failure of one of the robots occurs. This causes ASyMTRe to search for an alternative solution, which is found, resulting in a new coalition of robots.

Extensions to ASyMTRe have also been developed [46] that introduce an information quality based approach to model sensor constraints explicitly, and to provide a general method for maintaining the constraints during the task execution.

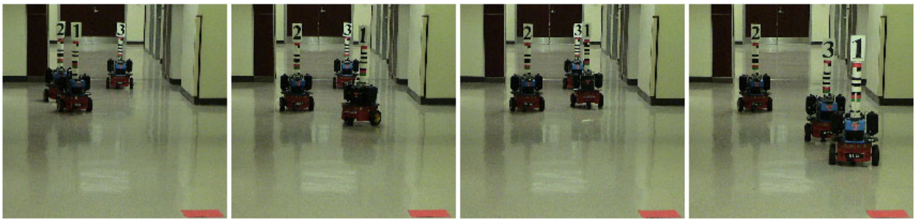


Fig. 2. Illustration of ASyMTRe dynamically forming coalitions for a cooperative navigation task. From [34].

5 Optimization in Path Planning

Another common multi-robot coordination problem is that of planning paths of multiple robots through a workspace. This type of challenge arises most commonly in *coordinative* types of multi-robot teams. Typically, these problems require individual robots to move appropriately through the workspace in order to achieve their own individual task objectives. Ideally, each robot moves as optimally as possible, but it is constrained by other robots also moving in the same workspace. Thus, the robots must work together to ensure that interference is minimized while individual paths are optimized, to the extent possible.

This challenge is also frequently formulated as an optimization problem. As described in [35], the multi-robot path planning can be formulated as an optimization problem as follows (using the notation of [22,23]):

Definition 2. *Let $\mathcal{A}$ be a rigid robot in a static workspace $\mathcal{W} = \mathbb{R}^k$, where $k = 2$ or $k = 3$. The workspace is populated with obstacles. A configuration $\mathbf{q}$ is a complete specification of the location of every point on the robot geometry. The configuration space $\mathcal{C}$ represents the set of all the possible configurations of $\mathcal{A}$ with respect to $\mathcal{W}$. Let $\mathcal{O} \subset \mathcal{W}$ represent the region within the workspace populated by obstacles. Let the closed set $\mathcal{A}(\mathbf{q}) \subset \mathcal{W}$ denote the set of points occupied by the robot when it is in the configuration $\mathbf{q} \in \mathcal{C}$. Then, the C-space obstacle region, C_{obs}, is defined as: $\mathcal{C}_{obs} = \{\mathbf{q} \in \mathcal{C} | \mathcal{A}(\mathbf{q}) \cap \mathcal{O} \neq \emptyset\}$. The set of configurations that avoid collision (called the free space) is: $\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}$. A free path between two obstacle-free configurations c_{init} and c_{goal} is a continuous map: $\tau[0,1] \rightarrow \mathcal{C}_{free}$ such that $\tau(0) = c_{init}$ and $\tau(1) = c_{goal}$.*

For a team of m robots, define a state space that considers the configurations of all the robots simultaneously: $X = \mathcal{C}^1 \times \mathcal{C}^2 \times \cdots \times \mathcal{C}^m$. The C-space obstacle region must now be redefined as a combination of the configurations leading to a robot-obstacle collision, together with the configurations leading to robot-robot collision. The subset of X corresponding to robot $\mathcal{A}^i$ in collision with the obstacle region, $\mathcal{O}$, is $X^i_{obs} = \{\mathbf{x} \in X | \mathcal{A}^i(\mathbf{q}^i) \cap \mathcal{O} \neq \emptyset\}$. The subset of X corresponding to robot $\mathcal{A}^i$ in collision with robot $\mathcal{A}^j$ is $X^{ij}_{obs} = \{\mathbf{x} \in X | \mathcal{A}^i(\mathbf{q}^i) \cap \mathcal{A}^j(\mathbf{q}^i) \neq \emptyset\}$.

The obstacle region in X is then defined as the combination of these latter two equations, resulting in:

$$X_{obs} = \left(\bigcup_{i=1}^{m} X^{i}_{obs}\right) \bigcup \left(\bigcup_{ij, i \neq j} X^{i}_{obs}\right). \tag{1}$$

The planning process for multi-robot systems treats X the same as $\mathcal{C}$, and X_{obs} the same as $\mathcal{C}_{obs}$, where c_{init} represents the starting configurations of all the robots, and c_{goal} represents the desired goal configurations of all the robots.

The optimization criteria that are typically used in multi-robot path planning are the minimization of total robot path lengths, the minimization of time for all robots to reach their goals, and the minimization of combined energy for robots to reach their goals. Other constraints can be added to guide the solution search, such as the incorporation of navigation restrictions (e.g., maximum slope, inability to traverse rough terrains).

As with other formulations, the centralized, global optimization problem cannot be solved in real-time. Thus, approximation techniques are used, such as decoupling the planning problem into independent components. One common decoupled approach is to divide the problem into planning independent robot paths, and then coordinating robot velocities along the paths in order to avoid collisions (e.g., [14,16]). Another common technique is *prioritized planning*, in which robot paths are planned in a priority order, with robots later in the order treating robots earlier in the order as moving obstacles (e.g., [3,10]). Figure 3 illustrates example results from the prioritized multi-robot path planning work of [3].

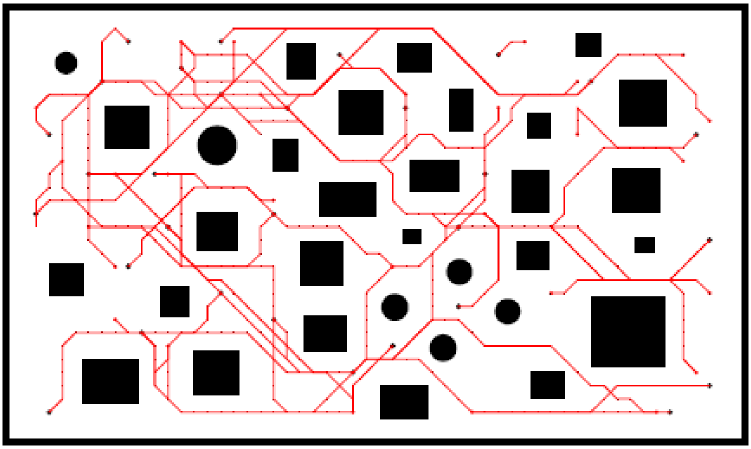


Fig. 3. Illustration of results of prioritized multi-robot path planning for 30 robots. From [3].

To provide more detail regarding an approximation technique to multi-robot path planning, we outline the work of [14], which proposes a decentralized motion planning algorithm for multiple robots. This approach incorporates optimal solutions to subcomponents of the path planning problem. The computationally expensive problem is decomposed into two modules – path planning and velocity planning. Each robot plans its own path independently using the D^* search algorithm [39], which it then broadcasts to all other robots. The D^* search algorithm produces an optimal path from the start position to the goal that minimizes a pre-defined cost function. The cost function used in this work is: $f_{pp} = \rho + \alpha_1 d + \alpha_2 s + \alpha_3 t$, where ρ is a large value if there is any obstacle penetrated by the path, and 0 otherwise; d is the geometric distance; s is the slope of the terrain; t is the penalty for turning; and $\alpha_1, \alpha_2, \alpha_3$ are positive weighting factors, where $\|(\alpha_1, \alpha_2, \alpha_3)\| = 1$. Such a cost function guarantees that D^* returns an optimal path that avoids static obstacles, and is the shortest, flattest, smoothest possible path if one exists.

After robot R_i obtains its own path P_i and all other paths $P_j, (j = 1, 2, \ldots, N)$, $j \neq i$, it executes a collision check procedure, which returns all collision regions. Since the configuration space is on a regular grid representation, the collision region is represented by sets of (x, y) pairs at which path intersections occur. Each path P_i can be seen as a continuous mapping $[0, l] \rightarrow \mathcal{W}^{\varepsilon}_{free}$, where l is the path length. Without loss of generality, one can assume that the parameterization of P_i is of constant speed. Then, define $\mathcal{S}_i = [0, l]$ to denote the set of parameter values that place the robot along the path P_i. The path coordination space is defined as $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_N$, and the *coordination diagram* (CD) is an N-dimensional diagram representing the path coordination space.

D^* then searches for a free trajectory in the CD by first mapping the collision regions into the path coordination space as static obstacles. As the path coordination space is parameterized by the non-decreasing path length, any possible movement in CD should be non-decreasing. Thus, the search objective is to find a non-decreasing curve that connects the lower left corner of the diagram $(0, 0, \ldots, 0)$ to the top right corner $(l_1, l_2, \ldots, l_N)$ avoiding penetration into the static obstacles. Such a free curve is called a *trajectory.* The computational expense is reduced by the non-decreasing constraint of the search. At each grid point, $2^N - 1$ action combinations are considered. Although the complexity is still exponential in the number of robots, the algorithm is efficient for a fixed N.

The trajectory is then converted into a velocity profile for each robot, and the performance index of the current trajectory solution is calculated. Since searching in the CD is distributed across the robots, each search can minimize a different cost function. The cost function for D^* velocity planning is chosen to be: $f_{vp} = \varrho + \beta_1 d + \beta_2 t_{idle} + \beta_3 p$, where ϱ is a large value if there are any collision regions penetrated by the trajectory, and 0 otherwise; d is the N-dimensional Euclidean distance; t_{idle} is the total idle time for all robots; p is the penalty if robot $\mathcal{A}_i$ has to give way to others; and $\beta_1, \beta_2, \beta_3$ are positive weighting factors, where $\|(\beta_1, \beta_2, \beta_3)\| = 1$.

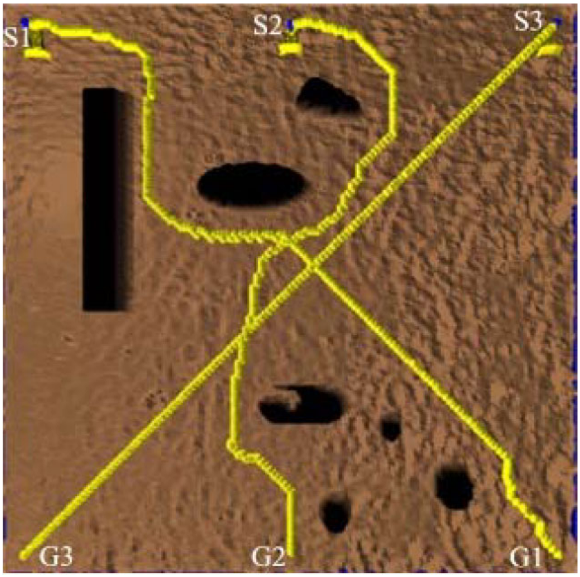


Fig. 4. Illustration of multi-robot path planning technique, which decouples path planning and velocity planning. Shown are the planned paths for three robots. From [14].

The performance index and velocity profile are then broadcast to all other robots. An evaluation is performed to obtain a minimum value of the performance index, and the corresponding velocity profile is chosen. Figure 4 gives example results from this technique for three robots.

6 Optimization in Formation Control

The *formation control* problem in multi-robot systems addresses the challenge of moving robots into a desired formation shape and/or having the robots move in a coordinated manner while maintaining a desired shape. This problem arises in *cooperative* multi-robot teams, since all the robots share the same objective of maintaining the specified formation. Many researchers address this problem from control theoretic principles, focusing especially on proving stability and convergence properties (e.g., [2,11,12,41]). The problem of generating an initial desired formation can be formulated in many ways as an optimization problem. For example, the work of [8] presents the formation problem as a convex optimization problem, showing that certain forms can be solved in real time for large-scale multi-robot teams. This formulation of [8] is as follows:

Definition 3. *Let $P = (p_1, \ldots, p_m)^T$ denote the concatenated coordinates of the m robots in their current pose. Let S represent the equivalence class of similarity transformations of the desired formation. The objective is to obtain a new formation pose $Q = (q_1, \ldots, q_M)^T$, where Q has the same shape as S under* *an equivalence relation, and either the maximum distance between the respective positions in P and Q are minimized, or the sum of the distances is minimized. The set of nonlinear equality constraints that formulate this problem is given by:*

$$\|s_2\|_2(q_i^x - q_1^x) = (s_i^x, -s_i^y)^T(q_2 - q_1)$$
$$\|s_2\|_2(q_i^y - q_1^y) = (s_i^y, -s_i^x)^T(q_2 - q_1)$$

for $i = 3, \ldots, m$. For the sum of the distances optimization criteria, the constrained optimization problem becomes: $min_q \sum_{i=1}^{m} \|q_i - p_i\|_2$, subject to $Aq = 0$.

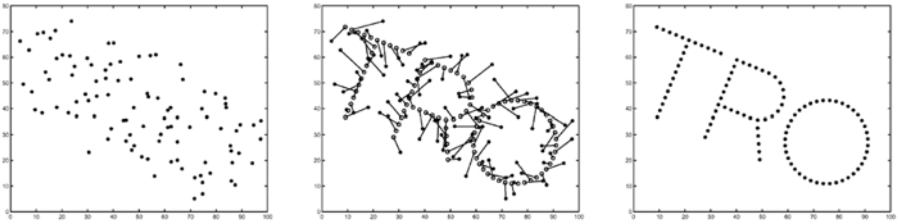


Fig. 5. Illustration of multi-robot formation generation. From [8].

The authors in [8] illustrate this technique for 100 simulated robots, as shown in Figure 5.

7 Optimization in Target Tracking/Observation

The domain of multi-target tracking and observation requires multiple robots to observe multiple targets moving through the environment. The objective is to keep as many of the targets within view by some robot on the team. This is a problem for *cooperative* and *collaborative* multi-robot teams. As pointed out in [31], this task is useful for studying *strong cooperation* in multi-robot teams, since the actions of each robot directly affect the performance of the others on the observation task.

The two-dimensional (planar) version of this task was introduced in [31] as the CMOMMT (Cooperative Multi-robot Observation of Multiple Moving Targets) problem. CMOMMT is formulated as an optimization problem as follows:

Definition 4. *Define the following: $\mathcal{S}$ is a two-dimensional, bounded, enclosed spatial region; $\mathcal{V}$ is a team of m robot vehicles $v_i, i = 1, 2, ...m$; $\mathcal{O}(t)$ is a set of n targets $o_j(t)$, $j = 1, 2, ..., n$, such that target $o_j(t)$ is located within region $\mathcal{S}$ at time t. We say that a robot, v_i, is observing a target when the target is within v_i's sensing range. Define an $m \times n$ matrix $B(t)$, as:*

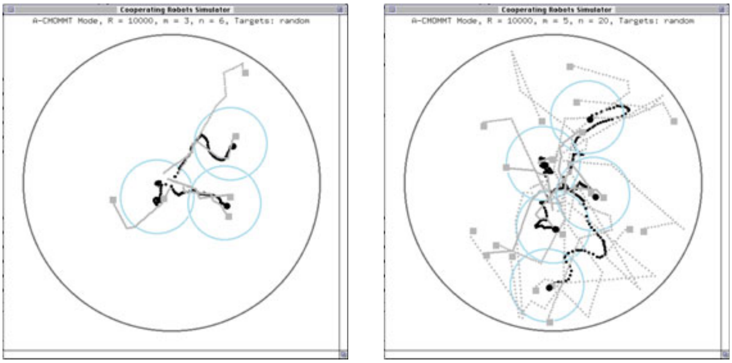
$$B(t) = [b_{ij}(t)]_{mxn} \text{ such that } b_{ij}(t) = \begin{cases} 1 \textit{ if robot } v_i \textit{ is observing target } o_j(t) \\ \quad \textit{in } \mathcal{S} \textit{ at time } t \\ 0 \textit{ otherwise} \end{cases}$$

Then, the objective is to maximize the metric $A = \sum_{t=1}^{T} \sum_{j=1}^{n} \frac{g(B(t),j)}{T}$, *where:*

$$g(B(t),j) = \begin{cases} 1 \textit{ if there exists an } i \textit{ such that } b_{ij}(t) = 1 \\ 0 \textit{ otherwise} \end{cases}$$

Similar problems have been studied by many researchers, including more complex versions in three dimensions (e.g., for aerial vehicles) and with more complex topography. This domain is related to problems in art gallery algorithms, pursuit evasion, and sensor coverage, and has practical relevance in security and surveillance applications. Example research in this domain includes [1,19,24,25].

For the multi-robot target observation problem, the approximation solution that is proposed in [31] is a weighted local force vector approach that attracts robots to nearby targets and repels them from nearby robots. The weights are computed in real-time, and are based on the relative locations of the nearby robots and targets. In this approach, each robot broadcasts to its teammates the position of all targets within its field of view. For all known targets, robots then perform a predictive tracking of that target's location, assuming that the target will continue linearly from its current state. Weights associated with known targets are decreased if other robots are known to be nearby. Setting the weights in this manner aims at generating an improved collective behavior across robots when utilized by all robot team members. Example results from this approach are illustrated in Figure 6.



A-CMOMMT Mode, R = 10000, m = 3, n = 6, Targets: random

A-CMOMMT Mode, R = 10000, m = 5, n = 20, Targets: random

Fig. 6. Illustration of simulation results of multi-robot target observation, with (left) 3 robots and 6 targets, and (right) 5 robots and 20 targets. From [31].

8 Conclusions

While not all multi-robot systems make use of decision making, it is common in more intentional types of interaction that involve cooperative, collaborative, and coordinative multi-robot teams. These types of teams are applied to a wide variety of applications, many of which involve common multi-robot capabilities such as task allocation, path planning, formation control, and target tracking/observation. Researchers often formulate these common capabilities by defining them as optimization problems. In this paper, we have discussed some common formulations for the four main multi-robot capability areas, and discussed some examples of approximation algorithms that are guided by the optimization formulation. While exact global solutions are not possible due to the intractability of the formulations, much research has shown that the approximate techniques work well in practice. These results show, therefore, that much of multi-robot decision making can be successfully viewed, and approximated, as optimization problems. The further development and application of distributed, approximate solutions to optimization problems is therefore expected to be beneficial for generating more effective decision making in multi-robot teams.

References

1. Beard, R.W., McLain, T.W., Goodrich, M.: Coordinated target assignment and intercept for unmanned air vehicles. In: Proceedings of IEEE International Conference on Robotics and Automation. IEEE (2002)
2. Belta, C., Kumar, V.: Abstraction and control for groups of robots. IEEE Transactions on Robotics 20(5), 865–875 (2004)
3. Bennewitz, M., Burgard, W., Thrun, S.: Optimizing schedules for prioritized path planning of multi-robot systems. In: Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2001, vol. 1, pp. 271–276 (2001)
4. Bertsekas, D.: The auction algorithm for assignment and other network flow problems: A tutorial. Interfaces, 133–149 (1990)
5. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge Univ. Pr. (2004)
6. Browning, B., Bruce, J., Bowling, M., Veloso, M.: STP: Skills, tactics and plays for multi-robot control in adversarial environments. IEEE Journal of Control and Systems Engineering 219, 33–52 (2005)
7. Chvatal, V.: A greedy heuristic for the set-covering problem. Mathematics of Operations Research, 233–235 (1979)
8. Derenick, J., Spletzer, J.: Convex optimization strategies for coordinating large-scale robot formations. IEEE Transactions on Robotics 23(6), 1252–1259 (2007)
9. Dias, M., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: A survey and analysis. Proceedings of the IEEE 94(7), 1257–1270 (2006)
10. Erdmann, M., Lozano-Perez, T.: On multiple moving objects. Algorithmica 2, 477–521 (1987)
11. Fax, J.A., Murray, R.M.: Information flow and cooperative control of vehicle formations. IEEE Transactions on Automatic Control 49(9) (2004)
12. Ge, S.S., Fua, C.H.: Queues and artificial potential trenches for multirobot formations. IEEE Transactions on Robotics 21(4), 646–656 (2005)
13. Gerkey, B., Mataric, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. International Journal of Robotics Research 23(9), 939–954 (2004)
14. Guo, Y., Parker, L.E.: A distributed and optimal motion planning approach for multiple mobile robots. In: Proceedings of IEEE International Conference on Robotics and Automation (2002)
15. Hoffman, K., Padberg, M.: Solving airline crew scheduling problems by branch-and-cut. Management Science, 657–682 (1993)

16. Kant, K., Zucker, S.W.: Toward efficient trajectory planning: the path-velocity decomposition. The International Journal of Robotics Research 5(3), 72–89 (1986)
17. Kitano, H., Kuniyoshi, Y., Noda, I., Asada, M., Matsubara, H., Osawa, E.: Robocup: A challenge problem for AI. AI Magazine 18(1), 73–85 (1997)
18. Klusch, M., Gerber, A.: Dynamic coalition formation among rational agents. IEEE Intelligent Systems 17(3), 42–47 (2002)
19. Kolling, A., Carpin, S.: Multirobot cooperation for surveillance of multiple moving targets – a new behavioral approach. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1311–1316. IEEE (2006)
20. Kube, C.R., Zhang, H.: Collective robotics: From social insects to robots. Adaptive Behavior 2(2), 189–219 (1993)
21. Kuhn, H.: The hungarian method for the assignment problem. Naval research logistics quarterly 2(1-2), 83–97 (1955)
22. Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Norwell (1991)
23. LaValle, S.M.: Planning Algorithms. Cambridge University Press (2006)
24. LaValle, S.M., Gonzalez-Banos, H.H., Becker, C., Latombe, J.C.: Motion strategies for maintaining visibility of a moving target. In: Proceedings of the 1997 IEEE International Conference on Robotics and Automation, pp. 731–736. IEEE (1997)
25. Luke, S., Sullivan, K., Panait, L., Balan, G.: Tunably decentralized algorithms for cooperative target observation. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 911–917. ACM Press (2005)
26. Mataric, M., Nilsson, M., Simsarian, K.: Cooperative multi-robot box pushing. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 556–561 (1995)
27. McLurkin, J.: Stupid robot tricks: Behavior-based distributed algorithm library for programming swarms of robots. M.S. Thesis, Massachusetts Institute of Technology (2004)
28. Michael, N., Zavlanos, M., Kumar, V., Pappas, G.: Distributed multi-robot task assignment and formation control. In: IEEE International Conference on Robotics and Automation, ICRA 2008, pp. 128–133 (2008)
29. Nemhauser, G., Wolsey, L.: Integer and combinatorial optimization, vol. 18. Wiley, New York (1988)
30. Parker, L.E.: ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. IEEE Transactions on Robotics and Automation 14(2), 220–240 (1998)
31. Parker, L.E.: Distributed algorithms for multi-robot observation of multiple moving targets. Autonomous Robots 12(3), 231–255 (2002)
32. Parker, L.E.: Chapter 40: Multiple mobile robot systems. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics. Springer, Heidelberg (2008)
33. Parker, L.E.: Distributed intelligence: Overview of the field and its application to multi-robot systems. Journal of Physical Agents 2(1), 5–14 (2008)
34. Parker, L.E., Tang, F.: Building multi-robot coalitions through automated task solution synthesis. Proceedings of the IEEE, special issue on Multi-Robot Systems 94(7), 1289–1305 (2006)
35. Parker, L.: Path planning and motion coordination in multiple mobile robot teams. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and System Science. Springer, Heidelberg (2009)
36. Sandholm, T., Larson, K., Andersson, M., Shehory, O., Tohme, F.: Coalition structure generation with worst case guarantees. Artificial Intelligence 111(1-2), 209–238 (1999)

37. Shehory, O.: Methods for task allocation via agent coalition formation. Artificial Intelligence 101(1-2), 165–200 (1998)
38. Simmons, R., Singh, S., Hershberger, D., Ramos, J., Smith, T.: First results in the coordination of heterogeneous robots for large-scale assembly. In: Proc. of the ISER 2000 Seventh International Symposium on Experimental Robotics (2000)
39. Stentz, A.: Optimal and efficient path planning for unknown and dynamic environments. International Journal of Robotics and Automation 10, 89–100 (1993)
40. Sukhatme, G., Montgomery, J.F., Vaughan, R.T.: Experiments with cooperative aerial-ground robots. In: Balch, T., Parker, L.E. (eds.) Robot Teams: From Diversity to Polymorphism, pp. 345–368. A K Peters (2002)
41. Tabuada, P., Pappas, G., Lima, P.: Motion feasibility of multi-agent formations. IEEE Transactions on Robotics 21(3), 387–392 (2005)
42. Vig, L., Adams, J.A.: Issues in multi-robot coalition formation. In: Parker, L.E., Schultz, A., Schneider, F. (eds.) Multi-Robot Systems. From Swarms to Intelligent Automata, vol. III. Kluwer (2005)
43. Vig, L., Adams, J.A.: Multi-robot coalition formation. IEEE Transactions on Robotics 22(4), 637–649 (2006)
44. Weigel, T., Gutmann, J.S., Dietl, M., Kleiner, A., Nebel, B.: CS Freiburg: coordinating robots for successful soccer playing. IEEE Transactions on Robotics and Automation 5(18), 685–699 (2002)
45. Wurman, P., D'Andrea, R., Mountz, M.: Coordinating hundreds of cooperative, autonomous vehicles in warehouses. AI Magazine 29(1), 9 (2008)
46. Zhang, Y., Parker, L.: Iq-asymtre: Synthesizing coalition formation and execution for tightly-coupled multirobot tasks. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5595–5602 (2010)
47. Zlot, R., Stentz, A.: Complex task allocation for multiple robots. In: Proceedings of IEEE International Conference on Robotics and Automation (2005)

Mstar : A New Two Level Interconnection Network

Nibedita Adhikari[1,*] and C.R. Tripathy[2]

[1] Department of CSE, PIET, Rourkela, Orissa, India
[2] Department of CSE, VSS University of Technology, Burla
Sambalpur, Orissa, India
head.csepiet@gmail.com

Abstract. In the literature various two level interconnection networks are proposed using hypercubes or star graphs. In this paper, a new two level interconnection network topology called the Metastar denoted as Mstar(k,m) is introduced. The proposed network takes the star graph as basic building blocks. Here, the network at the lower level is a star but at the higher level the network is a cube. Its various topological parameters such as packing density, degree, diameter, cost, average distance and hamiltonicity are investigated. Message routing and broadcasting algorithms are also proposed. Performance analysis in terms of topological parameters is done and the proposed network is proved to be a suitable candidate for large scale computing.

Keywords: Parallel processing, interconnection network, topological parameters.

1 Introduction

The rapid development in technology has resulted in development of systems incorporating a very large number of processors. As the number of processors in the system increases, the performance of such systems is faced with new challenges. The performance of any multiprocessor system mainly depends upon the communication efficiency of the underlying interconnection topology [7,11]. Numerous interconnection structures have been introduced in literature. Among the wide variety of interconnection structures, the cube based structures are most popular [4,8,15]. The strong connectivity, regularity, symmetry, embeddability, logarithmic diameter, partitionability, fault tolerance and simple routing. However the number of edges per node increases logarithmically as the total number of nodes or the packing density increases, thereby increasing the complexity of the entire system. For this reason there are several architectures suggested in the recent past. The prominent and recent networks are Crossed cube [5], Cube connected cycles [3], Hierarchical cubic networks [6], Dual cube [13], Metacube [14] and also Cayley graphs. These networks are introduced with some modifications in their properties in order to improve the overall performance. Investigations are also going on to study two level networks. Hierarchical Folded hypercube network, HFN(n,n) is a two level network using folded hypercubes as basic modules. The node degree is (n+2) and the packing density is 2^{2n} [16].

[*] Corresponding author.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 50–61, 2012.

One of the important class of Cayley graph, called the Star graph has been popular as an alternative to Hypercube. The Star network is an extensively studied Cayley Graph, considered to be an attractive alternative to the n-cube [10]. It is a node symmetric and edge symmetric graph consisting of $n!$ number of nodes and $n!\,(n-1)/2$ number of edges. Some of the important features of Star graph are fault tolerance, partitionability, node disjoint paths and easy routing and broadcasting. Inspite of these attractive features, the Star network has a major disadvantage. It grows to its next higher dimension by a large value. Another alternative of the Star called the Incomplete star has been introduced to eliminate this problem [9]. But the Incomplete star is a non symmetric and irregular graph. So it is not suitable to use in many practical systems.

The Metacube is a very large scale parallel interconnection network which can connect to millions of nodes with a fixed node degree [14]. The network is defined using two parameters namely k and m. The parameter k can take any value starting from 1 and m can take any value starting from 2. The smallest possible network that is MC(1,2), also called the Dualcube [13] contains 32 nodes with diameter 6. The diameter of this very large scale network has been further reduced and a new network has been introduced as Folded dualcube [1]. As a result the cost of the network is also reduced.

Recently Star-cube network a variation of Star graph is introduced in [2]. The Star-cube also known as Cube-star is a hybrid network. The Star-cube is regular, vertex, edge-symmetric, maximally fault tolerant and cost effective. When compared with Star, the growth of Star-cube is comparatively small. The smallest possible structure contains 24 nodes with node degree 4. But this network is not a very large scale network like the Metacube network as the MC network contains 32 nodes with the node degree as 3.

Another variation of the Star graph called the Hierarchical star network, HS(n,n) is introduced as a two level interconnection network [12]. The HS(n,n) network consists of $n!$ modules where each module is a Star graph. So the HS network contains $(n!)^2$ nodes with node degree n. The modules are interconnected with additional edges. The size of the network grows at a very high rate. When n is 3 the network size is 36 but when n is 4, the network contains 576 nodes. This significant gap in the two consecutive sizes of Hierarchical Star becomes a major disadvantage. Another disadvantage of Hierarchical star is that the dimension cannot take any values of n like Metacube. It only takes values like (3,3), (4,4), (5,5) etc. Keeping all these points in view the current paper proposes a new two level structure suitable for massive parallel systems.

The proposed Metastar network is a hybrid network. Here the basic building block is an n-Star. As discussed in Metacube network topology [14], the Metastar network contains 2^k classes. Each class in turn contains m! number of clusters and each cluster will contain again m! number of nodes as each cluster is an m-Star. Thus there are $2^k m!\, m!$ number of nodes. There is no direct link between clusters of same class.

The paper is organized as follows: the detail architecture is given in Section 2. The topological properties of the Metastar network are derived in Section 3. Routing and broadcasting algorithms are also described for the new network in Section 4.

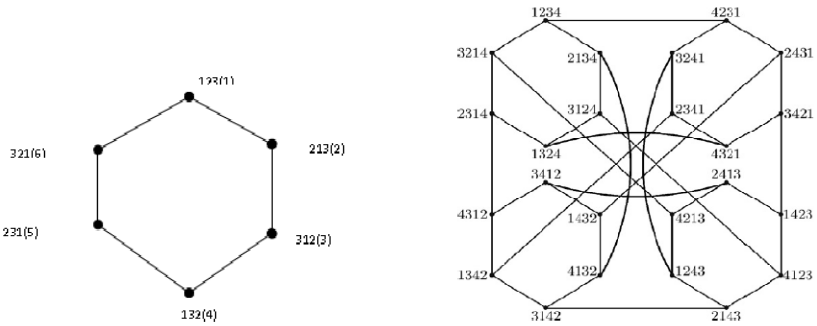
Performance analysis for Metastar is presented in Section 5. At last Section 6 presents the concluding remarks.

2 Proposed Architecture

This section first describes the Star graph and the Metacube networks. Next it presents the details of the new proposed architecture the Metastar network along with possible connections, neighbours and other details.

2.1 Star Graph

The n-dimensional Star graph, S_n ,is an edge- and node-symmetric graph containing n! nodes and (n-1)n!/2 number of edges. Each node in an n-Star has (n-1) incident edges. The nodes are assigned labels each being a distinct permutation of the set of integers {1,2,...n}. Two permutations are adjacent if and only if one of them can be obtained from the other by exchanging the symbols in position 1 and i, for some *i 1*. The Figure 1 below shows the Star graph of dimension 3 and 4.



(a) (b)

Fig. 1. Star graph of dimension 3 and 4 (a) 3-Star, (b) 4-Star

2.2 Metacube Network

The Metacube is a very large scale parallel system having two parameters k and m. The Metacube MC(k,m) contains 2^k classes and each class contains $2^{m(2^k-1)}$ clusters and each cluster contains 2^m nodes. Therefore the Metacube network uses $m2^k + k$ binary bits to identify a node. Thus the total number of nodes is 2^n , where $n = m2^k + k$. The node degree of MC network is *(m+k).* The MC network has a two level cube structure. The low level cluster is a hypercube. The nodes of a cluster are connected to nodes of a cluster in another class. There is no link between the nodes that belong to different clusters but same class. Thus the high level structure is also a cube. The Metacube of dimension 3 and 4 are shown below in Fig. 2 and 3.

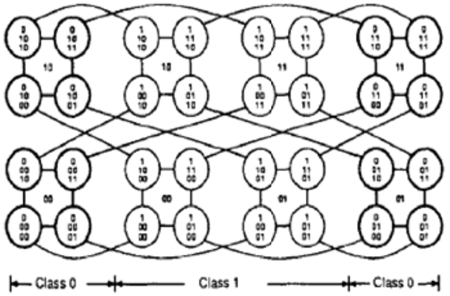


Fig. 2. Metacube network of dimension 3 MC(1,2)

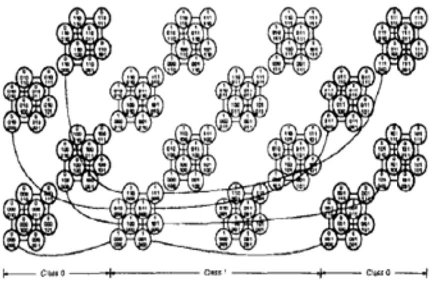


Fig. 3. Metacube of dimension 4, MC(1,3)

2.3 Metastar Network

This section proposes a new interconnection network called the Metastar network suitable for large scale parallel systems. The new network retains the efficient features of both the Metacube as well as the Star graph. Like MC network it can connect to millions of nodes with a quite small node degree as compared to that of MC network. Here each cluster is a Star graph of dimension 3 or more.

The Metastar network contains 2^k classes and each class contains *m!* number of culsters and each cluster then contains *m!* nodes, as each cluster is a Star graph. The Fig. 5 and 6 show the Metastar of dimension 2 and 3. The address of a node in the Metastar network has three parts as shown below in Fig. 4. The leftmost k-bits in binary represent the class address. Next m bit permutation represents the cluster and the right most m bit permutation represents the node within the cluster. Alternatively decimal notation can be used to reduce the complexity in the diagram. So the nodes in 3-Star can have labels 1,2,3,4,5 and 6 as shown in Fig. 1 (a). Figure 7 shows two individual clusters of Metastar(1,3) with their node addresses in equivalent decimal notations.

Given a node *v(x,y,z)* of Metastar(k,m), *x* is the *k* bit class label, *y* is the m bit permutation for cluster label and *z* is the m bit permutation for node label. So *v* can have following nodes as neighbour :

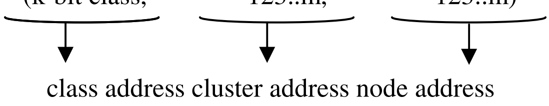


Fig. 4. Format of node address for Metastar network

$(x_{k-1}x_{k-2}\ldots\bar{x}_i..x_0, y_1\, y_2\ldots y_m, z_1z_2,..,z_m)$ when $y_i = z_i$ for all values of i, $(x_{k-1}x_{k-2}\ldots\bar{x}_i.x_0, z_1, z_2\ldots, z_m, y_1\, y_2\ldots y_{m,})$ and $(x_{k-1}\, x_{k-2}\ldots x_0, y_1y_2..y_m, \boldsymbol{Z}_i z_2..z_1..z_m)$.

The third type neighbour belongs to the same cluster. As per the Metacube properties, two nodes belonging to same class but different cluster are not connected. So to calculate the distance between such nodes two extra edges will be needed one for entering the other class and one for leaving the class. When the nodes belong to same cluster the distance is calculated as per the Star graph terminology depending upon the permutations in the node address.

2.4 Illustration

In case of Metastar(1,3), k=1,m=3. So each node will have (k+m-1) neighbours out of which k neighbours will be in other class and (m-1) neighbours in the same cluster. Metastar(1,3) contains 2 classes and each class contains 6 clusters. Each cluster in turn contains 6 nodes. Thus there are 36 nodes in each class and in total there are 72 nodes with node degree 3. So for node (0,123,123) the neighbours will be (1,123,123), (0,123, 213) and (0,123,321). In alternate notations as discussed above the node (0,1,1) will have neighbours (1,1,1), (0,1,2) and (0,1,6) as shown in Fig. 7. Out of these three neighbours, the first one belongs to the other class and the rest two belong to the same cluster of the same class. Neighbours in Metastar(1,3) : The following nodes are the neighbours through cross links in the Metastar(1,3).

(0,1,1)-(1,1,1)	(0,2,1)-(1,1,2)	(0,3,1)--(1,1,3)	(0,4,1)--(1,1,4)	(0,5,1)--(1,1,5)	(0,6,1)--(1,1,6)
(0,1,2)--(1,2,1)	(0,2,2)-(1,2,2)	(0,3,2)--1,2,3)	(0,4,2)--(1,2,4)	(0,5,2)--(1,2,5)	(0,6,2)--(1,2,6)
(0,1,3)--(1,3,1)	(0,2,3)--(1,3,2)	(0,3,3)--(1,3,3)	(0,4,3)--(1,3,4)	(0,5,3)--(1,3,5)	(0,6,3)--(1,3,6)
(0,1,4)--(1,4,1)	(0,2,4)--(1,4,2)	(0,3,4)--(1,4,3)	(0,4,4)--(1,4,4)	(0,5,4)--(1,4,5)	(0,6,4)--(1,4,6)
(0,1,5)--(1,5,1)	(0,2,5)--(1,5,2)	(0,3,5)--(1,5,3)	(0,4,5)--(1,5,4)	(0,5,5)--(1,5,5)	(0,6,5)--(1,5,6)
(0,1,6)--(1,6,1)	(0,2,6)--(1,6,2)	(0,3,6)--(1,6,3)	(0,4,6)--(1,6,4)	(0,5,6)--(1,6,5)	(0,6,6)--(1,6,6)

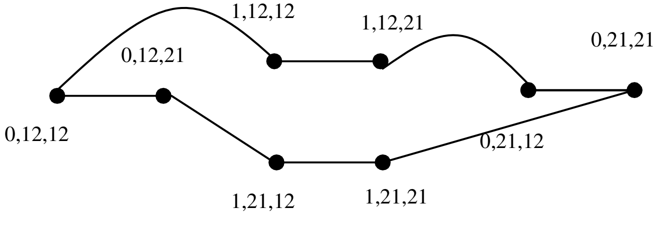


Fig. 5. Metastar network of dimension 3, Mstar(1,2)

3 Topological Properties

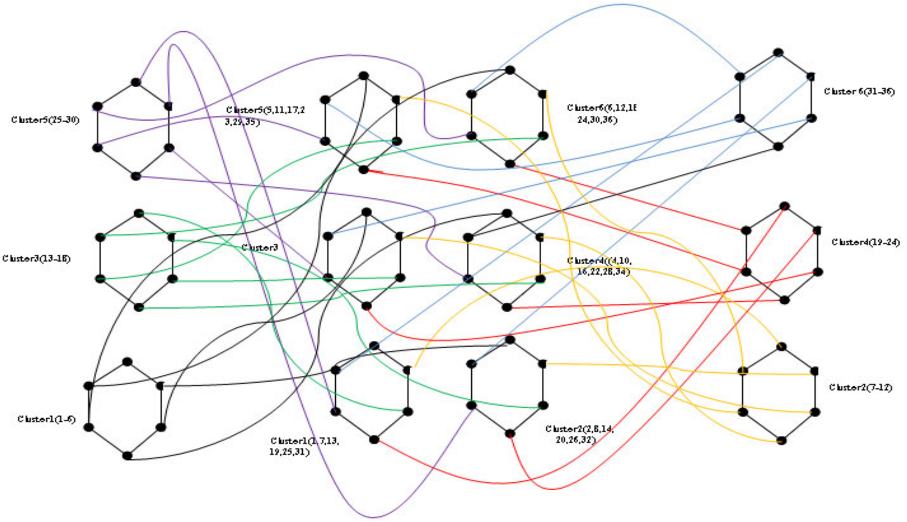
This section highlights some of the topological properties of the proposed Metastar network.

3.1 Nodes

Theorem 1: The total number of nodes in the Metastar network is given by $p = 2^{k}m!\,m!$.

Proof: The node address in the Metastar contains three parts namely class address, cluster address and node address. The Metastar(k,m) network contains 2^k classes. Each class contains m! number of clusters. Each cluster contains m! number of nodes. Hence, the total number of nodes is given by

$$p = 2^k m!\, m! \qquad (1)$$



Class 0 Class 1 Class 0

Fig. 6. Metastar network of dimension 3, Mstar(1,3)

3.2 Node Degree

Theorem2: The node degree of the Metastar network is *(k+m-1).*

Proof: In the Metastar network each cluster is an m-Star. So each node in the cluster is having *(m-1)* neighbours. Again the nodes of one cluster in a class are connected to k nodes in clusters of other classes. Nodes that belong to clusters of same class are not connected. Hence for a single node there will be *(k+m-1)* neighbours. Hence the node degree of the Metastar network is *(k+m-1).*

3.3 Number of Links or Edges

Theorem 3: The total number of edges in the Metastar network is given by

$$E = (2^k m!\, m!)*(k+m-1)/2$$

Proof: In the Metastar network the node degree is (k+m-1). The total number of nodes is $p = 2^k m!\, m!$. For cube based networks the number of edges is given by
E= (Number of nodes)* (Node Degree/2).

So for Metastar network the number of edges is given by

$$E = (2^k m!\, m!)*(k+m-1)/2 \tag{2}$$

3.4 Illustration

let k=1 and m=3 then, in the Metastar(1,3) there are 2^1 =2 classes. Each class contains m!=6 clusers and m!=6 nodes in each cluster that is a 3-Star. So the total number of nodes is given by

P=$2^1 * 3! * 3! = 72$ and the total number of edges is given by

$$E=(2^k m!\, m!)(k+m-1)/2=72*3/2=108$$

(0,1,1) (0,1,2) (0,1,3) (0,1,4) (0,1,5) (0,1,6) are the addresses of 6 nodes of cluster 1 of class 0 as shown in Fig. 7.

3.5 Diameter

The diameter is the maximum of the shortest distance between any two nodes of a network over all pairs of nodes.

Theorem4: The diameter of the Metastar network is given by D = $\left(\left\lfloor\frac{3(m-1)}{2}\right\rfloor + 1\right) 2^k$.

Proof: In Metastar the higher level structure is a cube where as the lower level structure is an m-Star. In the higher level there are 2^k number of classes. The maximum distance will be decided using two facts, namely a) number of hops within the cluster and b) number of hops between the classes. The length of the path covering the high level cube is 2^k. Next travelling within the cluster requires $\left\lfloor\frac{3(m-1)}{2}\right\rfloor$ steps as it is a star graph. So the diameter of the Metastar network is given by

$$\mathrm{D}= 2^k + (\left\lfloor\frac{3(m-2)}{2}\right\rfloor)2^k = 2^k(1 + \left\lfloor\frac{3(m-1)}{2}\right\rfloor) \tag{3}$$

Hence the result.

3.6 Cost

Theorem 5: The cost of the Metastar network is given by $\xi = \left(\left(\left\lfloor\frac{3(m-1)}{2}\right\rfloor + 1\right) * 2^k(\mathrm{k+m-1})\right)$

Proof: From Theorem 2, the node degree of the Metastar network is (k+m-1). From Theorem 4 , the diameter of the Metastar is $\left(\left\lfloor\frac{3(m-1)}{2}\right\rfloor + 1\right) 2^k$. As the cost of a network is the product of degree and diameter, hence the cost of the Metastar network is given by

$$\xi = \text{diameter * degree}$$

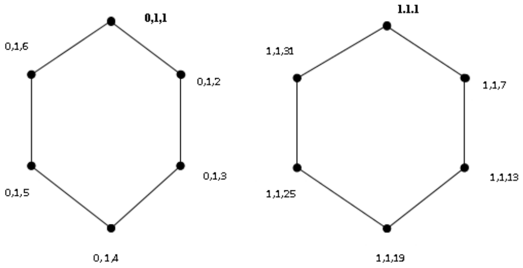
$$= \left(\left(\left\lfloor\frac{3(m-1)}{2}\right\rfloor + 1\right) * 2^k * (\mathrm{k+m-1})\right) \tag{4}$$

3.7 Bisection Width

The bisection width of a parallel interconnection network topology is defined as the total number of edges whose removal will result in two distinct sub networks. The minimum bisection width plays a vital role in measuring the area complexity of VLSI layouts of the network topology.

Theorem 6: The bisection width of the Metastar network is $(m!)^2 * 2^{k-2}$.

Proof: The total number of nodes in the Metastar network is given by Equ. 1. So



(a) (b)

Fig. 7. Clusters of Metastar(1,3) (a)Cluster 1 of Class0 and (b) Cluster 1 of Class1

$$p = 2^k m!\, m!$$

There are 2^k number of classes in total. After bisection the network will be devided into two equal halves suppose *Mstar^0(k,m)* and *Mstar^1(k,m)* such that *Mstar^0(k,m)* will contain half of the clusters of class $i, i = 1,2,..2^k - 1$ and *Mstar^1(k,m)* will contain the rest clusters. Hence, the bisection width of the Mstar(k,m) network is

$$((\frac{2^k m!m!}{2^k})/2) * 2^{k-1}$$

$$=(m!)^2 * 2^{k-2} \tag{5}$$

3.8 Average Node Distance

Theorem 6: The average node distance of the Metastar network is $\bar{D} = (\overline{d_s} + 1)2^k$ where $\bar{d}_s$ is the average node distance of the Star graph.

Proof: The Metastar is a hybrid network and contains n-Star as a cluster. So for moving within the cluster the average distance will be same as that of the star graph that is

$$\bar{d}_s\text{= n-4+}\frac{n}{2} + \sum_{i=1}^{n} \frac{1}{i} \tag{6}$$

Next in the higher level, routing algorithms will traverse to a cluster of each class and there are 2^k classes hence, the average distance of any two nodes in the Mstar(k,m) network is

$$\bar{\mathrm{D}} = \bar{\mathrm{d}}_\mathrm{s} 2^\mathrm{k} + 2^\mathrm{k} \tag{7}$$

4 Routing and Broadcasting

In this section the routing and broadcasting algorithms for the proposed Metastar network are suggested. Before this the Hamiltonian properties are discussed.

4.1 Hamiltonian Properties of Metastar

As discussed in [14], the Metacube is Hamiltonian. Next Star is also Hamiltonian as shown in Fig. 7 and 8. For 3-star the Hamiltonian path will be 1-2-3-4-5-6-1 or 1-6-5-4-3-2-1. The path length is (n!-1) =5. For 4-star the path will be as follows:

Cluster1(1-2-3-4-5)-Cluster2(6-1-2-3-4-5)-Cluster3(5-4-3-2-1-6)-Cluster4(5-4-3-2-1-6)-Cluster1(6). Hence , the path length= (n!-1)=23.

Hence n-star is also Hamiltonian. Metastar being a hybrid network will be also Hamiltonian.

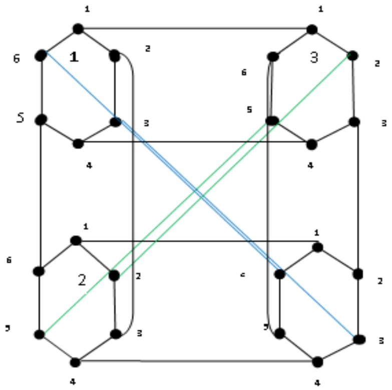


Fig. 8. Constructing Hamiltonian Path in 4-Star

4.2 Routing in Metastar

Suppose *s* and *t* be two nodes in the Metastar(k, m) network. The nodes *s* and *t* belong to classes C^s and C^t respectively. Then there will be three cases.

First case is *s* and *t* both belong to the same class and same cluster.

Second case is *s* and *t* both belong to the same class but different clusters.

Third case is *s* and *t* both belong to different class.

For first case routing will be same as n-Star routing. For second and third case following is the algorithm.

Algorithm Mstar Routing(k,m,s,t)

```
{
  Step1: Construct a Hamiltonian path from source class $C^s$ to destination class$C^t$.
  Step2: In each class, a shortest path $P_i$ will be decided using star routing within each cluster.
  Step3: Each $P_i$ will be concatenated to form the path from s to t.
}
```

Whenever there is a change of class a cross link will be inserted according to the Hamiltonian path found in step 1.

4.3 Illustration

Let *s* and *t* be two farthest nodes in the Metastar(1,3) network . The node addresses of s be (0,1,1) and t be (0,4,22). Then the path from *s* to *t* is as follows:

(0,1,1)*****(1,1,1)---(1,1,7)---(1,1,13)---(1,1,19)******(0,4,19)---(0,4,20)---(0,4,21)---(0,4,22), (*** represents cross link and --- represents star link that is link within cluster). So distance is 8.

4.4 Broadcasting

The broadcasting algorithm for the Metastar network is as follows.

Algorithm: MstarBroadcast(k,m)

```
{
Step 1: Broadcast the message from s to all other clusters.
Step 2: For i=0 to $2^k - 1$ do
          For each cluster in which at least one node has the message pardo
Step 3: Broadcast the message to all other nodes in the cluster
Step 4: Find the next neighbour in the cluster using star routing.
          endfor
endfor
}
```

5 Performance Analysis

To establish the superiority of the proposed network the comparison of various topological parameters of the Metastar network with the contemporary networks are done in this section.

The comparison of node degree with respect to dimension is shown in Fig. 9. The n-Star has the lowest degree with less number of nodes. The other three hybrid networks the Metacube, Starcube and the Metastar networks contain more number of nodes with same node degree. But the Metastar network exhibits the better values. The comparison of cost is shown in Fig.10. Metastar possesses lower cost property as compared to Metacube with higher packing density and lower node degree.

The diameter of the Metastar network is more or less equal to that of the Metacube while connecting to comparatively large number of nodes. The Star graph and the Starcube network both possess lower values and they also contain comparatively less number of nodes. The comparison is shown in Fig.11. Table 1 shows the total number of nodes or the packing density of all the four networks with respect to node degree. The shown values are computed with k=1 and m taking different values. The Metastar contains maximum nodes at all possible node degrees. Metacube and Hierarchical

Star do not exist with degree 2. Similarly Starcube does not contain any value at node degree 2 and 3. HFN also does not exist at node degree 2,3 and 4.

Table 1. Comparison of Packing density

Degree	MC	n-Star	Starcube	Metastar	HS	HFN
2	-	6	-	8	-	-
3	32	24	-	72	36	-
4	128	120	24	144,1152	576	-
5	512	720	96	28800	14400	64
6	2048	1520	480			256

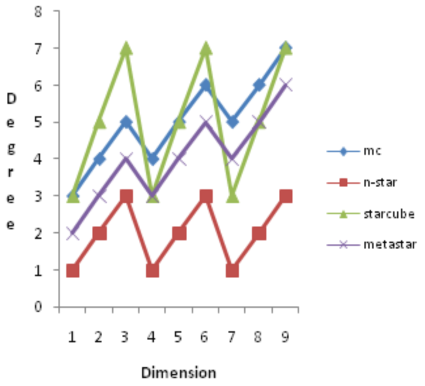


Fig. 9. Degree comparison

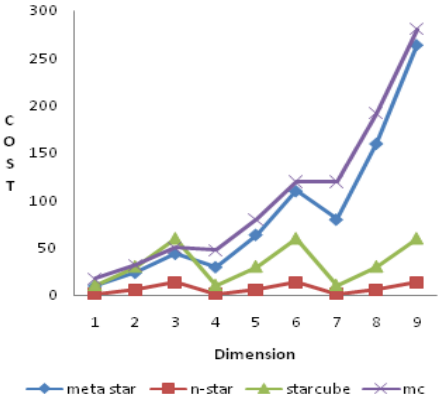
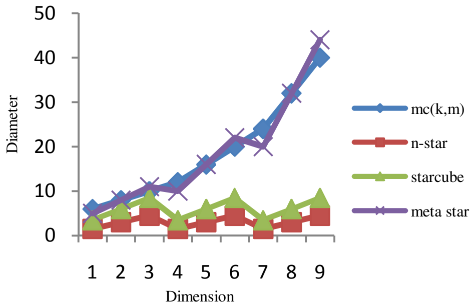


Fig. 10. Cost versus dimension



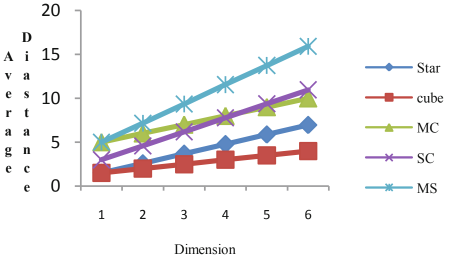


Fig. 11. Comparison of diameter against dimension

Fig. 12. Comparison of average distance against dimension

6 Conclusion

The current paper introduced a new large scale parallel interconnection network called the Metastar. Some of the topological properties for the Metastar network are derived. Routing and broadcasting algorithms are proposed. The most important feature of this network is that with reduced node degree and cost the proposed network can connect to a very large number of processing elements like Metacube as well as it can possess small values also. Thus it outperforms the other existing two level networks in terms of the topological parameters. The performance analysis proves the Metastar to be a better candidate for small as well as large scale parallel processing systems.

References

1. Nibedita, A., Tripathy, C.R.: Folded Dualcube: A New interconnection for Parallel Systems. In: Proceedings of 11th Int. Conf. on Information Technology, December 17-18, pp. 75–78. IEEE Comp. Society (2008)
2. Tripathy, C.R.: Star-cube: A New Fault Tolerant Interconnection Topology For Massively Parallel Systems. IE(I) Journal, ETE Div., 84(2), 83–92 (2004)
3. Preparta, F.P., Vullemin, J.: The Cube Connected Cycles: A versatile Network for parallel Computation. Communication ACM 24(5), 300–309 (1981)
4. Hayes, J.P., Mudge, T.N.: Hypercube Super-computers. Proc. IEEE 77(12), 1829–1841 (1989)
5. Efe, K.: The Crossed Cube Architecture For Parallel Computation. IEEE Tran. On Parallel and Distributed Systems 3(5), 513–524 (1992)
6. Ghose, K., Desai, K.R.: Hierarchical cubic networks. IEEE Transactions on Parallel and Distributed Systems 6(4), 427–435 (1995)
7. Bhuyan, L.N., Agrawal, D.P.: Performance Of Multiprocessor Interconnection Network. IEEE Computers (1989)
8. Bhuyan, L.N., Agarwal, D.P.: Generalized Hypercube and Hyperbus Structures For a Computer Network. IEEE Tran. On Computers C-33(4), 323–333 (1984)
9. Latifi, S., Bagherzadeh, N.: Incomplete Star: an Incrementally Scalable Network Based on the Star Graph. IEEE Transactions Parallel and Distributed Systems 5, 97 (1994)
10. Akers, S.B., Krishnamurthy, B.: The Fault-tolerance of Star Graphs. In: Proceedings of International Conference on Supercomputing, p. 270 (1987)
11. Feng, T.: A survey Of Interconnection Networks. IEEE Computers 1(4), 12–27 (1981)
12. Shi, W., Srimani, P.K.: Hierarchical star: a new two level interconnection network. Journal of Systems Architecture 51, 1–14 (2005)
13. Li, Y., Peng, S., Chu, W.: Efficient Collective Communications in Dual-cube. The Journal of Super Computing 28, 71–90 (2004)
14. Li, Y., Peng, S., Chu, W.: Metacube: A New Interconnection Network for Large Parallel System. ACSAC02, Australian Computer Science Communications 24(4), 29–36 (2001)
15. Saad, Y., Schultz, M.H.: Topological Properties of Hypercubes. IEEE Transactions on Computers 37(9), 867–872 (1988)
16. Duh, D.-R., Chen, G.H., Fang, J.F.: Algorithms and Properties of a New Two Level Network with Folded Hypercube As basic Modules. IEEE Transactions on Parallel and Distributed Systems 6(7), 714–723 (1995)

An Improved Scheme for False Data Filtering in Wireless Sensor Networks

C. Anudeep and Manik Lal Das

Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar - 382007, India
{cuv_anudeep,maniklal_das}@daiict.ac.in

Abstract. Wireless Sensor Networks (WSN) consist of a large number of sensor nodes equipped with limited computational capacity, memory and battery energy. As sensor nodes in WSN deployed in a hostile and unattended environment, the networks is susceptible to various malicious threats. False data injection is one of them. An adversary may attempt to inject false data containing non-existent events through some compromised nodes causing false alarms at the base station and draining out energy of forwarding nodes. As a result, detection of false data injection in WSN is an important concern. This paper presents a scheme using hash-based short signatures for filtering false data injection in WSN.

Keywords: Sensor networks, false data injection, short signature, node compromise.

1 Introduction

Wireless Sensor Networks (WSN) [1,2] have found increasing interests from researchers due to their ubiquitous nature, easy deployment and the range of applications they enable. Networks of thousands tiny sensor devices which have low processing power, limited memory and energy, provide an economical solution to some challenging problems such as military surveillance, behaviour monitoring, measurement of seismic activity, object tracking, healthcare and so on. Sensor nodes in WSN communicate with each other and with a base station through wireless channel, where a node can act as a sensing node and/or a forwarding node. As sensor nodes are deployed in an unattended and hostile environment, nodes may be captured or compromised by adversaries. Moreover, it is practically difficulty task to physically monitor all of them. By compromising node(s), secret information stored in memory could be known to the adversaries who could then inject false data into the network and exhaust the limited energy of the nodes in forwarding these reports, and thus, reducing the lifetime of the sensor networks. In order to minimize this threat, the falsified report should be identified and dropped as early as possible.

The dense deployment of sensing devices makes it possible for several sensors to detect the same event, adding reliability to the system. WSN can be designed as cluster based, where a node can act as the cluster head and can coordinate

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 62–70, 2012.

a few nodes in the cluster, collects and forwards data to the base station. The sensing nodes select one node among them to be cluster head, and all sensor nodes in the cluster send their reports to the cluster head. To balance energy consumption, all nodes in the cluster take turns to serve as cluster head. The cluster head aggregates all reports and forwards them to the base station through some forwarding nodes, where preventing false data injection is an important concern in WSN.

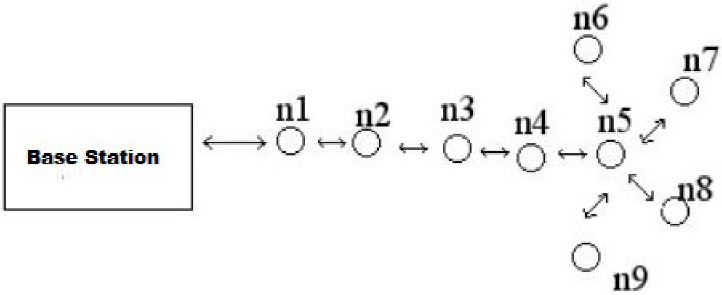


Fig. 1. Cluster-based Sensor Networks

[node $n5$ is cluster head, nodes $n1$-$n4$ are forwarding nodes and nodes $n5$-$n9$ form a cluster]

Several security solutions [3], [8] have been proposed using symmetric key cryptography for false data filtering in WSN. The dynamic en-route filtering scheme by Yu and Guan [3] provides a mechanism for false report filtering based on Message Authentication Codes (MAC). However, a pre-agreed number of MACs required in [3], which may not provide sufficient number of filtering in real-world applications of WSN. A large number of MACs computation needed for detecting false reports, but that consumes more energy in forwarding steps. Ye *et al* [9] proposed a statistical en-route false report filtering scheme. The scheme requires each report be endorsed by multiple sensor nodes by encrypting the report with their random predistributed symmetric keys. The intermediate nodes on the route compare their own keys with those used for encrypting the report, and check the corresponding encryption if matched keys are found. Zhu *et al* [8] proposed a scheme to detect the false report, where the scheme requires that the sensor nodes should maintain a pre-route interleaved associations for detecting false report. Lee *et al* [7] proposed an adaptive threshold determination method using fuzzy logic to choose an appropriate value. Zhang *et al* [10] proposed a public key approach for false report filtering that claims better security resilience. But, the scheme uses bilinear pairing operation, which is substantially costly operation in comparison to symmetric key operation. Wang and Li [6] proposed a scheme using public key operation for false data filtering in WSN.

Our Contributions. The symmetric key based schemes for false report filtering are efficient, but they require considerable memory space and communication overhead for key pre-distribution and key discovery. In contrast, public key based approach requires costly public/private operations, which are not suitable for resource-constrained sensor nodes. In this paper, we present a hybrid approach for filtering false reports in WSN using hash-based short signature. The proposed scheme requires slight more cost than symmetric key approach but substantially less cost compared to public key based approach. Our scheme is based on the Yu and Guan's scheme [3] by applying short signature for better security resilience.

Organization of the Paper. The remainder of this paper is organized as follows. Section 2 describes a hash based short signature scheme. As our scheme is based on Yu and Guan's scheme, section 3 details the Yu and Guan's scheme. Section 4 presents our scheme. Section 5 analyzes the scheme. We conclude the work in section 6.

2 Hash-Based Short Signature

Dahmen and Krauβ [4] proposed a hash-based short signature scheme. The scheme [4] is parameterized by three integers n, l, w, where

n : security level

l : $l \geq 1$, number of signatures to be generated
w : maximum bit length of message to be signed

The scheme has four phases - setup, key-pair generation, signature generation and signature verification, as explained below.

2.1 Setup

Let $f : \{0,1\}^n \rightarrow \{0,1\}^n$ and $g : \{0,1\}^{4n} \rightarrow \{0,1\}^n$ be one-way functions that map bit strings of length n and $4n$ to bit strings of length n, respectively. Let PRF: $\{0,1\}^n \rightarrow \{0,1\}^n \times \{0,1\}^n$ be a pseudo-random function that maps an n bit seed to an n-bit pseudo-random number and an n-bit updated seed, i.e., $(rand, seed_{out}) \leftarrow \text{PRF}(seed_{in})$.

2.2 Key-Pair Generation

A trusted party generates key-pair by the following steps:

(i) choose an initial seed $\psi \in \{0,1\}^n$ and an end link $z_l \in \{0,1\}^n$ at random.
(ii) generate $X_i = (x_i[0], x_i[1], x_i[2])$ for $i = 1, 2, \cdots l$ using the PRF$(\cdot)$ and initial seed as
$(x_i[0], \psi'_i) \leftarrow \text{PRF}(\psi_i)$
$(x_i[1], \psi''_i) \leftarrow \text{PRF}((\psi'_i)$
$(x_i[2], \psi'''_i) \leftarrow \text{PRF}(\psi''_i)$

(iii) compute $Y_i = (y_i[0], y_i[1], y_i[2])$ for $i = 1, 2, \cdots, l$ and hash chain links $z_i \in \{0,1\}^n$ for $i = 0, 1, \cdots, l-1$ as follows:
$y_i[0] \leftarrow f^{2^{w/2}-1}(x_i[0])$,
$y_i[1] \leftarrow f^{2^{w/2}-1}(x_i[1])$,
$y_i[2] \leftarrow f^{2^{w/2+1}-1}(x_i[2])$,
$z_{i-1} \leftarrow g(y_i[0] \| y_i[1] \| y_i[2] \| z_i)$

where w is the bit-length of the message $m(= m_1 \| m_2)$ to be signed.
Initially, z_0 is the public key, and pair (ψ_1, z_1) acts as the private key.

2.3 Signature Generation

For a w-bit message $m(= m_1 \| m_2) \in \{0, 1, \cdots, 2^{w/2} - 1\}$,
- compute checksum $c \leftarrow 2^{w/2+1} - 2 - m_1 - m_2$
- generate the one-time signature of m

$$\alpha_1 \leftarrow f^{m_1}(x_i[0]), \qquad \alpha_2 \leftarrow f^{m_2}(x_i[1]), \qquad \alpha_3 \leftarrow f^{c}(x_i[2])$$

The signature of m is given as $\sigma = (i, \alpha_1, \alpha_2, \alpha_3, z_i)$, where i is the index of the signature.

2.4 Signature Verification

A verifier uses $\alpha_1, \alpha_2, \alpha_3$ to compute verification key $Y_i = (\beta_1, \beta_2, \beta_3)$ and checks whether z_{i-1} can be computed using Y_i and z_i. The verification proceeds as follows:

$$\beta_1 = f^{2^{w/2}-1-m_1}(\alpha_1)$$
$$\beta_2 = f^{2^{w/2}-1-m_2}(\alpha_2)$$
$$\beta_3 = f^{2^{w/2+1}-2-c}(\alpha_3)$$

Check if $g(\beta_1 \| \beta_2 \| \beta_3 \| z_i) \leq z_{i-1}$. If it holds, the signature is accepted; otherwise, rejected. The verifier discards z_{i-1} and stores z_i for verification of next signature.

3 Yu and Guan's Scheme

Yu and Guan [3] proposed a scheme for false report detection in WSN. The scheme has three phases - *key predistribution*, *key dissemination* and *report forwarding*. In their scheme, a set of sensor nodes monitored by one or more cluster heads and the cluster head forwards data to the base station as and when required.

3.1 Key Predistribution

Each sensor node is loaded with a seed key and l+1 secret keys, where l keys are picked from a global key pool called *y-keys* and one key from another pool called *z-keys*. It is assumed that base station is aware of the seed key of every node. The nodes compute a hash-chain of authentication keys (*auth-key*) using a secure hash function $h(\cdot)$ and seed value.

$$\begin{aligned} k^{v_i}_{m-1} &= h(k^{v_i}_{m}) \\ k^{v_i}_{m-2} &= h(k^{v_i}_{m-1}) = h^2(k^{v_i}_{m}) \\ &\vdots \\ k^{v_i}_{1} &= h^m(k^{v_i}_{m}) \end{aligned}$$

where m : length of the hash-chain
v_i : index of the node
$k^{v_i}_m$: seed value of the node

3.2 Key Dissemination

In this phase, the cluster head discloses the sensing nodes' *auth-keys* after sending the reports. Naturally, there could be possibility that a malicious node can pretend to be a cluster head and can inject arbitrary reports followed by falsified keys. With key dissemination, the cluster head disseminates the first unused *auth-keys* of all nodes to the forwarding nodes before sending the reports. The key dissemination phase works as follows.

- Each node constructs an authenticated message that contains $l{+}1$ copies of current *auth-key*, each encrypted using one of its secret keys. For instance, for node v_i,
  $auth(v_i) = \{v_i,\ j_i,\ \mathrm{id}(y_1^{v_i}),\ \{\mathrm{id}(y_1^{v_i}),\ k^{v_i}_{j_i}\}_{y_1^{v_i}},\ \cdots,\ \mathrm{id}(y_l^{v_i}),\ \{\mathrm{id}(y_l^{v_i}), k^{v_i}_{j_i}\}_{y_l^{v_i}},\ \mathrm{id}(z^{v_i}),\ \{\mathrm{id}(z^{v_i}),\ k^{v_i}_{j_i}\}_{z^{v_i}}\}$,
  where j_i indicates the index of the key and v_i indicates index of the node.
- The cluster head aggregates authenticated messages of all the nodes into a single message $K(n)$ as $K(n) = \{auth(v_1),\ auth(v_2),\ \cdots,\ auth(v_n)\}$.
- The cluster head chooses q forwarding nodes and forwards message $K(n)$ to them.
- When a forwarding node receives $K(n)$, the node
  - verifies the validity of $K(n)$. If it contains at least t distinct indexes of z-keys then $K(n)$ is accepted; otherwise, rejected.
  - checks the indices of secret keys in $K(n)$ to see if it has any shared key. Upon finding a shared key, the node decrypts the corresponding *auth-key* using that key and stores it in memory.

3.3 Report Forwarding

In this phase, sensing node generates sensing reports in rounds. The sensing report is $r(v_i) = \{\text{event\_info},\ v_i,\ j_i,\ \text{MAC}(\text{event\_info},\ k^{v_i}_{j_i})\}$, where j_i indicates index of the key, v_i indicates index of the node.
The phase works as follows:

- cluster head collects the sensing reports, generates an aggregated report $R = \{r(v_{i_1}), r(v_{i_2}), \cdots, r(v_{i_t})\}$ and sends R and an OK message to forwarding node u_j.
- upon receiving R, u_j broadcasts the report to next hop u_{j+1}.
- cluster head on overhearing the broadcast from u_j, discloses the *auth-key* to u_j by message $K(t)$, where $K(t) = \{ \mathit{auth}(v_1), \mathit{auth}(v_2), \cdots, \mathit{auth}(v_t)\}$.
- upon receiving $K(t)$,
  - u_j checks whether $K(t)$ contains t distinct indexes of z-keys. If not, it drops $K(t)$.
  - to verify the correctness of the *auth-keys* in $K(t)$, u_j checks if each *auth-key* it stored can be generated by hashing a corresponding key in $K(t)$ with certain number of times.
  - to verify the integrity and authenticity of reports, u_j checks the MACs in these reports using the disclosed *auth-keys* that it decrypts from $K(t)$.
- if reports are valid, u_j forwards an OK message to u_{j+1}. Otherwise, it will ask u_{j+1} to drop the report.

The above mentioned steps continue until the reports reach the base station.

3.4 Yu and Guan's Modified Scheme

Yu and Guan modified their scheme [3] using the *Hill Climbing approach* [5]. The modifications are performed in the key pre-distribution and in the key disseminate phases. The main reason for modifying their scheme was to filter false reports at the nodes closer to clusters, as nodes closer to the base station have no chance to use the *auth-keys* they stored. The modified steps are given below:

- In the key pre-distribution phase, instead of picking y-keys from a global pool, every node selects y-keys randomly from its hash-chain. Therefore, a forwarding node holding a larger index y-keys can now decrypt a sensing node's *auth-key* from $K(n)$, as long as the sensing node's y-keys has a smaller index.
- In the key dissemination phase, once a forwarding node decrypts an *auth-key* from $K(n)$, it updates $K(n)$ by encrypting the *auth-key* using its own y-key. This substitution at every forwarding node increases gradually the indexes of y-keys contained in $K(n)$, just like climbing the Hill. It makes harder for the nodes closer to base station to decrypt the *auth-keys* from $K(n)$.

The modification is needed because when multiple clusters disseminate keys at the same time, some forwarding nodes may need to store the *auth-keys* for different clusters. The nodes closer to base station need to store more *auth-keys* than others, as they are usually the hot spots and have to serve more request forwarded by some clusters.

4 The Proposed Scheme

We present a scheme based on Yu and Guan's scheme [3] for filtering false reports in WSN using the concept of hash-based short signature [4]. The scheme has four phases - Setup, Keypair generation, Key dissemination and report forwarding.

4.1 Setup

Let $f : \{0,1\}^n \rightarrow \{0,1\}^n$ and $g : \{0,1\}^{4n} \rightarrow \{0,1\}^n$ be one-way functions that map bit strings of length n and $4n$ to bit strings of length n, respectively. Let $PRF : \{0,1\}^n \rightarrow \{0,1\}^n \times \{0,1\}^n$ be a pseudo-random function that inputs an n bit seed and outputs an n bit random number and an n-bit updated seed, i.e., $(rand, seed_{out}) \leftarrow PRF(seed_{in})$.

4.2 Key-Pair Generation

The base station performs the following operations:

- Choose a random initial seed $\psi \in \{0,1\}^n$ and end link $z_l \in \{0,1\}^n$.
- Generate $X_i = (x_i[0], x_i[1], x_i[2])$ for $i = 1, \cdots, l$ using the $PRF(\cdot)$ and seed.
  $(x_i[0], \psi_i') \leftarrow PRF(\psi_i)$
  $(x_i[1], \psi_i'') \leftarrow PRF(\psi_i')$
  $(x_i[2], \psi_i''') \leftarrow PRF(\psi_i'')$.
- Compute $Y_i = (y_i[0], y_i[1], y_i[2])$ for $i = 1, 2, \cdots, l$ and hash chain links $z_i \in \{0,1\}^n$ for $i = 0, 1, \cdots, l-1$.
  $y_i[0] \leftarrow f^{2^{w/2}-1}(x_i[0])$
  $y_i[1] \leftarrow f^{2^{w/2}-1}(x_i[1])$
  $y_i[2] \leftarrow f^{2^{w/2+1}-1}(x_i[2])$
  $z_{i-1} \leftarrow g(y_i[0] \| y_i[1] \| y_i[2] \| z_i)$

where w is the length of the message $m(= m_1 \| m_2)$ to be signed.
Initially z_0 acts as the public key and the private key consists of the seed ψ_1 and the link z_1. Every node is loaded with seed value and hash chain-link.

4.3 Key Dissemination

The base station broadcasts z_{i-1} values of all nodes in the cluster. Every forwarding node stores these values.

4.4 Report Forwarding Phase

Sensing nodes generate reports and forward them to cluster head after creating a one-time signature as $r(v_i) = \{\text{event\_info}, v_i, j_i, \alpha_1, \alpha_2, \alpha_3, z_i\}$.
The cluster head aggregates all the reports and forwards $R = \{r(v_1), r(v_2), \cdots, r(v_t)\}$ to the next forwarding node.
The verification by the forwarding node is done by the steps mentioned in section 2.4.

5 Analysis

5.1 Security Analysis

We assume that the adversary is given the one-time signature $(\alpha_1, \alpha_2, \alpha_3)$ of message m $(= m_1 \| m_2)$ and the adversary wants to generate a valid signature

$\sigma' = (\alpha'_1, \alpha'_2, \alpha'_3)$ on a different message $m'(= m'_1 \| m'_2)$. The following two cases arise:

Case 1. $m'_1 < m_1$ or $m'_2 < m_2$
In order to generate a valid one-time signature, the adversary must compute $\alpha'_1 = f^{m'_1 - m_1}(\alpha_1)$ and $\alpha'_2 = f^{m'_2 - m_2}(\alpha_2)$. This requires the adversary to compute preimage of the one-way function *f*, as either $m'_1 - m_1 < 0$ or $m'_2 - m_2 < 0$ holds.

Case 2. $m'_1 \geq m_1$ and $m'_2 \geq m_2$
In this case, the adversary can compute $\alpha'_1 = f^{m'_1 - m_1}(\alpha_1)$ and $\alpha'_2 = f^{m'_2 - m_2}(\alpha_2)$. However, to compute $\alpha'_3 = f^{c'-c}(\alpha_3)$, the adversary needs to compute preimage of the function f, because the checksum c' of the message m' is smaller than the checksum c of the message m, that is, $c' - c = m_1 - m'_1 + m_2 - m'_2 < 0$.
Therefore, the proposed scheme is secure as long as the functions used in the protocol are preimage resistant.

When a cluster head is compromised.
In our scheme, normal sensor nodes can act as cluster heads, so there is no difference between a cluster head and a sensor node. There could be possibility that a cluster head is compromised or any compromised node can claim to be a cluster head. In that scenario, the cluster head requires to broadcast a forged aggregate report. However, as discussed above, it is practically difficult to forge a signature by compromising a cluster head. As a consequence, the attempt of forging report will be caught by next forwarding nodes or by the base station.

5.2 Performance Analysis

In the proposed scheme, all the computations for the setup and key-pair generation phases are done offline. The online computations primarily are involved in report forwarding phase. The scheme has a trade-off between signature generation/verification time and the maximum length of message to be signed. It has been observed that based on the nature of applications in WSN, a 14-bit message size would provide intended information [4].

We consider the digest size of 160-bit, l is 2^{10} and the size of w is 14-bit. In that case, the storage cost for one-time signature keys would be $2^{10} \times 3 \times 80 = 245760$ bits and $2^{10} \times 80 = 81920$ bits for hash-chain links. Therefore, a total 327680 bits or 40 KB storage required in each node deployed in WSN. As far as the computational cost is concerned, our scheme is also efficient, as the signature and verification require only hash operations. Moreover, about 330 bits needed by a node for report forwarding. Therefore, the scheme is efficient with respect to communication cost. The summary of storage, computation and communication cost of our scheme is provided in Table 1.

6 Conclusion

We have discussed false report filtering in WSN and proposed a scheme for filtering false report at an early stage using the hash-based short signature.

Table 1. Performance analysis of the proposed scheme

	Storage	**Computation**	**Communication**
Offline	40 KB	2^{17} hashes	Nil
Online	40 KB	253 hashes	330 bits

In the proposed scheme, the false report can be identified when a verifier of the signature notices any forgeries on signed messages. The proposed scheme requires little more cost than Yu and Guan's scheme [3] but less cost in comparisons with the public key based approaches [10], [6]. The proposed scheme is secure as long as the function used for signature creation resists preimage property.

Acknowledgement. This work is supported in part by Department of Science and Technology, Ministry of Science & Technology, Government of India through DST/INT/SPAIN/P-6/2009 Indo-Spanish Joint Programme of Cooperation in Science and Technology.

References

1. Callaway Jr., E.H.: Wireless Sensor Networks. Architectures and Protocols. Auerbach Publications (2003)
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: A survey. Computer Networks 38(4), 393–422 (2002)
3. Yu, Z., Guan, Y.: A dynamic en-route scheme for filtering false data injection in wireless sensor networks. In: Proc. of the IEEE International Conference on Computer Communications (INFOCOM), pp. 1–12 (2006)
4. Dahmen, E., Krauß, C.: Short Hash-Based Signatures for Wireless Sensor Networks. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 463–476. Springer, Heidelberg (2009)
5. Weise, T.: Global Optimization Algorithms - Theory and Application (2009), http://www.it-weise.de/
6. Wang, H., Li, Q.: PDF: A Public-key based False Data Filtering Scheme in Sensor Networks. In: Proc. of the International Conference on Wireless Algorithms, Systems and Applications, pp. 129–138 (2007)
7. Lee, S.J., Lee, H.Y., Cho, T.H.: Environment-based Selection Method for Enroute Filtering Scheme using Fuzzy Logic. Journal of Networks 5(3), 292–299 (2010)
8. Zhu, S., Setia, S., Jajodia, S., Ning, P.: An Interleaved Hop-by- Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks. In: Proc. of IEEE Symposium on Security and Privacy, pp. 259–271 (2004)
9. Ye, F., Luo, H., Lu, S., Zhang, L.: Statistical En-route Filtering of Injected False Data in Sensor Networks. In: Proc. of the IEEE International Conference on Computer Communications, INFOCOM (2004)
10. Zhang, Y., Liu, W., Lou, W., Fang, Y.: Location based Compromise-tolerant Security Mechanisms for Wireless Sensor Networks. IEEE Journal on Selected Areas in Communications 24(2), 247–260 (2006)

Anonymity and Security in Mobile Ad Hoc Networks

Jhansi Vazram Bolla[1,*], Valli Kumari Vatsavayi[2], and J.V.R. Murthy[3]

[1] Dept.of CSE., Narasaraopeta Engg. College, Andhra pradesh, India-522601
[2] Dept. of CS & SE., Andhra University, Andhra Pradesh, India – 530003
[3] Dept. of CSE., J N T University, Kakinada, Andhra Pradesh, India – 533003
{jhansi.bolla,vallikumari,mjonnalagedda}@gmail.com

Abstract. In mobile adhoc networks, generating and maintaining anonymity for any adhoc node is challenging because of the node mobility, dynamic network topology, cooperative nature of the network and broadcast nature of the communication media. Anonymity is provided to protect the communication, by hiding the participants as well as the message contents. Existing techniques based on cryptosystem and broadcasting cannot be easily adapted to MANETs because of their extensive cryptographic computations and/or large communication overheads. In this paper, we first propose an unconditionally secure privacy preserving message authentication scheme (PPMAS) which uses Modified New variant ElGamal signature Scheme (MNES). This scheme enables a sender to transmit messages, providing authentication along with anonymity, without relying on any trusted third parties. The anonymous message uses privacy preserving communication protocol for MANET, which is capable of anonymous end to end connections. It also allows the untraceability of the link between the identifier of a node and its location. The experimental analysis of the proposed system is presented.

Keywords: Network security, anonymity, mobile adhoc networks.

1 Introduction

A mobile ad hoc network (MANET) comprises of a set of wireless devices that can move around freely and cooperate in relaying packets on behalf of one another. A MANET does not require a fixed infrastructure or centralized administration. Distant mobile nodes communicate through multi hop paths, as they have limited transmission range. Their ease of deployment makes MANETs an attractive choice for a variety of applications like include battleground communications, disaster recovery efforts, communication among a group of islands or ships, conferencing without the support of a wired infrastructure, and interactive information sharing. In MANETs, mobile nodes cooperate to forward data on behalf of each other. Typical protocols used for self organizing and routing in these networks expose the node identifiers (network and link layer addresses), neighbors, and the end-points of communication. Some modes of operation further mandate that the nodes freely divulge their physical location. In short, nodes must advertise a profile of their online presence to participate in the MANETs, which is highly undesirable.

[*] Corresponding author.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 71–82, 2012.

Both military and civilian MANETs may find the mandated exposure of information unacceptable, a node should be able to keep its identity, its location and its correspondents private, i.e., remain anonymous [8], [17]. Any solution providing anonymity must overcome the broadcast nature of wireless environments (which enables eavesdropping) and operate under often tight resource constraints, unlike wired networks. Simple solutions like packet encryption are also largely ineffective because of ease of traffic analysis over a broadcast media. Hence, supporting privacy in MANETs is enormously challenging.

Outline of the paper: Section 2 presents the related work done discusses. Section 3 gives overview of the proposed privacy preserving unconditionally secure message authentication scheme. Section 4 proposes a privacy preserving communication protocol. Section 5 discusses security analysis. Section 6 gives the performance analysis. Section 7 concludes the paper and suggests possible extensions.

2 Related Work

2.1 Terminology

Unlinkability[18] of two or more items of interest (subjects, messages, actions, ...) means that within the system (comprising these and possibly other items), from the attacker's perspective, these items of interest are no more and no less related after his observation than they are related concerning his apriori knowledge.

In a MANET, this definition involves: *sender anonymity* when an attacker cannot *link* a message with its originator; *recipient anonymity* when the message is *unlinkable* to its final destination; and *relationship anonymity* if the attacker cannot determine that sender S is communicating with destination D, even though it could determine either S or D (not both for the same message). Therefore, sender or recipient anonymity implies relationship anonymity.

Privacy (sometimes referred to as anonymity), refers to the state of not being identifiable with in a set of objects, called ambiguity set (AS).

We will begin with the definition of unconditionally secure privacy preserving message authentication scheme (PPMAS) [7].

Definition 1 (PPMAS)**.** A PPMAS consists of the following two algorithms:

(i) ***generate*** $(\boldsymbol{m}, \boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_n})$: Given a message m and the public keys y_1, y_2, . . . , y_n of the anonymity set *(AS)* $\mathcal{S} = \{A_1, A_2, \ldots, A_n\}$, the actual message sender A_t, $1 \leq t \leq n$, produces an anonymous message $\mathcal{S}(m)$ using her own private key x_t; In this paper, the user ID and user public key will be used interchangeably.

(ii) ***verify*** $\boldsymbol{\mathcal{S}(m)}$: Given a message m and an anonymous message $\mathcal{S}(m)$, which includes the public keys of all members in the AS, a verifier can determine whether $\mathcal{S}(m)$ is generated by a member in the AS.

The security requirements for PPMAS include

(i) *Sender anonymity:* The probability that a verifier successfully determines the real sender of the anonymous message is exactly $1/n$, where n is the size of AS;

(ii) *Unforgeability:* An anonymous message scheme is unforgeable if no adversary, given the public keys of all members of the AS and the anonymous messages m_1, m_2, . . . ,m_l adaptively chosen by the adversary, can produce in polynomial time a new valid anonymous message with nonnegligible probability.

2.2 Modified New Variant ElGamal Signature Scheme (MNES)

Definition 2: Based on the New variant ElGamal signature scheme [3], we propose a modified new variant ElGamal signature scheme (MNES), which consists of the following 3 algorithms:

i) Key generation algorithm

Let P be a large prime, α be a generator of Z^*. Both p and α are made public. For a random private key x ∈ Z_p^*, the public key y is computed from

$$y = \alpha^{x} \bmod p.$$

ii) Signature algorithm

The MNES can also have many variants. For the purpose of efficiency, we will describe the variant, called optimal scheme. To sign a message m, one chooses a random k, l ∈ Z_{p-1}^* , then computes the exponentiation r= α^{k} mod p d s = α^{l} mod p , q = α^{h} mod p and solves w from

$$w = k + l + h + xr + ks + lq \;\; mod \;\; (p-1) \tag{1}$$

where h is a one way hash function .The signature of the message m is defined as the quadruple (r, s, w).

iii) Verification algorithm

The verifier checks the signature equation
$\alpha^{w} = rsqy^{r}r^{s}s^{q} \;\; mod\, p$, h=h(m, rs). If the equality holds true, then the verifier accepts the signature and rejects otherwise.

We use the following Notations throughout the paper

- PPMAS: Privacy preserving message authentication scheme
- PPMAC: Privacy preserving message authentication code
- MNES: Modified new variant ElGamal signature scheme
- $\mathscr{s}$ or AS: Ambiguity set
- $\mathscr{s}(m)$: PPMAS of message m
- $\mathcal{M}$(i, j): Message from node i to node j
- m: Original message
- n: Number of nodes in the AS

2.3 A Discussion on Existing Works

The existing anonymous communication protocols are largely stemmed from either mixnet[2] or DC-net[4]. The secrecy of user's communication relationship can be protected by using mixnet. Anonymity can be provided by packet reshuffling through at least one trusted “mix”. The outgoing message and the ID of the recipient are encrypted by the sender using the public key of the “mix”. Recently, Moler presented a secure public-key encryption algorithm for mixnet[10]. This algorithm has been adopted by Mixminion[9]. However, since mixnet like protocols rely on the statistical properties of background traffic, they cannot provide provable anonymity.

Crowds[13] extends the idea of anonymizer and is designed for anonymous web browsing. However, Crowds only provides sender anonymity. Packet content and receivers would not be hidden by the en route nodes. Hordes [11] builds on the Crowds. It provides only sender anonymity and uses multicast services.

DC-net [4, 6] is an anonymous multiparty computation amongst a set of participants, some pairs of which share secret keys. Without the need of trusted servers DC-net provides perfect sender anonymity. In order to achieve receiver anonymity users have to send encrypted broadcasts to the entire group.

Recently, message sender anonymity based on ring signatures was introduced [7] and [12].This method provides an assurance to the sender that the generated message has source anonymous signature along with content authenticity, while hiding the message sender’s real identity.

In this paper, we first propose an unconditionally secure privacy preserving message authentication scheme (PPMAS) based on the modified new variant ElGamal signature scheme. This is because the original ElGamal signature scheme is existentially forgeable with a generic message attack [14, 15]. While the modified ElGamal signature (MES) scheme [7] is secure against no-message attack and adaptive chosen message attack in the random oracle model [16], it cannot be used for more than one messages. The modified new variant ElGamal signature scheme (MNES) is almost very similar to MES, and also [3] we can transmit more than one message without changing the secret exponents.

3 Unconditionally Secure Privacy Preserving MAC (PPMAS)

In this section, we propose an efficient privacy preserving unconditionally secure message authentication scheme (PPMAS). The main idea is that for each message *m* to be released, the sending node generates a privacy preserving message authentication for the message *m*. The generation is based on the MNES scheme. Unlike ring signatures, which requires to compute a forgery signature for each member in the AS separately, our scheme only requires three steps to generate the entire PPMAS. This scheme links all non-senders and the message sender to the PPMAS alike. In addition, our design enables the PPMAS to be verified through a single equation without individually verifying the signatures.

3.1 The Proposed PPMAS Scheme

Suppose that the message sender (say Alice) wishes to transmit a message *m* anonymously from her network node to any other node. The AS includes *n* members, A_1, A_2, . . . , A_n, for example, $\mathscr{S}= \{A_1,A_2, . . . ,A_n\}$, where the actual message sender Alice is A_t, for some value t, $1 \leq t \leq n$.

Let p be a large prime number and α be a primitive element of Z_p^*. Then α is also a generator of Z_p^*. That is $Z_p^* = <\alpha>$. Both p and α are made public and shared by all members in $\mathcal{S}$. Each $A_i \in \mathcal{S}$ has a public key $y_i = \alpha^{x_i} \, mod \, p$, where x_i is a randomly selected private key from Z_{p-1}^*. In this paper, we do not distinguish between the node A_i and its public key y_i. Therefore, we also have $\mathcal{S} = \{y_{1,} y_{2,............,} y_n\}$.

Suppose m is a message to be transmitted. The private key of the message sender Alice is x_t, $1 \le t \le n$. To generate an efficient PPMAS for message m, Alice performs the following three steps:

(1) Select a random and pair wise different k_i, l_i for each $1 \le i \le n$, $i \ne t$ and compute $r_i = \alpha^{k_i} \, mod \, p$, $s_i = \alpha^{l_i} \, mod \, p$, $q_i = \alpha^{h_i} \, mod \, p$ where $1 \le k_i, l_i, q_i < p$.

(2) Choose two integers k, l $randomly \; where$ $1 \le k, l < p$ and compute $r_t = \alpha^k \prod_{i \ne t} y_i^{-r_i} \, mod \, p$, $s_t = \alpha^l \prod_{i \ne t} r_i^{-s_i} \, mod \, p$ and $q_t = \alpha^h \prod_{i \ne t} s_i^{-h_i} \, mod \, p$, such that $r_t \ne 1$, $s_t \ne 1$, $q_t \ne 1$, $r_i \ne r_t$, $s_i \ne s_t$, $q_i \ne q_t$ $for \; each \; i \ne t$.

(3) Compute

$$w = k + s + \mathrm{h} + \textstyle\sum_{i \ne t} k_i + \sum_{i \ne t} l_i + \sum_{i \ne t} h_i + x_t r_t + k_t s_t \quad + \; l_t h_t \;\; mod \, (p-1).$$

The PPMAS of the message m is defined as

$$\mathcal{S}(m) = \left(m, \mathcal{S}, r_1 \ldots . r_{n,} s_1 \; \ldots . s_n, q_1 \ldots . q_n, h_1 \; \ldots . h_n, w\right) \tag{2}$$

where $\alpha^w = r_1 \ldots . r_n s_1 \ldots . s_n q_1 \ldots q_n y_i^{r_1} \ldots y_n^{r_n} \ldots r_1^{s_1} r_n^{s_n} \ldots s_1^{q_1} \ldots s_n^{q_n} \; mod \;\; p$ (3)

3.2 Verification of PPMAS

A verifier can verify an alleged PPMAS$(m, \mathcal{S}, r_1 \ldots \ldots . r_n, s_1 \ldots \ldots s_n, w)$ for message m by verifying whether the following equation

$$\alpha^w = r_1 \ldots . r_n s_1 \ldots . s_n q_1 \ldots . q_n y_i^{r_1} \ldots . y_n^{r_n} r_1^{s_1} \ldots . r_n^{s_n} \ldots . s_1^{q_1} \ldots . s_n^{q_n} \; mod \;\; p$$

holds. If (3) holds true, the verifier accepts the PPMAS as valid for message m. Otherwise the verifier rejects the PPMAS.

In fact, if the PPMAS has been correctly generated, then we have

$$\textstyle\prod_{i=1}^n r_i \; \prod_{i=1}^n s_i \; \prod_{i=1}^n q_i \prod_{i=1}^n y_i^{r_i} \; \prod_{i=1}^n r_i^{s_i} \; \prod_{i=1}^n s_i^{q_i} \;\; mod \, p$$

$$= (\textstyle\prod_{i \ne t}^n r_i) \, r_t (\prod_{i \ne t}^n s_i) \, s_t \, \left(\prod_{i \ne t}^n q_i\right) q_t \left(\prod_{i \ne t}^n y_i^{r_i}\right) y_t^{r_t}$$

$$\left(\textstyle\prod_{i \ne t}^n r_i^{s_i}\right) r_t^{s_t} \left(\prod_{i \ne t}^n s_i^{q_i}\right) s_t^{q_t} \;\; mod \, p$$

$$= \alpha^{\sum_{i \ne t} k_i} \alpha^{\sum_{i \ne t} l_i} \alpha^{\sum_{i \ne t} h_i} (\alpha^k \textstyle\prod_{i \ne t} y_i^{-r_i}) \; (\prod_{i \ne t} y_i^{r_i}) \; y_t^{r_t}$$

$$(\alpha^l \textstyle\prod_{i \ne t} r_i^{-s_i}) \, (\prod_{i \ne t} r_i^{s_i}) \, r_t^{s_t} \;\; (\alpha^h \prod_{i \ne t} s_i^{-q_i}) \;\; (\prod_{i \ne t} s_i^{q_i}) \, s_t^{q_t} \;\; mod \, p$$

$$= \alpha^{\sum_{i\neq t} k_i} \alpha^{\sum_{i\neq t} l_i} \alpha^{\sum_{i\neq t} h_i} \alpha^{k}\ y_t^{r_t}\ \alpha^{l}\ r_t^{s_t}\ \alpha^{h}\ s_t^{q_t}\ mod\, p$$

$$= \alpha^{\sum_{i\neq t} k_i\ + \sum_{i\neq t} l_i + \sum_{i\neq t} h_i + k + l + h}\ y_t^{r_t}\ r_t^{s_t}\ s_t^{q_t}\ mod\, p$$

$$= \alpha^{\sum_{i\neq t} k_i\ + \sum_{i\neq t} l_i + \sum_{i\neq t} h_i + k + l + h}\ \alpha^{x_t r_t}\ \alpha^{k_t s_t}\ \alpha^{l_t q_t}\ mod\, p$$

$$= \alpha^{k+l+h+ \sum_{i\neq t} k_i\ + \sum_{i\neq t} l_i + \sum_{i\neq t} h_i\ + x_t r_t + k_t s_t + l_t q_t}\ mod\, p$$

$$= \alpha^{w}\ \ mod\, p$$

Therefore, the verifier should always accept the PPMAS if it is correctly generated without being modified.

As a trade-off between computation and transmission, the PPMAS can also be defined as $\mathcal{s}(m) = (m, \mathcal{s}, r_1\ .\ .\ .\ r_n$, $s_1 .... s_n$, $q_1 ... q_n$, $h_1\ .\ .\ .\ h_{n,}$ w). In case $\mathcal{s}$ is also clear, it can be eliminated from the PPMAS.

3.3 Security Analysis

In this subsection, we prove that the proposed PPMAS scheme is unconditionally anonymous and provably unforgeable against adaptive chosen-message attack.

3.3.1 Anonymity

In order to prove that the proposed PPMAS is unconditionally anonymous, we have to prove that (i) for anybody other than the members of $\mathcal{s}$, the probability to successfully identify the real sender is $1/n$, and (ii) anybody from $\mathcal{s}$ can generate PPMAS.

3.3.2 Unforgeability

The design of the proposed PPMAS relies on the ElGamal signature scheme. Different levels of security can be achieved by signature schemes. The maximum level of security is a counter to existential forgery under adaptive chosen message attack.

4 The Proposed Privacy Preserving Communication Protocol

4.1 Network Assumption

As any physical transmission in a world can be monitored and traced to its origin, it is probably impossible to keep confidential who is communicating to whom by which messages. Our paper addresses the above problem. Assume that our network model similar to that discussed in [7], consists of networks with multiple MANETs, i.e., the participating nodes are divided into set of small groups and are formed in ring topology. The network nodes are categorized into two (every ring consists of both):

- ➢ *Ordinary nodes*
- ➢ *Special nodes*

An ordinary node is one that is unable to communicate directly with the nodes in other MANETS. A special node can be an ordinary node that can also provide message forward services to other MANET nodes. In some peculiar situations e.g.: energy optimization, an ordinary node can be automatically converted to a special node. Prior to the network deployment, the administrator is responsible

i) for the selection of security parameters and a group wise master key $K_G \in Z_p^*$. K_GShould be kept confidential from unauthorized access and never be uncovered to the ordinary nodes of the group.

ii) For choosing a collision resistant, one way cryptographic hash function h, e.g.: SHA-1, which maps arbitrary inputs to fixed length outputs on Z_p.

iii) To assign sufficiently large set of collision free, random pseudonyms to each special node, that can be used to substitute real IDs in communications to counter passive attacks. Because a pseudonym can be analyzed in the same way as its real ID, each has to register at the administrator, and has to get a set of random and collision free pseudonyms.

$$\mathcal{N}_A = \{PS_1^A, PS_2^A, \ldots., PS_\tau^A\}$$

For each special node, a corresponding secret set will be assigned,

$$\mathcal{S}_A = \{\alpha^{K_G h(PS_1^A)}, \ldots\ldots \ldots.\alpha^{K_G h(PS_\tau^A)}\}$$

4.2 Anonymous Communication among Ordinary Nodes and Special Nodes with in Local MANET

In anonymous communications, the message content should not consist of any explicit information such as the message sender and recipient addresses. Everything is embedded into the anonymizing message payload.

The administrator selects a set of security parameters for the entire system, before the network deployment, including a large prime p and a generator α of Z_p^*. The network nodes A_i, $1 \le i \le n$, the corresponding public keys y_i, $1 \le i \le n$ of the n participating nodes, x_i's are randomly selected private keys of A_i's, where $x_i \in Z_p$, $1 \le i \le n$, then y_i is computed from $y_i = \alpha^{x_i} \, mod \, p$.

An ordinary node can only communicate with other nodes in the same MANET. To communicate with an ordinary node in different MANET, it has to take help of the special node in the respected local MANET. Each message contains a secret key (sk), a message flag (Fm), a recipient flag (Fr), and a nonce (N). To encrypt the message payload, a secret key is used through symmetric encryption algorithm.

More specifically, for a node A_i to transmit a message m anonymously to a node A_j in the same MANET, through the nodes $A_{i+1}, \ldots\ldots. A_{j-1}$, where $j > i + 1$, node A_i generates a new message $\mathcal{M}(i, j)$ defined in (4),

$$\mathcal{M}(i, j) = pk_{i+1}(N_{i+1}, Fm_{i+1}, Fr_{i+1}, sk_{i+1}) \parallel sk_{i+1}(\mathcal{M}(i + 1, j))$$

$$\mathcal{M}(i + 1, j) = pk_{i+2}(N_{i+2}, Fm_{i+2}, Fr_{i+2}, sk_{i+2}) \parallel sk_{i+2}(\mathcal{M}(i + 2, j))$$

$$\circ \quad \circ \quad \circ \quad \circ \quad \circ$$

$$\mathcal{M}(j - 1, j) = pk_j(N_j, Fm_j, Fr_j, sk_j) \parallel sk_j(\mathcal{S}(m)) \tag{4}$$

where for $l = i + 1, . . . , j$, N_l is a nonce, Fm_l is a message flag, Fr_l is a recipient flag, sk_l is the secret key used for one time message encryption, and || stands for message concatenation.

When the message packet is received by the node A_{i+1}, it decrypts the first block of received message using its private key corresponding to y_{i+1}. Then the node will get the recipient flag and message flag which give instructions for the subsequent actions.

When a message reaches the targeted recipient, to ensure traffic balance, the node will generate a dummy message to its subsequent nodes. Only the special nodes can terminate or initiate a dummy message. In this way, the amount of traffic flow that a node creates as the initiator is concealed in the traffic that it forwards since the overall traffic that it receives is the same as the traffic that it forwards. Also the message is encrypted with the public key that only be recovered by the recipient. While the intermediate nodes can only view the instruction of the message allowed. The sender's message is indistinguishable by other nodes. The sender and the recipient are thus hidden amongst the other nodes. It is infeasible for the adversary to correlate messages using traffic analysis and timing analysis due to message encryption.

4.3 Dynamic Local MANET Formation

Due to node mobility in the MANET, the local MANET will dynamically change over time. This makes reforming of the local MANET an essential part of our proposed scheme. The dynamic updating of the MANET can be characterized through mobility of each individual node that can leave and join a local MANET. The process for a node to join a local MANET, and a node to leave a local MANET are similar which are defined in [7].

4.4 Anonymous Communications between Two Arbitrary Special Nodes

Anonymous authentication allows two nodes in the same group to authenticate each other secretly in the sense that each party reveals its group membership to the other party only if the other party is also a group member.

The scheme consists of a set of special nodes and an administrator who creates groups and enrolls special nodes in groups. For this purpose, the administrator will assign each special node A a set of pseudonyms $\mathrm{PS}_1^{\mathrm{A}}, \mathrm{PS}_2^{\mathrm{A}}, \ldots. \mathrm{PS}_\tau^{\mathrm{A}}$, where τ is a large security parameter. In addition, the administrator also calculates a corresponding secret set $\{\alpha^{\mathrm{K_G h(PS_1^A)}} \mathrm{mod}\ \mathrm{p}, \ldots\ldots\ \ldots. \alpha^{\mathrm{K_G h(PS_\tau^A)}} \mathrm{mod}\ \mathrm{p}\}$ for special node A, where $\mathrm{K_G}$ is the group's secret and h is a hash function. The pseudonyms will be dynamically selected and used to substitute the real IDs for each communication. This means that two special nodes A and B can know each other's group membership only if they belong to the same group. When the special node A wants to authenticate to the special node B, the following secret handshake can be conducted:

(1) $A \rightarrow B$: Special node A randomly selects an unused pseudonym $\mathrm{PS_i^A}$ and a random nonce N_1, then sends $\mathrm{PS_i^A}$, N_1 to special node B;

(2) $B \rightarrow A$: Special node B randomly selects an unused pseudonym $\mathrm{PS_j^B}$ and a random nonce $N2$, then sends $\mathrm{PS_i^B}$, N_2, $V_0 = h(K_{BA} \| \mathrm{PS_i^A} \| \mathrm{PS_j^B} \| \mathrm{N_1} \| N_2 \| 0)$ to special node A, where $K_{BA} = \alpha^{K_G h(\mathrm{PS_i^A}).\mathrm{h}(\mathrm{PS_j^B})} mod\ p$;

(3) $A \rightarrow B$: Special node A sends $V_1 = h(K_{AB} \parallel \mathrm{PS}_i^A \parallel \mathrm{PS}_j^B \parallel \mathrm{N}_1 \parallel N_2 \parallel 1)$ to special node B, where $K_{BA} = \alpha^{K_G h(\mathrm{PS}_j^B).\mathrm{h}(\mathrm{PS}_i^A)} mod\ p$;

Since $K_{BA}= K_{AB}$, A can verify V_0 by checking whether $V_0 = h(K_{AB} \parallel \mathrm{PS}_i^A \parallel \mathrm{PS}_j^B \parallel \mathrm{N}_1 \parallel N_2 \parallel 0)$. If the verification succeeds, then A knows that B is an authentic group peer. Similarly, B can verify A by checking whether $V_1 = h(K_{AB} \parallel \mathrm{PS}_i^A \parallel \mathrm{PS}_j^B \parallel \mathrm{N}_1 \parallel N_2 \parallel 1)$. If the verification succeeds, then B knows that A is also an authentic group peer. However, in this authentication process, neither special node A, nor special node B can get the real identity of the other node. In other words, the real identities of special node A and special node B remain anonymous after the authentication process.

4.5 Anonymous Communication between Two Arbitrary Ordinary Nodes

The sender first randomly selects a local special node and transmits the message as discussed above. On receiving the message, the local special node first determines the destination MANET ID by checking the message recipient flag Fr. If it is 0, then the recipient and the special node are in the same MANET. Otherwise they are in a different MANET. The communication is done using the procedure discussed above.

While providing message recipient anonymity, the message can be encrypted to achieve confidentiality. The proposed anonymous communication is quite general and can be used in a variety of situations for communication anonymity in MANET including anonymous file sharing.

5 Security Analysis

We study several attacks designed [7] to analyze the security of the privacy preserving communication protocol.

5.1 Anonymity

(I) It is computationally infeasible for an adversary to identify the message sender and recipient on the local MANET for the following reasons:

i) Since the number of message packages that each mode receives from its immediate predecessors is the same as the number of packets that it forwards to its immediate successor, the adversaries cannot determine the message source based on the traffic volume or the number of message packets.

ii) since the message packets are encrypted using either public keys or the shared secrets keys of the immediate nodes, no adversary is able to distinguish the real meaningful menace from the dummy menace in the transmission in any of network nodes due to the traffic balance property and message contact encryption. Therefore the adversary cannot distinguish initiator traffic from the indirection traffic and learn whether the node is a recipient, a receiver, or simply a node that provides message forward service.

Hence the privacy preserving communication protocol provides to the sender and recipient anonymity in the local MANET.

(II) The proposed communication protocol offers both message sender and recipient anonymity among any two special nodes.

As told earlier, each special node is being assigned a large set of pseudonyms. A dynamically selected pseudonym will be used for any two ordinary nodes in different MANETs to communicate anonymously. The pseudonyms do not carry user information implicitly. The communication can be broken into three segments:

- the communication between the sender and local special node in the message sender‘s local MANET
- the communication between the special nodes in the corresponding MANETs
- the communication between the recipient special node and the receiver.

(I) has assured the communication anonymity between a special node and an ordinary node in the local MANETs. Therefore we only need to ensure anonymity between two special nodes in different MANETs in order to achieve full anonymity between the sender and receiver.

5.2 Impersonation Attacks

As told above, the forgery attack performed by an adversary, to carry out an impersonation attack is infeasible. For an adversary to forge as a special node, he needs to authenticate himself with a special node A. For this the adversary $\mathcal{A}$ needs to compute $\alpha^{K_G PS^{\mathcal{A}}.PS_i^A}\ mod\ p,$ where $PS^{\mathcal{A}}$ is the identity of the adversary and PS_i^A is the i^{th} pseudonym of the special node A.

However, since the adversary does not know the master secret K_G, he is unable to compute $\alpha^{K_G PS^{\mathcal{A}} PS_i^A}\ mod\ p$ and impersonate as a special node.

5.3 Message Replay Attacks

According to (4), each message packet in communication has a unique one-time session ID (nonce) to protect it from being modified or replayed. In addition, these fields are encrypted using the intermediate receiver nodes’ public key so that only the designated receiver nodes can decrypt the message. In this fashion, each packet transmitted across different MANETs bears different and uncorrelated IDs and content for adversaries.

Even if the same message is transmitted multiple times, the adversary still cannot link them together without knowing all the private keys of the intermediate nodes.

6 Performance Analysis

In this section, we will provide experimental results based on complexity of our proposed unconditionally secure privacy preserving Message authentication scheme (PPMAS) compared with source anonymous message authentication scheme (SAMAS) discussed in [7]. Our proposed method PPMAS has an advantage that we can send more than one anonymous message without changing the secret exponents, where as in the method discussed in [7], for every anonymous message transmission it has to change the secret exponents.

For a single node in a ring with n nodes, to generate m messages, with SAMAS the complexity is m (n-1), whereas with PPMAS the complexity is (n-1). The result is shown in Figure. 1. For all nodes, in a ring with n nodes, to generate m messages each, with SAMAS the complexity is mn (n-1) computations, whereas with PPMAS the complexity is (n(n-1) + mn) computations. The result is shown in Figure. 2.

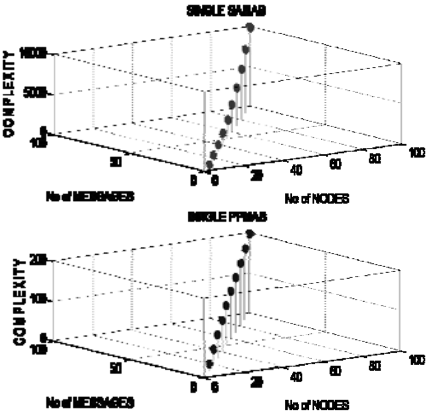


Fig. 1. Complexity measure of SAMAS versus PPMAS

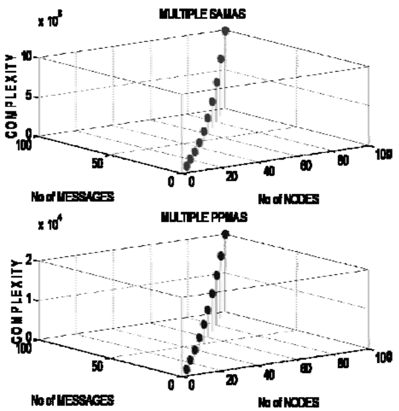


Fig. 2. Complexity measure of SAMAS versus PPMAS

7 Conclusion

In this paper, we first propose an efficient unconditionally secure privacy preserving message authentication scheme (PPMAS) that can be applied to any messages and any no. of messages without changing the secret exponents. PPMAS ensures message sender privacy along with message content authenticity. To ensure communication privacy without effecting transmission delay and collusion problems, we then propose a new and efficient privacy-preserving communication protocol for MANET that can provide both message sender and recipient privacy protection. Security analysis shows that the proposed protocol is secure against various attacks. The performance analysis results show that the proposed protocol is efficient and practical. It can be applied for several applications like secure routing protection and file sharing.

References

1. Pfitzmann, A., Hansen, M.: Anonymity, unlinkability, unobservability, pseudonymity, and identity management proposal for terminology (February 2008), http://dud.inftu-dresden.de/literatur/AnonTerminologyv0.31.pdf

2. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM 24, 84–88 (1981)
3. Khadir, O.: New Variant of ElGamal Signature Scheme. Int. J. Contemp. Math. Sciences 5(34), 1653–1662 (2010)
4. Chaum, D.: The dining cryptographers problemml: Unconditional sender and recipient untraceability. Journal of Cryptology 1(1), 65–75 (1988)
5. Reed, M., Syverson, P., Goldschlag, D.: Anonymous connections and onion routing. IEEE Journal on Selected Areas in Communications 16(4), 482–494 (1998)
6. Waidner, M.: Unconditional Sender and Recipient Untraceability in Spite of Active Attacks. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 302–319. Springer, Heidelberg (1990)
7. Ren, J., Li, Y., Li, T.: SPM: Source Privacy for Mobile Ad Hock Networks. EURASIP Journal on Wireless Communications and Networking 2010, article ID 534712, 10 pages (2010)
8. Reed, M.G., Syverson, P.F., Goldschlag, D.M.: Anonymous Connections and Onion Routing. Journal on Selected Areas in Communication Special Issue on Copyright and Privacy Protection (1998)
9. Danezis, G., Dingledine, R., Mathewson, N.: Mixminion: Design of a type III anonymous remailer protocol. In: Proc. of the IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, Calif, USA, pp. 2–15 (May 2003)
10. Möller, B.: Provably Secure Public-Key Encryptionfor Length-Preserving Chaumian Mixes. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 244–262. Springer, Heidelberg (2003)
11. Shields, C., Levine, B.N.: A protocol for anonymous communication over the Internet. In: Gritzalis, D. (ed.) Proc. of the 7th ACM Conference on Computer and Communication Security. ACM Press, Athens (2000)
12. Rivest, R.L., Shamir, A., Tauman, Y.: How to Leak a Secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001)
13. Reiter, M., Rubin, A.: Crowds: anonymity for web transaction. ACM Transactions on Information and System Security 1(1), 66–92 (1998)
14. ElGamal, T.A.: A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory 31(4), 469–472 (1985)
15. Goldwasser, S., Micali, S., Rivest, R.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing 17, 281–308 (1988)
16. Pointcheval, D., Stern, J.: Security Proofs for Signature Schemes. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 387–398. Springer, Heidelberg (1996)
17. Reiter, M.K., Rubin, A.D.: Crowds: Anonymity for Web Transactions. ACM Transactions on Information and System Security 1(1), 66–92 (1998); Symposium on Security & Privacy, Oakland, Calif, USA (May 2003)
18. Pfitzmann, A., Köhntopp, M.: Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology. In: Federrath, H. (ed.) Anonymity 2000. LNCS, vol. 2009, pp. 1–9. Springer, Heidelberg (2001), doi:10.1007/3-540-44702-4_1

Circle Formation by Asynchronous Fat Robots with Limited Visibility

Ayan Dutta[1], Sruti Gan Chaudhuri[2],
Suparno Datta[1], and Krishnendu Mukhopadhyaya[2]

[1] Computer Science and Engineering Department,
Heritage Institute of Technology, Kolkata - 700107, India
[2] Advanced Computing and Microelectronics Unit,
Indian Statistical Institute, Kolkata
203 B.T. Road, Kolkata - 700108, India
{duttaayan.cs,suparno.datta}@gmail.com,
{sruti_r,krishnendu}@isical.ac.in

Abstract. This paper proposes a distributed algorithm for circle formation by multiple autonomous mobile robots. The vision of each robot is limited to a maximum distance. The robots do not store past actions or records of past data. They are anonymous and cannot be distinguished by their appearances. All robots agree on a common origin and axes. Earlier works report algorithms for *gathering* of multiple autonomous mobile robots in limited visibility considering the robots to be dimensionless or points. This paper models a robot as a unit disc (*fat robot*). The algorithm presented in this paper also assures that there is no collision among the robots. The robots do not share a common clock. They execute the algorithm asynchronously.

Keywords: Asynchronous, Limited Visibility, Fat Robot, Transparent, Circle Formation.

1 Introduction

The field of cooperative mobile robotics has received a lot of attention from various research groups in institutes as well as industries. A primary focus of these research and development activities is the distributed motion coordination which allows the robots to form certain patterns [12]. Motion planning algorithms for robotic systems are very challenging due to severe limitations, such as, communication between the robots, hardware constraints, obstacles etc. The significance of positioning the robots based on some given patterns may be useful for various tasks, e.g., operations in hazardous environments, space mission, military operations, tumor excision[9] etc. In addition, formation of patterns and flocking of a group of mobile robots are also useful for providing communication in ad-hoc mobile networks. Pattern formation by cooperative mobile robots involves many geometric issues [12]. This paper addresses one such geometric problem, circle formation. Each robot is capable of sensing its immediate surrounding, performing computations on the sensed data, and moving towards

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 83–93, 2012.

the computed destination. The robots are free to move on the 2D plane. They are anonymous, i.e., they cannot be distinguished by a unique identity or by their appearances. They are unable to remember past actions. Furthermore, the robots are unable to communicate directly and can only interact by observing the positions of other robots. A robot can observe other robots around itself within a fixed distance. Based on this model, we study the problem of the mobile robots positioning themselves to form a circle of a given radius and center. The robots are represented as unit discs (also known as *fat robots* [2]).

2 Earlier Works

A large body of research work exists in the context of multiple autonomous mobile robots exhibiting cooperative behavior. The aim of such research is to study the issues as group architecture, resource conflict, origin of cooperation, learning and geometric problems [14]. The computational model popular in literature for mobile robot is called the *weak model* [5]. Under this model, robots are considered to be points which move on the plane. Each robot executes a cycle consisting of three phases, *look-compute-move* described as follows.

- *Look*: Determine the current configuration by identifying the location of all visible robots and marking them on the robot's private coordinate system.
- *Compute*: Based on the location of all the visible robots, compute a location T, where the robot should move now.
- *Move*: Travel towards the point T.

The robots do not communicate directly. They can only interact by observing others positions (observed in *Look* phase). The robots may execute the cycle synchronously (the robots are driven by a common clock and are active in every cycle), semi-synchronously (the robots are driven by a common clock and are not active in every cycle) or asynchronously (the clock cycles are independent).

Formation of circle by multiple autonomous mobile robots is defined as follows - *A set of robots is given. The robots are asked to position themselves on the circumference of a circle in finite time.* Sugihara and Suzuki [12] proposed a simple heuristic algorithm for the formation of an approximation of a circle under limited visibility. Suzuki and Yamashita[13] proposed a regular polygon formation algorithm for non-oblivious robots. Défago and Konogaya [3] came up with an improved algorithm by which a group of oblivious robots eventually form a circle. The above solutions assume semi-synchronous (SSM)[13] model in which the cycles of all robots are synchronized and their actions are atomic. Katrenaik [8] proposed an algorithm to form a biangular circle[1] under asynchronous CORDA [10] model. Défago and Souissi [4] presented an algorithm where a group of mobile robots, starting from any arbitrary configuration, can be self-organized to form a circle. The robots do not have a global coordinate system nor they share

[1] In a biangular circle, there is a center and two non zero angles α and β such that the center between each two adjacent points is either α or β and these angles alternate.

the knowledge of the coordinate systems of the other robots. However, robots agree on the chirality of the system (i.e., clockwise/counterclockwise orientation). During its observation, a robot obtains the positions of all robots according to its own local coordinate system. In this algorithm, it is assumed that each robot has full visibility of all other robots. They also do not obstruct the view of each other. All of these algorithms assume that a robot is a point and it neither creates any visual obstruction nor acts as an obstacle in the path of other robots. Obviously, such robots are not practical. However small a robot might be, it must have certain dimensions. Czyzowicz et al.,[2] extend the traditional *weak model* [5] of robots by replacing the point robots with unit disc robots. They named these robots as *fat robots*. Gan Chaudhuri and Mukhopadhyaya [7] proposed an algorithm for gathering multiple *fat robots*. Many of the previous circle formation algorithms required the system to be synchronous which is also an ideal situation. *We propose a circle formation algorithm which assures the formation of circle by asynchronous fat robots.*

Most of the earlier works considered that the robots have unlimited visibility range, i.e., a robot can see infinite radius around itself. Ando et al. [1] proposed a point convergence algorithm for oblivious robots with limited visibility. They assume that the motion of the robots is instantaneous. The collision issue has been ignored. They also simulate the model under realistic environment. Later Flocchini et al. [6] presented a gathering algorithm for asynchronous, oblivious robots in limited visibility having common coordinate system. Souissi et al. [11] studied the solvability of gathering with limited visibility under semi-synchronous model of robots using unreliable compass. They assume that the compass are unstable for some arbitrary long periods and stabilize eventually. Circle formation for *fat robots* in limited visibility is not yet reported. This problem is challenging because of the issue of finding collision free path for the robots with limited visibility. This paper presents a distributed circle formation algorithm for asynchronous, oblivious, *fat robots* having limited visibility.

3 Circle Formation by Asynchronous Fat Robots with Limited Visibility

In this section, we first describe the robot model used in this paper and present an overview of the problem. Then we move to the solution approach and present the algorithms with the proofs of their correctness.

3.1 Underlying Model

We use the basic structure of *weak model* [5] of robots and add some extra features which extend the model towards reality. Let $R = \{r_1, r_2, \ldots, r_n\}$ be a set of *fat robots*. A robot is represented by its center, i.e., by r_i we mean a robot whose center is r_i. The set of robots R deployed on the 2D plane is described as follows:

- The robots are autonomous.
- The robots have a common origin, common x-y axis, common sense of direction and common unit distance.

- Robots are anonymous and homogeneous in the sense that they are unable to uniquely identify themselves, neither with a unique identification number nor with some external distinctive mark (e.g. color, flag).
- The robots are oblivious in the sense that they can not remember any past action.
- A robot is a fat robot and represented as a unit disc.
- A robot can see up to a fixed distance around itself on the 2D plane.
- CORDA[10] model is assumed for the robots. Under this model a robot in motion is visible to other robots.
- Robots can not communicate explicitly. The robots communicate only by means of observing other robots within its visibility range.
- Each robot executes a cycle of *look* – *compute* – *move* asynchronously.

3.2 Overview of the Problem

A set of robots R (as described above) is given. Our objective is to form a circle (denoted by CIR) of radius k and centered at C by moving the robots from R. Following assumptions and definitions are used in this paper.

Definition 1. *Each robot can see up to a fixed distance around itself. This distance is called visibility range of that robot. The visibility range of $r_i \in R$ is denoted by R_v. Visibility Range is equal for all robots in R (Fig. 1).*

Definition 2. *The circle, centered at robot $r_i \in R$ and having radius R_v is called the visibility circle of the robot r_i, denoted by $VC(r_i)$. r_i can see everything within and on the circumference of $VC(r_i)$. r_i cannot see beyond $VC(r_i)$ (Fig. 1).*

In Fig. 1, r_i can see r_j and r_k (since, these are inside and on $VC(r_i)$) but can not see r_l (since, it is outside $VC(r_i)$).

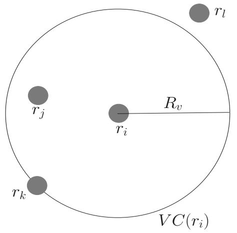


Fig. 1. Visibility Radius (R_v) and Visibility Circle ($VC(r_i)$)

Assumptions:

- All robots in R agree on a common origin, axes, sense of direction and unit distance. C, the center of the circle CIR is considered as the origin of the coordinate system.
- The radius (k) of the circle to be formed (CIR) is given. The length of k is such that CIR can accommodate all the robots in R.
- Initially all robots in R are inside CIR. No robot is on the circumference of CIR.

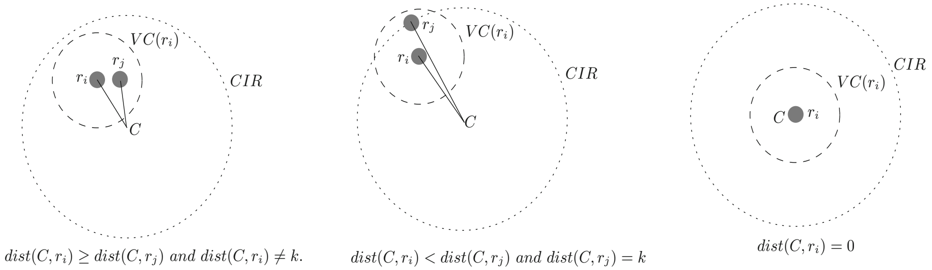
3.3 Destination Computation

This section presents the algorithm $Compute\_destination(r_i)$ for computing the destination for the robot $r_i \in R$. The following notations are used in the algorithm as well as throughout the paper.

- $dist(p1, p2)$: Euclidean distance between two points $p1$ and $p2$.
- $cir(r_i, C)$: A circle centered at C and having radius of $dist(r_i, C)$.
- $projpt(r_i, A)$: The intersection point where the ray starting at the center of the circle A and passing through r_i intersects A.
- $arc(a, b)$: The arc of a circle between the points a and b on the circumference of that circle.
- $T(r_i)$: Destination point for robot $r_i \in R$. This is the output of the algorithm $Compute\_destination(r_i)$.

Definition 3. *If the circular area of radius* 2 *and centered at point* p *does not contain the center of any other robot, then* p *is called a vacant point.*

Two constraints have been put on the movement of any robot $r_i \in R$.



$dist(C, r_i) \geq dist(C, r_j)$ and $dist(C, r_i) \neq k$.

$dist(C, r_i) < dist(C, r_j)$ and $dist(C, r_j) = k$

$dist(C, r_i) = 0$

Fig. 2. An example to represent Constraint 1

Constraint 1. *Let* $r_j \in R$ *be any robot inside* $VC(r_i)$ *(the visibility circle of* r_i*).*

- *If* $dist(C, r_i) \geq dist(C, r_j)$ *and* $dist(C, r_i) \neq k$*, then* r_i *is eligible to move.*
- *If* $dist(C, r_i) < dist(C, r_j)$ *and* $dist(C, r_j) = k$*, then* r_i *is eligible to move.*
- *If* r_i *is at* C*, then* r_i *is eligible to move.*
- *For all other cases* r_i *will not move.*

Constraint 2. *The robot* $r_i \in R$ *moves only in any of the following fixed directions.*

- *Radially outwards following the ray starting from* C *and directed towards* r_i*.*
- *Right side of the ray, starting from* C *and directed towards* r_i*.*

We categorize different configurations depending on the positions of the visibility circles of two robots r_i and r_j. We denote these configurations as Φ_1, Φ_2, Φ_3 and Φ_4 .

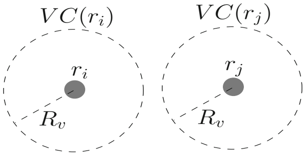
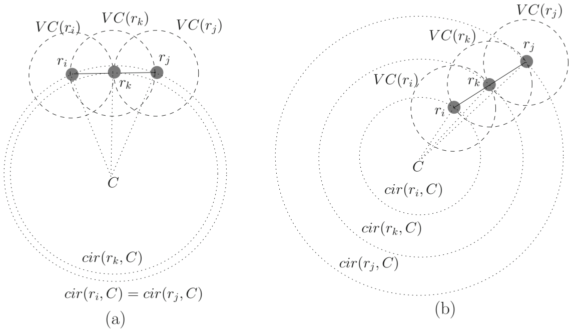


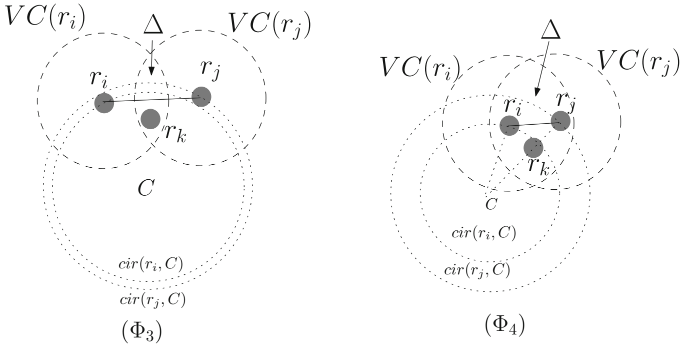
Fig. 3. An example of the configuration Φ_1



(a)

(b)

Fig. 4. An example of the configuration Φ_2



(Φ_3)

(Φ_4)

Fig. 5. An example of the configurations Φ_3 and Φ_4

- Φ_1: $V(r_i)$ and $V(r_j)$ do not touch or intersect each other (Fig. 3). Here $dist(r_i, r_j) > 2R_v$ and r_i and r_j can not see each other.
- Φ_2: $V(r_i)$ and $V(r_j)$ touch each other at a single point say q. Here $dist(r_i, r_j) = 2R_v$ (Fig. 4) and r_i and r_j can not see each other. If there is a robot at q say r_q, then r_i & r_q and r_j & r_q are mutually visible.
- Φ_3: $V(r_i)$ and $V(r_j)$ intersect each other at two points such that $dist(r_i, r_j) < 2R_v$ and $dist(r_i, r_j) > R_v$ (Fig. 5). r_i and r_j can not see each other. Let Δ be the common visible region of r_i and r_j. If there is a robot in the region Δ, say r_k, then r_k can see r_i and r_j, and both r_i and r_j can see r_k.

- Φ_4: $V(r_i)$ and $V(r_j)$ intersect each other at two points such that $dist(r_i, r_j) \leq R_v$. (Fig. 5). r_i and r_j can see each other. Let Δ be the common visible region of r_i and r_j. If there is a robot in the region Δ, say r_k, then r_k, r_i and r_j can see each other.

There are five configurations depending on the position of $r_i \in R$ inside the circle CIR. We denote these configurations as Ψ_1, Ψ_2, Ψ_3, Ψ_4 and Ψ_5.

- Ψ_1: r_i is on the circumference of CIR (Fig. 6).

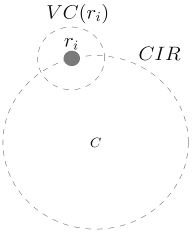


Fig. 6. An example of the configuration Ψ_1

- Ψ_2: $VC(r_i)$ touches CIR (at some point say h) (Fig. 7).

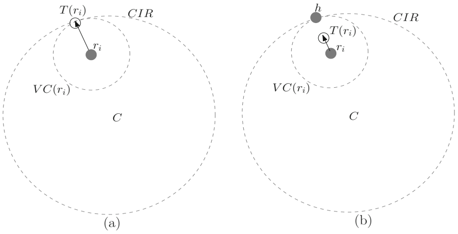


Fig. 7. An example of the configuration Ψ_2

- Ψ_3: r_i is not at C and $VC(r_i)$ does not touch or intersect the circumference of CIR (Fig. 8).

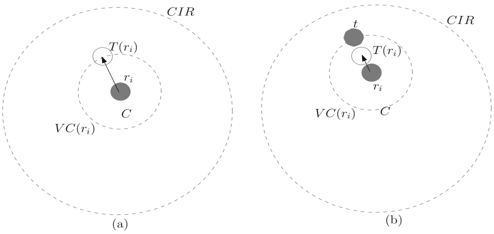


Fig. 8. An example of the configuration Ψ_3

- Ψ_4: r_i is at C (Fig. 9).
- Ψ_5: $VC(r_i)$ intersects CIR (at two points say g and l) (Fig. 10).

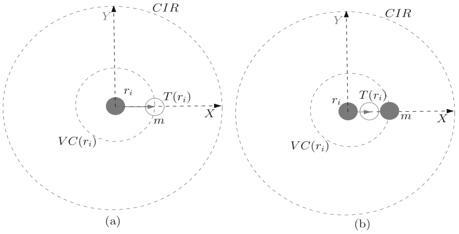


Fig. 9. An example of the configuration Ψ_4

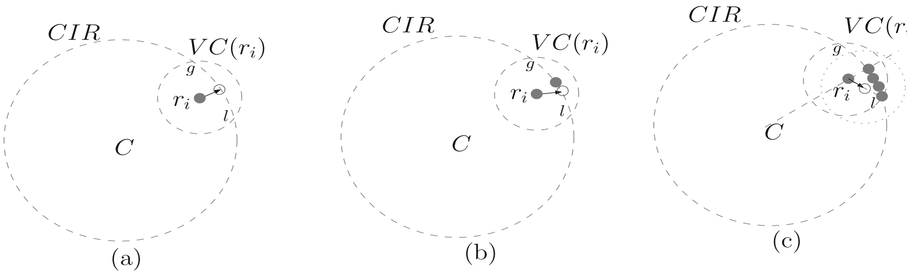


Fig. 10. An example of the configuration Ψ_5

Algorithm 1. Compute_destination(r_i)

```
Input: (i) A set of robots $R = \{r_1, \ldots, r_n\}$ (ii) A robot $r_i \in R$.
Output: The destination for $r_i$, $T(r_i)$ such that $T(r_i)$ is deterministic.
switch()
Case 1. ($r_i$ is in $\Psi_1$) (Fig. 6)
$r_i$ does not move;
Case 2. ($r_i$ is in $\Psi_2$)(Fig. 7)
$h \leftarrow$ point where $VC(r_i)$ touches $CIR$;
if $h$ is a vacant point then
    $T(r_i) \leftarrow h$ (Fig. 7(a));
else
    $T(r_i) \leftarrow$ midpoint of the line joining $r_i$ and $h$(Fig. 7(b));
Case 3. ($r_i$ is in $\Psi_3$)(Fig. 8)
$t \leftarrow projpt(r_i, VC(r_i))$;
if $t$ is a vacant point then
    $T(r_i) \leftarrow t$ (Fig. 8(a));
else
    $T(r_i) \leftarrow$ midpoint of line joining $r_i$ and $t$ (Fig. 7(b));
Case 4. ($r_i$ is in $\Psi_4$)(Fig. 9)
$m \leftarrow$ Intersection point of positive $X$ axis of robot $r_i$ and $VC(r_i)$;
if $m$ is a vacant point then
    $T(r_i) \leftarrow m$ (Fig. 9(a));
else
    $T(r_i) \leftarrow$ midpoint of the line joining $r_i$ and $m$(Fig. 9(b));
Case 5. ($r_i$ is in $\Psi_5$)(Fig. 10)
$g \leftarrow$ point where $VC(r_i)$ intersects $CIR$ at left side of $r_i$ ;
$l \leftarrow$ points where $VC(r_i)$ intersects $CIR$ at right side of $r_i$ ;
$t \leftarrow projpt(r, C)$;
if $t$ is a vacant point then
    $T(r_i) \leftarrow t$ (Fig. 10(a)) ;
else
    if there is vacant point(s) on the arc($tl$) then
        $T(r_i) \leftarrow$ first vacant point on the arc($tl$) (Fig. 10(b));
    else
        $T(r_i) \leftarrow$ the midpoint of the line joining $r_i$ and $l$ (Fig. 10(c));
return $T(r_i)$;
```

Definition 4. *The destination $T(r_i)$, computed by the robot r_i is called a deterministic destination if no other robot in R computes the same destination $T(r_i)$ for itself.*

Correctness of Compute_destination(r_i): The following lemma and observations prove the correctness of the algorithm *Compute_destination*(r_i).

Observation 1. *r_i never goes outside of $VC(r_i)$.*

Observation 2. *If two robots r_i and r_j are in Φ_1 (Fig. 3), their movements are not affected by each other.*

Proof. If r_i and r_j are in Φ_1, then r_i and r_j can not see each other. Following constraint 1 and 2, r_i and r_j execute the algorithm and find deterministic destinations of their own. (Fig. 3). Hence, the movements of r_i and r_j are not affected by each other. □

Observation 3. *If two robots r_i and r_j are in Φ_2 (Fig. 4), and there is a robot (say r_k) at the touching point of $VC(r_i)$ and $VC(r_j)$ (r_k is visible by both r_i and r_j), then the movements of r_k, r_i and r_j are not affected by each other.*

Proof. $VC(r_i)$ and $VC(r_j)$ touch at one point. Let r_k be at the touching point. r_k is on the straight line joining r_i and r_j. r_k can see both r_i and r_j. r_i and r_j both can see r_k. However r_i and r_j are not mutually visible.

Let us consider the case in Fig. 4(a). Here $dist(C, r_i) = dist(C, r_j)$ and $dist(C, r_i), dist(C, r_j) > dist(C, r_k)$. Following constraint 1, r_k will not move. However, both r_i and r_j will move to their own deterministic destinations considering constraint 1 and 2. Let us consider the case in Fig. 4(b). Here r_j, r_k and r_i will move one by one following constraint 1 and 2. Hence, the movements of r_k, r_i and r_j are not affected by each other. □

Observation 4. *If two robots r_i and r_j are in Φ_3 (Fig 5), and there is any robot (say r_k) in common visible region (r_k is visible by both r_i and r_j), namely Δ, the movements of r_k, r_i and r_j are not affected by each other.*

Proof. $VC(r_i)$ and $VC(r_j)$ intersect each other and there is a robot r_k in the common visibility region Δ. r_k can see both r_i and r_j. r_i and r_j can see r_k but r_i and r_j can not see each other (Fig 5).

If r_k is on the line joining r_i and r_j or below the line in Δ. Following constraint 1, r_k will not move. r_i and r_j will move to their own deterministic destinations using constraint 1 and 2.

If r_k is above the line joining r_i and r_j in Δ, then following constraint 1, r_k is eligible to move and r_i or r_j will not move. □

Observation 5. *If two robots r_i and r_j are in Φ_4 (Fig 5), (they are mutually visible) and there is any robot (say r_k) in common visible region Δ (r_k is visible by both r_i and r_j), then the movements of r_k, r_i and r_j are not affected by each other.*

Proof. $VC(r_i)$ and $VC(r_j)$ intersect each other in such a way that r_i and r_j are mutually visible. r_k is in Δ. Therefore, r_i, r_j and r_k can see each other. Since r_i, r_j and r_k are mutually visible, following constraint 1 and 2, the robots will move to their deterministic destinations. □

Lemma 1. *The destination $T(r_i)$ computed by the robot $r_i \in R$ using Compute_destination(r_i) is deterministic.*

Proof. Follows from observations 1, 2, 3, 4 and 5. □

3.4 Circle Formation Algorithm

Each robot executes the algorithm $Circle\_formation(r_i)$ and places itself on the circumference of CIR after finite number of executions of the algorithm. Each robot computes its destination using $Compute\_destination(r_i)$ and moves to the destination.

Algorithm 2. Circle_formation(r_i)

```
Input: (i) A set of robots $R$ (ii) A robot $r_i \in R$
Output: $r_i$ placed on the circumference of $C$ after a finite number of
        executions of the algorithm.
if $r_i$ is not eligible to move according to constraint 1 then
    $r_i$ does not move;
else
    $T(r_i) \leftarrow Compute\_destination(r_i)$;
    $r_i$ moves to $T(r_i)$;
```

Correctness of Circle_formation(r_i): When a robot r_i finds its destination using $Compute\_destination(r_i)$, it moves towards the destination in a straight line. Constraint 1 and 2 assures that no robot appears in the path[2] of r_i. Hence, no collision occurs.

Lemma 2. *The path of each robot is obstacle free.*

Proof. Follows from observations 1, 2, 3, 4 and 5. □

Lemma 1 also ensures that no two robots will have same destination. This also assures the collision free path for the robots. Finally we state the following theorem.

Theorem 1. *$Circle\_formation(r_i)$ forms circle CIR by R in finite time.*

4 Conclusion

In this paper, a distributed algorithm is presented for circle formation by autonomous, oblivious, homogeneous, non communicative, asynchronous, *fat robots* having limited visibility. The algorithm ensures that multiple mobile robots will form a circle of given radius and center in finite time. The algorithm also ensures the collision free path for each robot.

[2] The line starting from the current position of the robot and ending at its destination.

Acknowledgement. S Gan Chaudhuri would like to acknowledge the Council of Scientific and Industrial Research (CSIR), Govt. of India, for their assistance to continue the research work.

References

1. Ando, H., Suzuki, I., Yamashita, M.: Distributed memoryless point convergence algorithm for mobile robots with limited visibility. Trans. on Robotics and Automation 15(5), 818–828 (1999)
2. Czyzowicz, J., Gasieniec, L., Pelc, A.: Gathering Few Fat Mobile Robots in the Plane. Theoretical Computer Science 410, 481–499 (2009)
3. Défago, X., Konagaya, A.: Circle Formation for Oblivious Anonymous Mobile Robots with no Common Sense of Orientation. In: Proc. 2nd International Annual Workshop on Principles of Mobile Computing, pp. 97–104 (2002)
4. Défago, X., Souissi, S.: Non Uniform Circle Formation Algorithm for Oblivious Mobile Robots with Convergence Towards Uniformity. Theoretical Computer Science 396(1-3), 97–112
5. Efrima, A., Peleg, D.: Distributed Algorithms for Partitioning a Swarm of Autonomous Mobile Robots. Theoretical Computer Science 410, 1355–1368 (2009)
6. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Gathering of Asynchronous Oblivious Robots with Limited Visibility. In: Ferreira, A., Reichel, H. (eds.) STACS 2001. LNCS, vol. 2010, pp. 247–258. Springer, Heidelberg (2001)
7. Gan Chaudhuri, S., Mukhopadhyaya, K.: Gathering Asynchronous Transparent Fat Robots. In: Janowski, T., Mohanty, H. (eds.) ICDCIT 2010. LNCS, vol. 5966, pp. 170–175. Springer, Heidelberg (2010)
8. Katreniak, B.: Biangular Circle Formation by Asynchronous Mobile Robots. In: Pelc, A., Raynal, M. (eds.) SIROCCO 2005. LNCS, vol. 3499, pp. 185–199. Springer, Heidelberg (2005)
9. Payton, D., Estkowski, R., Howard, M.: Pheromone Robotics and the Logic of Virtual Pheromones. In: Şahin, E., Spears, W.M. (eds.) Swarm Robotics WS 2004. LNCS, vol. 3342, pp. 45–57. Springer, Heidelberg (2005)
10. Prencipe, G.: *InstantaneousActions* vs. *FullAsynchronicity*: Controlling and Coordinating a Set of Autonomous Mobile Robots. In: Restivo, A., Ronchi Della Rocca, S., Roversi, L. (eds.) ICTCS 2001. LNCS, vol. 2202, pp. 154–171. Springer, Heidelberg (2001)
11. Soussi, S., Défago, X., Yamashita, M.: Using Eventually Consistent Compasses to Gather Memory-Less Mobile Robots with Limited Visibility. ACM Transactions on Autonomous and Adaptive Systems 4(1), Article 9 (January 2009)
12. Sugihara, K., Suzuki, I.: Distributed Motion Coordination of Multiple Mobile Robots. In: Proc. IEEE International Symposium on Intelligent Control, pp. 138–143 (1990)
13. Suzuki, I., Yamashita, M.: Distributed Anonymous Mobile Robots: Formation of Geometric Patterns. SIAM Journal of Computing 28(4), 1347–1363 (1999)
14. Uny Cao, Y., Fukunaga, A.S., Kahng, A.B.: Cooperative Mobile Robotics: Antecedents and Directions. Autonomous Robots (4), 1–23 (1997)

High Concurrency for Continuously Evolving OODBMS

V. Geetha and N. Sreenath

Dept. of Information Technology, Dept. of Computer Science & Engg
Pondicherry Engineering College
Puducherry – 605014
vgeetha@pec.edu

Abstract. OODBMS is widely favored for mapping domains like CAD, with complex representation requirements. The transactions accessing OODBMS can be categorized into runtime transactions and design time transactions. Runtime transactions are meant for accessing data and design time transactions are meant for accessing schema. Parallel execution of transactions is supported to promote concurrency and throughput. In continuously evolving business domains, frequent schema changes are done to reflect the changes of business domain. Then it implies that more number of design time transactions arrive parallely with runtime transactions. Concurrency control schemes are employed to maintain the consistency of the database. Several semantic multi-granular lock based concurrency control schemes have been proposed in the literature. They have the drawback of either poor performance or high maintenance overhead when applied to continuously evolving domains. This paper proposes semantic lock-based concurrency control mechanism with better performance and nil maintenance overhead for continuously evolving OODBMS. It uses lock rippling to improve the performance.

Keywords: Distributed object oriented databases, continuously evolving domains, concurrency control, multi granular lock model.

1 Introduction

Object Oriented Database System (OODBMS) is highly favored for building business applications that have complex representation requirements. OODBMS is a collection of objects. Objects are of two types namely classes and instances. The class objects define the structure of entities participating in the domain in the form of attributes and methods. Instances are defined by classes and map to the underlying data. A transaction in OODBMS is defined as a partially ordered set of method invocations on a class or an instance object [13]. Two types of transactions access the database namely runtime transaction and design time transaction.

The runtime transactions usually operate on the data to serve the clients' requests. They are executed by mapping the underlying data in the database on attributes of instances and the associated methods in the class operate on them. The structure of the business domain is represented using schema. In OODBMS, the schema is

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 94–105, 2012.

represented as a class diagram. So, design time transactions modify the class diagram to reflect the changes done in evolving business domain. The operations allowed in design time transactions are to add/delete/read/modify attribute definitions, method definitions, class definitions and class relationship definitions. In general, the access to data or schema can be read or write operations. The read operations are executed in shared mode while write operations are to be serialized to maintain consistency. The transactions can be in one of these granularities: - class lattice level, class level and instance level.

When several transactions are allowed to execute parallely, the consistency of the database is threatened. Hence, Concurrency control schemes are applied to synchronize the transactions. A good concurrency control mechanism must give maximum throughput while maintaining the consistency of the database. The throughput can be improved by maximizing the concurrency. Among the three popular concurrency control mechanisms such as locking, optimistic and time stamp ordering, locking is widely used due to its ease of implementation.

In lock based concurrency control scheme, a transaction has to acquire locks before accessing the database and release locks after use. A concurrency control scheme satisfies database consistency by enforcing a correctness criterion. Serializability is a widely used correctness criterion. Transactions are called *serializable,* if the interleaved execution of their operations produces the same results as that of a serial execution of the same transactions. In the lock-based scheme, the concurrency is determined by the concept of compatibility. Compatibility decides whether a transaction can be run concurrently with those in progress on the same object. Two transactions are compatible, if their execution orders do not affect the result. Otherwise, they are not compatible. Then, the newly arrived transaction is blocked. It has to wait until the transaction currently holding the resource releases the lock. Usually, transactions in CAD, CAM like applications using OODBMS, are long duration in nature. So, maximum concurrency is needed to improve the performance of the system.

Several lock based concurrency control mechanisms have been proposed in the literature. Multi-granular lock models are favored as they provide maximum concurrency. This is possible because the resource can be accessed in different granularities. As a result, deadlocks due to lock escalation are minimized. Initially locks based on object relationships like inheritance, aggregation and association are defined by extending Gray's multi-granular lock model [2] for RDBMS to OODBMS. They had the drawback of coarse granularity. Later, concurrency control using access vectors is introduced to further enhance the concurrency. They perform better for the stable domains. In such stable domains, the users make frequent runtime transactions to access the data. The design time transactions are few and are far in-between. However, in the case of continuously evolving domains, the class diagram has to be changed frequently. Then, more number of design time transactions will come with runtime transactions. Then, the access vectors should be updated every time a schema change is made. Because of this, the maintenance overhead is more. This introduced the need for a new concurrency control scheme to support evolving systems with nil or less overhead.

In this paper, the author proposes a multi-granular lock based concurrency control scheme using semantic lock modes for continuously evolving domains. The proposed scheme has these advantages. It is based on multi-granularity locking. As this scheme does not use any access vectors, it does not have any overhead of updating them every time the schema is changed. It does not need prior knowledge of structure of objects. Further the proposed work allows more parallelism between design time transactions and runtime transactions than existing works.

The paper is organized as follows. In the next section, the basic semantics and related works are reviewed. In section 3, a concurrency control scheme based on semantic lock modes is proposed. In section 4, the performance evaluation of the proposed scheme is presented. The paper concludes in section 5.

2 Basic Semantics and Literature Survey

2.1 Basic Semantics

The schema of OODB is represented as a class diagram. The class diagram is a collection of classes related by inheritance, aggregation and association relationships. Group of classes related by inheritance (excluding *multiple inheritance*) is called *class hierarchy*. Group of classes related by a combination of all types of relationships mentioned above is called *class lattice*. Then class diagram can be viewed as a class lattice and represented as Directed Acyclic Graph (DAG). The classes are viewed as nodes and the relationship links connecting classes are viewed as edges. The design time transactions can do changes to schema in two ways as specified in Kim et al.17] and Bannerjee et al. [11].

The schema changes are categorized into

1. Changes to the contents of node or class
   1.1 Changes to instances
   - Add a new instance to a class
   - Delete an existing instance from a class
   - Modify the definition of an instance
   - Move an instance from one class to another class
   - Read the definition of an instance

   1.2 Changes to attributes
   - Add a new attribute to a class
   - Delete an existing attribute from a class
   - Modify the definition of an attribute
   - Move an attribute from one class to another class
   - Read the definition of an attribute

   1.3 Changes to methods
   - Add a new method to a class
   - Delete an existing method from a class
   - Modify the definition of an method
   - Move a method from one class to another class
   - Read the definition of an method

2. Changes to an edge
   - Make a class S as superclass of class C
   - Delete a class S from the super class list of class C.
   - Modify the order of superclasses of class C
   - Read the superclass list of class C.
3. Changes to a node or class
   - Add a new class
   - Delete an existing class
   - Modify the definition of a class
   - Move a class from one location to another position
   - Read the definition of a class

From the above group of operations, certain semantic aspects can be inferred. During runtime transactions, the values of attributes are read or modified by executing the associated methods in a class. The attribute values are locked in read and write lock modes. In design time transactions, the attribute definitions are read or modified. Thus, attribute has two facets and they are chosen depending on the type of transaction.

During runtime transactions, the methods are locked in read mode, as their contents are not modified by execution. In design time transactions, the method definitions are read or modified. When any attribute or method definition is modified, runtime transactions accessing them should not be allowed.

A runtime transaction can have attribute, instance or class level of granularity. It is based on the property of the method as to whether the method is 1. *primitive* or *composed* and 2. *instance* or *class* level as defined by Reihle and Beczuck [1].

Further, it is pointed out in Geetha and Sreenath [16] that when runtime transaction requests a base class instance, it is enough to lock the base class instance alone. However, when a runtime transaction requests a sub class object, it is required to lock the associated base class objects that access the same record also to preserve database consistency. Thus, there is an upward dependency from sub classes to base class. This is applicable to aggregation and association also. In aggregation, the component objects are locked with composite objects. In association, associative objects are dependent on associated objects.

The operations allowed during schema changes mentioned above can be grouped into five types. They are:

1. Add a new attribute/ method/ instance/ class/ edge (relationship).
2. Delete an existing attribute/ method/ instance/ class/ edge (relationship).
3. Modify the definitions, values or implementation of attribute/ method/ instance/ class/ edge (relationship) as appropriate.
4. Read the definitions, values or implementation of attribute/ method/ instance/ class/ edge (relationship) as appropriate.
5. Move an existing attribute/ method/ instance/ class/ edge (relationship) from one location to another location.

From the above classes of operations, the dependency on creation, deletion and modification can be inferred. The attributes and methods can be added only to an existing class. Similarly instances can be created only after defining a new class and adding its attributes and methods. New relationships can be established only among existing classes. When a new class is defined, until it is related to the existing classes by a relationship edge, it can be parallely done with any other schema change without affecting consistency. Once the new class is included in the already existing class lattice as a base class or component class or associative class by adding an edge, then its attributes and methods have to be included in its subclasses, composite classes or associated classes respectively. Then it can be inferred that the possible granularities for addition operation is class level or sub class lattice level (group of related classes).

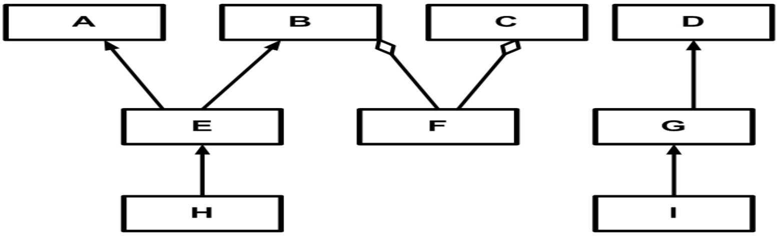


Fig. 1. Sample class diagram

A class can be deleted only after deleting its attributes, methods and instances. If the class is related to other classes by inheritance or aggregation or association, its attributes and methods are to be deleted from them. Any other schema change in the related sub class lattice should not be allowed when deletion is done. If the class is a base class or component class or associative class, then its attributes and methods have to be deleted in its subclasses, composite classes or associated classes respectively. For example in fig 1, the attributes in A inherited into E and H are to be deleted, while deleting A.

When a transaction reads the definitions of attribute/ method/ instance/ class/ relationship (edge), it can be done in parallel with all other read operations. It can also be noted that when an attribute definition is read, its value can be modified by a runtime transaction. When a design time transaction tries to move an attribute/ method/ instance/ class/ relationship (edge), then whole class diagram has to be locked in exclusive mode. This is because the move operation may involve the entire class lattice as the destination is unpredictable.

2.2 Existing Works

The existing concurrency control schemes can be assessed based on the level of concurrency they provide for parallel execution of design time and runtime transactions without compromising on consistency. These two types of transactions induce three different types of conflicts among transactions to a class: Conflicts among runtime transactions, Conflicts among design time transactions and Conflicts between runtime and design time transactions. Compatibility among transactions is defined in the literature in two ways namely based on relationships and based on commutativity.

In Garza and Kim [3], Kim et al. [4], Lee and Liou [7], Geetha and Sreenath [5], Jun and Gruenwald [6], Saha and Morrisey [10], the concurrency control is based on object relationships. These algorithms offer smallest granularity of object level. Most of the time, the database is accessed for data through attributes. In these works, the entire object is locked instead of an attribute. So, concurrency is limited. In class diagrams, the class relationships namely inheritance, aggregation and association exist in different combinations. These concurrency control schemes define lock modes for each relationship separately. They have not defined lock modes for objects which have combination of relationships. Hence they are not suitable.

In the second group of concurrency control schemes, compatibility is defined based on commutativity. In Agrawal and Abbadi [13], the idea of right backward (RB) commutativity is introduced. It states that "an operation o_1 is said to have RB commutativity with another operation o_2 on an object if for every state in which executing o_2 followed by o_1 has the same state and result as executing o_1 followed by o_2". This is less restrictive than commutativity relationship, as it is included in commutativity. However application programmers need to know all possible results of each method. In Badrinath and Ramamritham [14], attribute is the smallest granularity supported. They state that any two methods can be parallely executed if they do not share any attribute. But it requires knowledge of the structure of all methods in a class. In Badrinath and Ramamritham [15], the idea of recoverability is defined. i.e., the methods can be executed in any order. But the commit order is fixed. This also requires apriori knowledge of class structure.

In Malta and Martinez [8], the commutativity is automated by defining Direct Access Vector (DAV) for each method. A DAV is a vector whose field corresponds to each attribute defined in the class on which the method operates. Each value composing this vector denotes the most restricted access mode used by the method while accessing the corresponding attribute. Access mode of an attribute can have one of these values: null (N), read (R) and write (W) with $N < R < W$ for their restrictiveness. DAV is constructed at compile time by extracting syntactic information from the source code. Then the commutativity table is constructed for each class based on the rule that two methods in a class commute, if their DAV commutes. In Jun [9], concurrency is further improved by providing granularity lower than attribute level up to break points. But this also requires prior knowledge of structure of objects. In all these schemes, whenever the signature or implementation of a method changes due to design time transaction, its DAV as well as commutativity matrix has to be updated. So this scheme is not suitable for continuously evolving systems.

The conflicts among design time transactions are handled as follows. In Lee and Liou [7], all the above operations are done using only one lock mode by locking the entire schema with Read Schema (RS) and Write Schema (WS). In Malta and Martinez [8], lock mode for changing the class definition (class contents) is provided by RD (Read Definition) and MD (Modify Definition) lock modes. They have omitted the other types of schema changes. Agrawal and Abbadi [5] provided finer granularity by defining separate lock modes for attributes and methods (class contents). They also have not defined any lock mode for operations involving changes to nodes and edges. In Jun [9], separate lock modes are defined for attributes, methods and class relationships. In this paper, Attribute Access Vector (AAV) and Method Access Vector (MAV) define the lock status of all attributes and methods. The use of these access vectors provides fine granularity and thus brings maximum concurrency. But, operations involving class relationships, like changing class position and relationship are still serialized.

In Geetha and Sreenath [16], the granularity is further improved by defining lock modes for signature and implementation of method separately. They also ensure semantic consistency between classes where attributes and methods are defined and where they are used. This was overlooked in Jun [9]. Concurrency is further enhanced by defining Relationship Access Vector (RAV) which is used to lock only the related classes instead of locking the entire class diagram for operations involving node changes. This improved the performance further. In Geetha and Sreenath [18], separate lock modes have been defined to handle changes to nodes and edges.

All the above schemes perform better for stable domains, but involve maintenance overhead of access vectors for continuously evolving domains. They also require prior knowledge of the class diagram.

3 Proposed Scheme

3.1 Locking Scheme

The proposed scheme is based on the semantics mentioned in section 2.1. The class diagram is represented as Bidirected Acyclic Graph (BAG). In Directed Acyclic Graph (DAG), the edges flow from independent classes or parent classes (base classes, component classes and associated classes) to dependent classes or child classes (sub classes, composite classes and associative classes). It is unidirectional. The children can be reached from parent, but the reverse is not possible. In BAG, the edges link both ways. This is needed for the reduction of search time and for lock rippling.

Fig 2 shows a sample class diagram (schema) represented as BAG. In the proposed scheme, when a design time request is made for a base class, the requested lock mode is set on the class. Then the lock is rippled to all its children including the edges. When a change is done on the definition of an attribute/ method/ instance/ relationship or the class itself, all the classes related to this class (called sub class lattice), should also be locked in the same lock mode to maintain the consistency as mentioned in section 2.3. Fig 3 shows how the lock is rippled when a modify request is made to class A. In the case of design time requests, the locks are rippled downwards from parent to children.

Similarly when a runtime transaction requests to modify the attributes of an object in a subclass, its associated objects in parent classes are also locked by lock rippling. In the case of runtime requests, the locks are rippled upwards from child to all its parents.

Fig 4 shows how the runtime lock is rippled to parent class using the upward links. Let a runtime transaction request for class I. Note that A, E and C are the parents of class I. It can also be noted that the edges are also locked to block any request to change the relationship between these classes. This eliminates the problem of setting intension locks for multiple inheritance in ORION scheme [3]. Intension locks are always set in ORION scheme from root to leaf. ORION scheme can lock the classes A, E and I along the path. However, it will not lock C, which has to be locked to preserve consistency.

It is also worth noting that if runtime transactions request for base classes, component classes and associative classes, the locking will be only on the object of that class, as it will not have any upward edges. For example, in the sample class diagram, classes A, B and C are parent classes that do not have any parents. Hence, these classes alone are locked. Similarly, if child classes are requested for design time transactions they alone are locked, as they will not have any downward edges. For example consider the classes L, I, J and K in the sample class diagram.

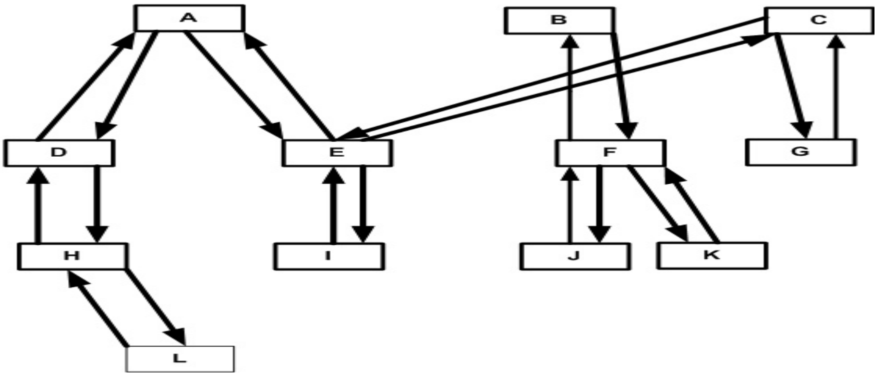


Fig. 2. Sample class diagram represented as Bidirected Acyclic Graph

Thus, the proposed scheme allows locking from root to leaf for design time requests and leaf to root locking for runtime transactions. The procedure for locking can be summarized as follows:

- Check for lock compatibility, when a lock is requested on a particular resource on the specified granularity.
- If the lock modes are compatible, Set the lock on the resource at the requested lock mode.
- Ripple the locks from root to leaf, if it is a design time transaction.
- Ripple the locks from leaf to root, if it is a run time transaction.
- Release the locks in the reverse order after release request.

3.2 Lock Semantics

Table 1 shows the proposed compatibility matrix for runtime and design time transactions. The lock modes have been defined based on the inferences in section 2.3. The semantics of the lock modes are as given below.

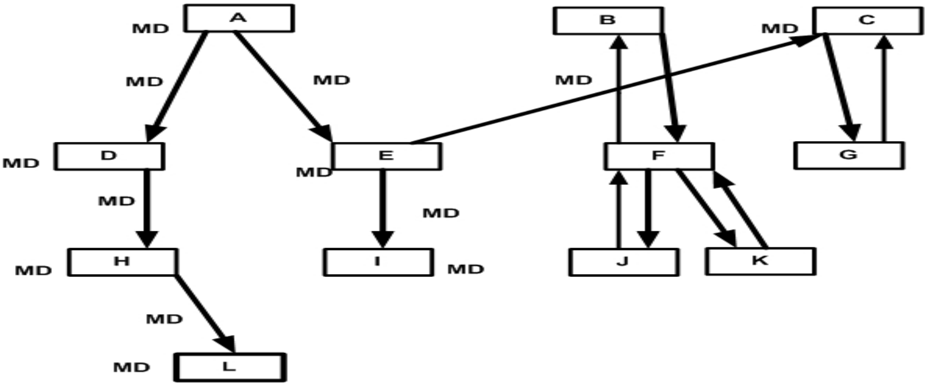


Fig. 3. Lock rippling to children on a modify definition request to class A

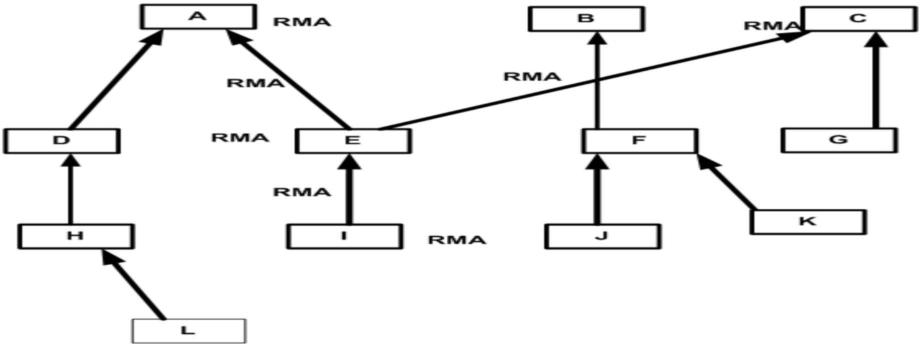


Fig. 4. Lock rippling to lock associated parent classes on runtime request to lock class I

Table 1. Compatibility matrix for runtime and design time transactions

	RRA	RMA	MD	RD	AE	DE	MCL
RRA	Y	N	N	Y	Y	N	N
RMA	N	N	N	Y	Y	N	N
MD	N	N	N	N	Y	N	N
RD	Y	Y	N	Y	Y	N	N
AE	Y	Y	Y	Y	Y	N	N
DE	N	N	N	N	N	N	N
MCL	N	N	N	N	N	N	N

1. **RRA** – Runtime Read Access- Read values of attributes as a runtime transaction.
2. **RMA**- Runtime Modify Access - Modify values of attributes as a runtime transaction.
3. **RD-** Read Definition – Read the domain and name of attribute/ instance/ class/ relationship (edge or link) and signature of a method. Signature of a method includes name of the method, input arguments and output arguments.

4. **MD-** Modify Definition – Read the domain and name of attribute/ instance/ class/ relationship. Read the signature and implementation of a method in a class.
5. **AE –** Add Entity – Add a new attribute/ method/ instance/ class/ edge (relationship).In the case of adding edges, a new relationship is defined between two existing classes.
6. **DE –** Delete Entity – Delete an existing attribute/ method/ instance/ class/ edge (relationship).
7. **MCL** – Modify Class Lattice – This lock mode is used to move the entities namely attribute/ method/ instance/ class/ edge (relationship) from one position to another position in the lattice.

In section 2.3, it is discussed that request for moving an entity alone requires locking the entire class lattice. The other operations can be parallelized in different sub class lattices. Hence, concurrency is more and at the same time, consistency is ensured wherever necessary.

4 Performance Analysis

Figure 5 shows the performance of the proposed scheme for continuously evolving OODBMS. It is compared with ORION scheme and Geetha and Sreenath scheme (G & N scheme) [18]. G & N scheme is applied without AAV, MAV and RAV. For runtime requests, the DAV is used. The objective of choosing these schemes is – ORION scheme is a popular concurrency control scheme with lock modes based on relationships and G &N scheme is the latest scheme with lock modes based on

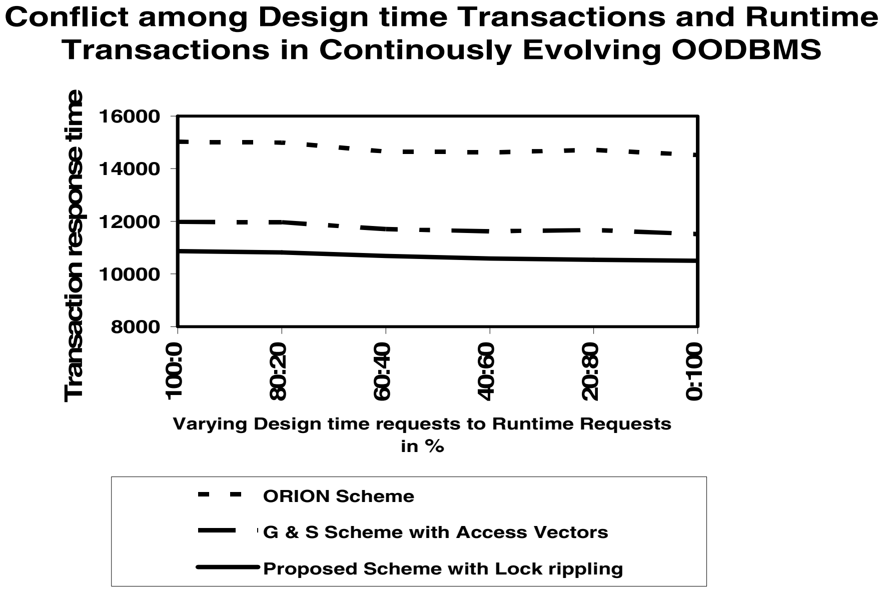


Fig. 5. Performance varying Design time request to Runtime request ratio

operations. ORION takes more time because of coarse granularity of objects for runtime requests. It takes coarse granularity of locking the entire class lattice for all design time transactions. The G&N scheme performs better than ORION scheme, because it provides fine granularity of attributes for runtime transactions. It provides separate semantic lock modes to change the contents of node, changes to edges and nodes. But it has the overhead of updating access vectors for attributes and methods, every time the schema is changed. The Direct Access Vector (DAV) of the methods in a class that is accessed every time, when the runtime transaction is executed. It also has to be updated when the schema is changed.

Table 2 gives the simulation parameters. 007 Benchmark by Carey [16] is well known for testing performance of OODBMS. But 007 benchmark defines the benchmark only for runtime requests. It does not define any testing cases for design time requests. So it cannot be fully adopted for the proposed scheme. However in the proposed scheme, some of the aspects of 007 bench mark are adopted for performance evaluation. The database model and testing cases of runtime requests are adopted. 007 benchmark classifies databases into small, medium and large, based on their size. Here, small size is chosen for simplicity. The design time requests are framed to cover all three types of schema changes.

Table 2. Simulation parameters

Parameters	Default value(range)
Time to process one operation	0.00000625ms
Mean time to set lock by instant access	0.3301 ms
Mean time to set lock by class definition access transaction	0.3422ms
Mean time to release lock	0. 0015ms
Multiprogramming level	8 (5-15)
Prob. Of Traversal	0.25 (0-1)
Prob. Of Query	0.25 (0-1)
Prob. Of Schema change	0.5 (0-1)
Prob. Of Changes to nodes	0.15 (0-1)
Prob. Of Changes to edges	0.20(0-1)
Prob. Of changes to node contents	0.15(0-1)
Transaction inter-arrival time	500 (100-1000)
Database model [16]	Small (small, medium, large)

5 Conclusion

In this paper, a multi- granular lock based concurrency control scheme with lock rippling is proposed. This paper provides maximum concurrency without affecting consistency of the database. The algorithm requires nil maintenance overhead and does not require apriori knowledge about the schema. It performs better than existing schemes for continuously evolving OODBMS. It covers all types of design time requests. It can be extended for fine granularity as future work.

References

1. Riehle, D., Berczuk, S.P.: Properties of Member Functions in C++, Report (2000)
2. Gray, J.N., Lorie, R.A., Putzolu, G.R., Traiger, L.I.: Granularity of locks and degrees of consistency in shared database. In: Nijssen, G.M. (ed.) Modeling in Database Management System, pp. 393–491. Elsevier, North Holland (1978)
3. Garza, J.F., Kim, W.: Transaction management in an object oriented database system. In: Proc. ACM SIGMOD Int'l Conference, Management Data (1987)
4. Kim, W., Bertino, E., Garza, J.F.: Composite Objects revisited. Object oriented Programming, Systems, Languages and Applications, 327–340 (1990)
5. Geetha, V., Sreenath, N.: Impact of Object Operations and Relationships on Concurrency Control in DOOS. In: Kant, K., Pemmaraju, S.V., Sivalingam, K.M., Wu, J. (eds.) ICDCN 2010. LNCS, vol. 5935, pp. 258–264. Springer, Heidelberg (2010)
6. Jun, W., Gruenwald, L.: An Effective Class Hierarchy Concurrency Control Technique in Object – Oriented Database Systems. Journal of Information and Software Technology, 45–53 (1998)
7. Lee, S.Y., Liou, R.L.: A Multi-Granularity Locking model for concurrency control in Object – Oriented Database Systems. IEEE Transactions on Knowledge and Data Engineering 8(1) (1996)
8. Malta, C., Martinez, J.: Automating Fine Concurrency Control in Object Oriented Databases. In: 9th IEEE Conference on Data Engineering, Austria, pp. 253–260 (1993)
9. Jun, W.: A multi-granularity locking-based concurrency control in object oriented database system. Journal of Systems and Software, 201–217 (2000)
10. Saha, D., Morrissey, J.: A self – Adjusting Multi-Granularity Locking Protocol for Object – Oriented Databases. IEEE (2009)
11. Banerjee, J., Kim, W., Kim, H.J., Korth, H.F.: Semantics and Implementation of Schema evolution in Object–Oriented Databases. In: Proc. ACM SIGMOD Conference (1987)
12. Eswaran, K., Gray, J., Lorrie, R., Traiger, I.: The notion of consistency and predicate locks in a database system. ACM Communications 19(11), 624–633 (1976)
13. Agrawal, D., Abbadi, A.: A Non-Restrictive Concurrency Control for Object- Oriented Databases. In: Pirotte, A., Delobel, C., Gottlob, G. (eds.) EDBT 1992. LNCS, vol. 580, pp. 469–482. Springer, Heidelberg (1992)
14. Badrinath, B., Ramamirtham, K.: Synchronizing transactions on objects. IEEE Transactions on Computers 37(5), 541–547 (1988)
15. Badrinath, B., Ramamritham, K.: Semantic- based concurrency control: beyond commutativity. ACM Transactions of Database Systems 17(1), 163–199 (1992)
16. Geetha, V., Sreenath, N.: A Multi–Granular Lock Model for Distributed Object Oriented Databases Using Semantics. In: Natarajan, R., Ojo, A. (eds.) ICDCIT 2011. LNCS, vol. 6536, pp. 138–149. Springer, Heidelberg (2011)
17. Kim, W.: Introduction to Object Oriented Databases. MIT Press, Cambridge
18. Geetha, V., Sreenath, N.: Semantic Based Concurrency Control in OODBMS. In: International Conference on Recent Trends in Information Technology, Chennai, India, June 3-5. IEEE Computer Society (2011)
19. Carey, M., Dewitt, D., Naughton, J.: 007 Benchmark. In: Proceedings of ACM SIGMOD Conference in Management of Data, Washington, USA (1993)

A New Scheme for IPv6 BD-TTCS Translator

J. Hanumanthappa[1] and D.H. Manjaiah[2]

[1] Dept. of Studies in Computer Science, University of Mysore, Manasagangotri, Mysore, India
[2] Dept. of Computer Science, Mangalore University, Mangalagangothri, Mangalore, India
hanums_j@yahoo.com, ylm21@yahoo.co.in

Abstract. In this work we investigate the effect of simple and efficient implementation of transition of IPv4 to IPv6 for BD-TTCS translator with list ranking algorithm using parallel computing task graph model (Diminution Tree) concept. In this work an unprecedented BD-TTCS IPv4/IPv6 translator is implemented with parallel computing based diminution tree. To corroborate the efficacy of the proposed method an experiment was conducted for various important performances issues namely Throughput, Round trip time, End-to-End delay, CPU utilization and simulated on NS2 simulator etc. In order to plot bar graph and Line Graph we used Matlab 7.11.0(R2010b).

Keywords: BD-TTCS, IPv4, IPv6, List ranking, Transition, Task graph model.

1 Introduction

Due to shortage of IPv4 public addresses the IETF has developed a unprecedented version of the Internet Protocol called IPv6. So many Institutions throughout the world had already started the transition to IPv6. We can find a couple of works carried out in this direction. Srisuresh and Egevang in the year 2001 describe a popular solution to the shortage of IPv4 addresses is Network Address Translation (NAT) which consists of hiding networks with private IPv4 addresses behind a NAT-enabled router with few public IPv4 addresses.

In their work Zeadally and Raicu in the year 2003 proposed the IPv6/IPv4 performance on Windows 2000 and Solaris 8. In their work they connected two identical personal computer's using a point-to-point connection. In order to calculate various performance issues namely throughput, round-trip time, CPU utilization, socket-creation time, and client–server interactions, for both TCP and UDP. They used packets ranging from 64 to 1408 bytes. Their experimental results show that IPv6 for Solaris 8 outperform IPv6 for Windows 2000,while IPv4 outperform IPv6 for TCP and UDP for both operating systems.

In their work Zeadally and Raicu(2003) noted the IPv6/IPv4 performance on Windows 2000 (Microsoft IPv6 Technology Preview for Windows 2000) and Solaris 8. They connected two identical workstations using a point-to-point connection and reported results such as throughput, round-trip time, CPU utilization, socket-creation time, and client–server interactions, for both TCP and UDP. They used packets ranging from 64 to 1408 bytes. Their experimental results show that IPv6 for Solaris 8

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 106–116, 2012.

outperform IPv6 for Windows 2000,while IPv4 outperform IPv6 for TCP and UDP for both operating systems.

Zeadally et al.(2004) Designed and calculated IPv6/IPv4 performance on Windows 2000,Solaris 8, and RedHat 7.3. The authors experimentally measured throughput of TCP and UDP, latency, CPU utilization, and web-based performance characteristics. Mohamed et al.(2006) evaluated IPv6/IPv4 performance on Windows 2003, FreeBSD 4.9 and Red Hat 9. They measured throughput, round-trip time, socket-creation time, TCP-connection time, and number of connections per second in three different test-beds. The first test-bed consisted of a single computer and communication was limited to processes running in this computer using the loopback interface. In the second test-bed, two computers were connected through an Ethernet hub. The Ethernet hub was replaced by a router in the third test-bed. They used packets ranging from 1 byte up to the limits of an IP packet (which is typically around 65,535 bytes).

Another solution to the problem of the shortage of public IPv4 addresses that faces the Internet consists to migrate to the new version of the Internet protocol(Davies, 2002;Deering and Hinden,1998;Popoviciu et al.,2006), called IPv6,or the coexistence between both protocols(Blanchet,2006). IPv6 fixes a number of problems in IPv4, such as the limited number of available IPv4 addresses. IPv6 has a 128-bit address, while IPv4 has a 32-bit address.

Shiau et al.(2006) evaluated IPv6/IPv4 performance in two different scenarios. In the first scenario,they connected two identical computers using a point-to-point connection. In the second scenario, the two identical computers were each connected to a real large-scale network environment through a Cisco 3750GB switch, and a Cisco 7609 router. Fedora Core II was the operating system of the two computers. The authors reported results such as throughput, round-trip time, packet loss rate, for both TCP and UDP. None of these previous works compares the experimental results for TCP and UDP throughput to the maximum possible throughput.

The organization of the paper is as follows: In section 2 presents proposed BD-TTCS model with the support of the BD-TTCS Translator architecture along with a brief introduction to BD-TTCS Framework. The Proposed Task Graph Model along with List ranking methodology in Transition of IPv4/IPv6 in BD-TTCS Translator with the Division and Merging Operation in transmission of IPv6 packets in BD-TTCS Translator are discussed in Section 3. Implementation and Evaluation of BD-TTCS translator is discussed in Section 4. The section 5 shows Simulation results with simulated graphs in NS2. Finally the paper is concluded in section 6.

2 Proposed Methodology

BD-TTCS is a general frame work for making use of transition of IPv4/IPv6 responses to calculate the various performances issues such as end-to-end delay, throughput and round trip time, CPU utilization etc. In this section we describe the BD-TTCS Architecture. As mentioned earlier the algorithm is based on clearly identifying the IPv4 and IPv6 internet addresses. For completeness we summarize the basic elements of BD-TTCS are mentioned as follows.1.Native IPv4 host,2.Native IPv6 host,IPv4 DNS(DNSv4),IPv6 DNS(DNSv6),DHCPv4,DHCPv6,IPv4 to IPv6 empowered Gateway. In this paper we put forward the IPv4 /IPv6 Bi-Directional mapping BD-TTCS which works when hosts in the native IPv4 network start connections with hosts in the connate IPv6 network and vice versa. Our newly implemented system such as BD-TTCS mainly reliable on determining two public IPv4 and IPv6 internet addresses for each communicating host. After the completion of determining whether an address is of type IPv4 category or IPv6 category, then try to divide the 32 bits IPv4 address into 4 octets by using parallel computing technique and 128 bits IPv6 address into 8 word lengths (16 bits). The second point is to understand the received datagram to determine whether it is of IPv4 or IPv6. The third important stage is apprehensioning and determining the header in order to classify the packet such as IPv4 or IPv6. The next step is translating the header (i.e IPv4 header to IPv6 header and IPv6 header to IPv4 header.

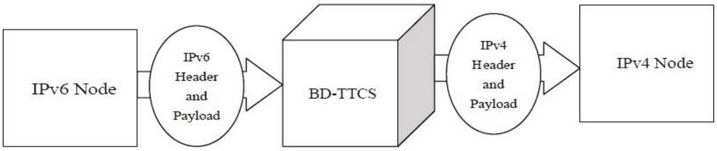


Fig. 1. Conversion of IPv6 header into IPv4 header

2.1 BD-TTCS Frame-Work

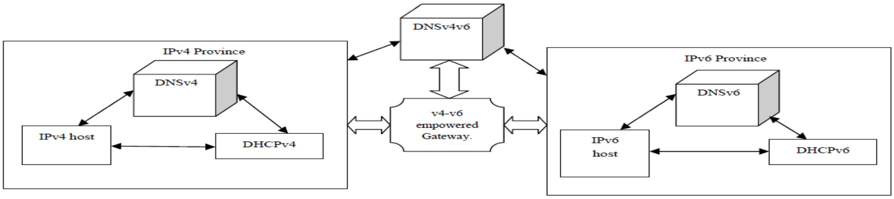


Fig. 2. Block diagram of the proposed work

2.2 Diminution Tree Model with List Ranking Based Transition of IPv4/IPv6 Algorithm in BD-TTCS

The Task dependency graph is mainly used to divide an 128 bits IPv6 address into 8 sections by using a Task Graph Model. In the task dependency graph the octets are also called as tasks or nodes. The Task graph model is mainly used to calculate the number of bit’s in an IPv6 address. In IPv6 segregation is a process of decomposing the entire 128 bits IPv6 address into 16 bits of 8 nodes. So we can convert the IPv6 BD-TTCS problem into 8-dimensional packet separation problem. The Static BD-TTCS translator diminution tree has many various tasks and also very few octet tables are preprocessed based on the total size of each IPv6 address in each task level. The diminution tree is constructed and operated depending upon the tradeoff between lookup performance and memory consummation.

2.3 List Ranking Methodology in BD-TTCS Translator

In this research work we have created parallel computing based list ranking technique using diminution tree model. The importance of list ranking to parallel computations is determined by Wyllie. List ranking and other calculations on linked list are really essential to the graphs and machines. The list ranking mechanism has been applied to calculate and count the number of octets in each level of the operation and it is also utilized to determine rank of each segregated node. In order to calculate rank of a node we will count and consider the number of nodes to its right. Array concept is an input given to the list ranking problem as a list given in the form of collection of nodes. A node mainly contains some data and a pointer to its right neighbor in a list. Initially we assume that each node contains only a pointer to its right neighbor. The rightmost node has pointer field as zero. While applying the parallel computing based list ranking with diminution tree technique to divide 8 octets of IPv6 address at level-1 has 8 very useful nodes which are scanned basically from right to left order. In Diminution tree based list ranking technique the right most node is Node-8 and whose rank is zero. Node-7 has rank 1 since the only node to its right is Node-8. Node-6 has rank 2 since the nodes Node-7 and Node-8's are to its right. Finally the Node-1 has a rank 7 because it is a first octet while scanning and counting the node's from left to right order. In Fig.3.the left to right scanning of all the 8 list of nodes has been specified in the order form like A-1,A-2,A-3,A-4,A-5,A-6,A-7,A-8. The same list ranking procedure can be also applied to calculate ranks of other various nodes at level-2, level-3 etc. Finally at recipient side the Diminution tree exhibits the rules and regulations of top-down approach to return node-1 consisting of 128 bits Destination IPv6 address.

Rank(1st octet(node) ->7--(2),Rank(2nd octet(node)->6--(3),Rank(3rd octet(node)->5--(4),Rank(4th octet(node)->4-(5),Rank(5th octet(node)->3-(6),Rank(6th octet(node)->2--(7),Rank(7th octet(node)->1--(8),Rank(8th octet(node)->0--(9)

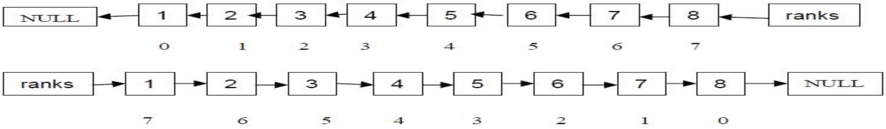


Fig. 3. Ordered list ranking method

2.4 The Division and Merging Process in Transmission of IPv6 Packets over BD-TTCS Translator

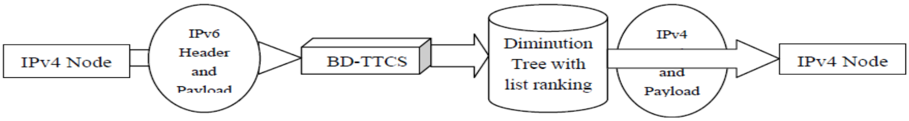


Fig. 4. IPv4/IPv6 using Diminution tree with list ranking in BD-TTCS

```
    struct node {unsigned char dm1;unsigned int start;
                 unsigned int stop;}node[10];
                 dm1=Total_Size_of_an_IPv6_Address;
                 e=dm1/16;mode=dm1%16;for(i=0;i<e;i++)
                                 {
node[i].start=node[1].stop=prefix[i];node[i].dml=16;}
node[s].start=prefix[s].prefix[s]>>mode<<mode;
 node[s].stop=node[s].stop[s].start   ||((1<<(mode+1))-1);
node[s].dm1=mode;for(i=e+1;e<8;i++)node[i].start=0;
 node[i].stop=0xFFFF;node[i].dm1=0; }
{                for (int j=1;j<=[logn];j++)
                 processor in parallel (for 1<=i<=n) does
 if(abutting[i]!=0;)rank[i]+=rank[abutting[i]];
 abutting[i]=abutting(abutting[i]); }
 where dm1=default mask length,s=aggregation,
 prefix[i]=Total_number_of_16_bits_in_a_one_word_size;
```

Fig. 5. Proposed BD-TTCS IPv4/IPv6 Transition pseudo code

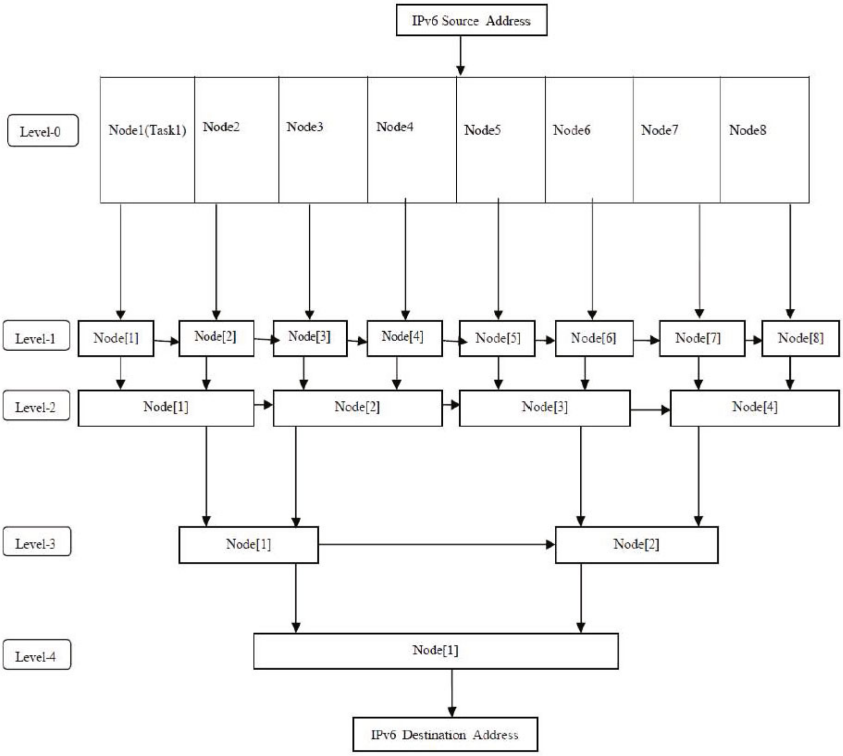


Fig. 6. Proposed Diminution tree using List ranking Methodology in BD-TTCS

3 Performance Evaluation Metrics

Our primary performance metrics in this research work is RTT(Round Trip Time),Throughput. In order to calculate IPv4 and IPv6 RTT Ping command is one of an essential tool. The network performance is different for both TCP and UDP Protocols. We present the findings of the research in this section. The majority of the tests were done for a sufficiently long period of time and resulted in the swapping of 50,000 packets to 1,00,00,00 packets depending on the various size packets ranging from 0 bytes to 65538 bytes of packets sent and the corresponding test. We conducted an empirical calculations based on the following performance metrics: Round trip time(RTT),End-to-End delay(EED),Throughput for UDP traffic and TCP traffic, using the packet size ranging from 32 bytes to 1024 bytes in order to calculate the impact of small size packet as well as the larger size packet on the translation and address mapping processes used in BD-TTCS translator. All the performance metrics are computed using the NS-2 simulation with maximum bursty traffic where each packet arrival follows a Poisson distribution process with rate χ=2. In order to plot bar graph and Line Graph we used Matlab 7.11.0(R2010b).

3.1 Round Trip Time(RTT)(Latency)

It is also said to be latency is the amount of time taken in a network communication when one packet likes to travel from one source host to another destination host and back to the originating host(Source host). The RTT is one of the important performance metric i.e measured in our simulation for parallel computing based diminution tree with list ranking based BD-TTCS IPv4/IPv6 transition scenario. The performance metric for RTT can be calculated in micro seconds. The mean RTT for a specific size packets in each communication can be calculated as follows [Eq.(1) and Eq.(2)]

$$Mean\ RTT = \frac{\sum_{i=1}^{n}}{N} \mathrm{RTT_i} \tag{1}$$

Where i is a packet number and N is the number of packets sent. It is worth noted that the packet size is directly proportional to round trip time (RTT).

$$\mathrm{RTT_i} = \mathrm{Tr_i} - \mathrm{Ts_i} \tag{2}$$

Whereas $\mathrm{RTT_i}$ is the Round trip time of packet “i”, $\mathrm{Ts_i}$ is the created time of a packet “i” at source host, $\mathrm{Tr_i}$ is the received time of packet “i” at the destination host at the End of its journey. N is the number of packets received at the source node and the mean RTT is the mean RTT cost for each communication session.

3.2 Packet Loss

Packet loss is one which is defined as the network traffic fails to reach its destination in a timely manner. Most commonly the packet gets dropped before the destination can be reached.

Packet dropped/loss $(P_d) = P_s - P_r$ (3)

Where P_s is the amount of packet sent at Source and P_r is the amount of packets received at Destination.

3.3 Jitter

Jitter is one which is defined as fluctuation of end to end delay from one packet to a next connection flow packet.

Jitter(J) = $| D_{i+1} - D_i |$ (4)

Where D_{i+1} are delay of $i^{th}+1$ packet and D_i is the delay of i^{th} communication packet.

3.4 End-to- End Delay

End to End delay refers to the time taken for a packet to be transmitted across a network from Source to Destination.

End-to-End $(D) = T_d - T_s$ (5)

Where T_d = Packet time received at destination node and T_s= Packet sent time at Source node.

4 Simulation Results

The Simulation results for various performance issues namely Throughput, Latency(RTT),End-to-End delay and CPU Utilization for TCP and UDP are shown in Figures Figs.7 to 10.

Table 1. Simulation parameters

Sl.No	Description	Values
1	IFQ Length(Buffer Size)	100 Packets
2	Number of Nodes	15
3	Grid Size	500 Meters *500 Meters
4	Very traffic Loads	5~300 Nodes
5	Payload Size	300 Bytes
6	Propogation delay	20 ms
7	Simulation Time	200 Seconds
8	Propogation Model	TwoRayGround
9	IFQ(Queue Management Scheme)	Drop tail

Fig.7,shows the part of our simulation result over the simulation time 0 to 60 seconds for the RTT of each created and sent packet when the size of the packet varies from 0 to 65536.Fig.8,also clearly presents that the mean RTT for BD-TTCS when a packet size varies from 0 to 65536.

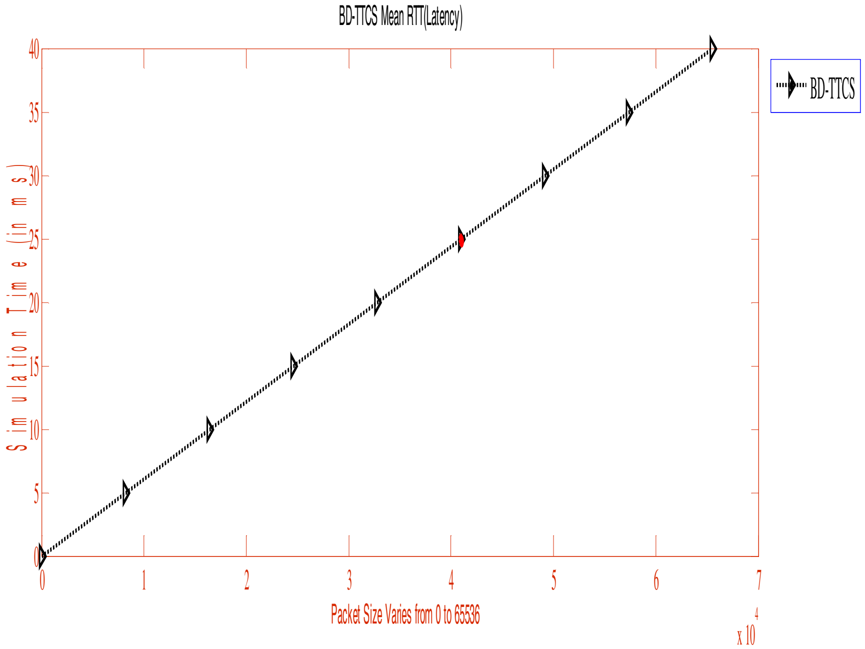


Fig. 7. Mean RTT When the Packet size varies from 0 to 65536

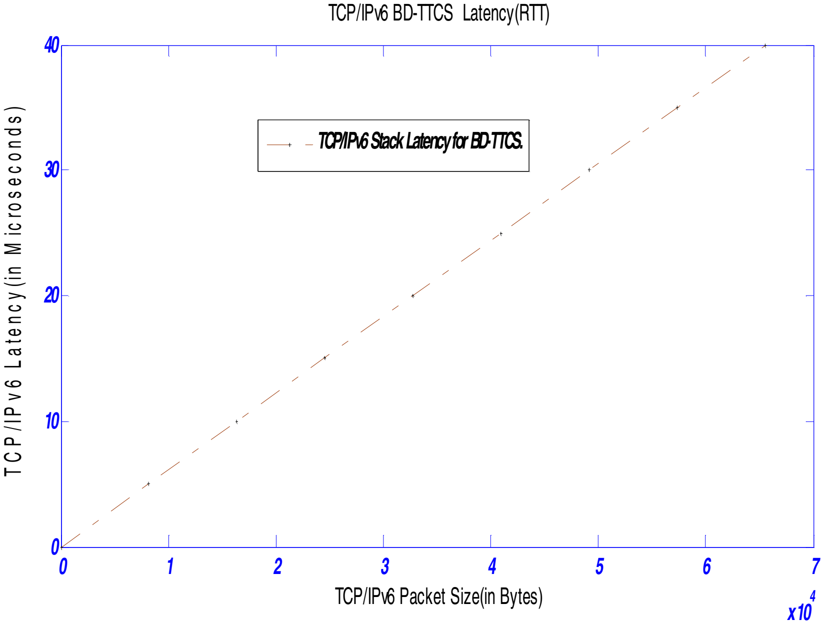


Fig. 8. TCP/IPv6 RTT for BD-TTCS when packet size ranges from 64 bytes to 64Kbytes

Throughput: The throughput is the rate at which a network sends or receives a data.

It is a good channel capacity of network connections and it is measured in terms of bits per second (bits/second).

Throughput (T) = P_r/P_f (6)

Where P_r is the amount of packets received and P_f is the amount of packets forwarded over certain time of interval.

The Fig.9, shows every layer of complexity adds additional overhead for all packet sizes. In Fig.10 we calculated TCP Round trip time(Latency) when a packet size varies from 64 Bytes to 64 Kbytes.

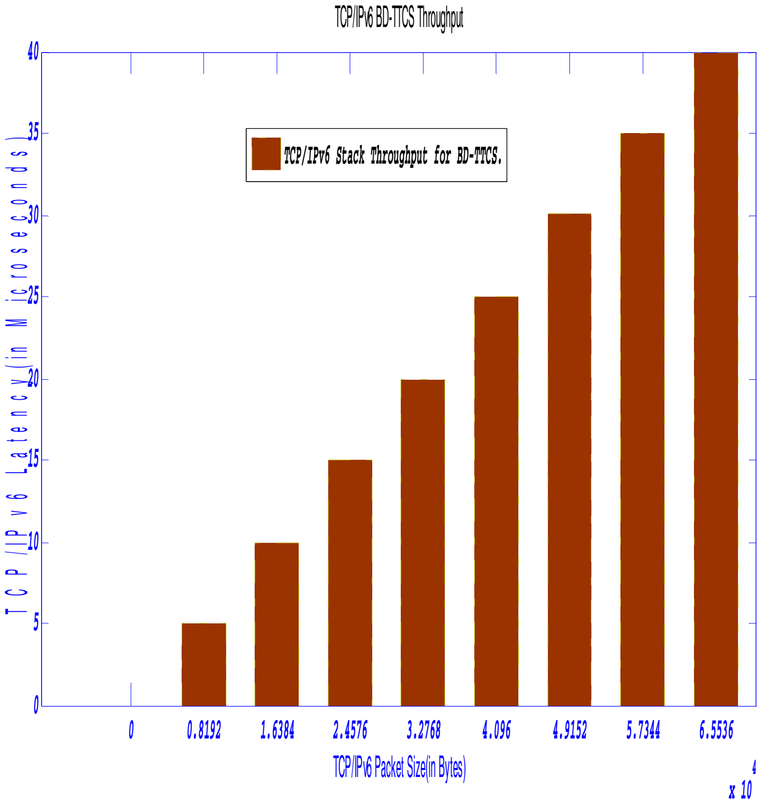


Fig. 9. TCP Throughput when a Packet size varies from 64 Bytes to 64 Kbytes

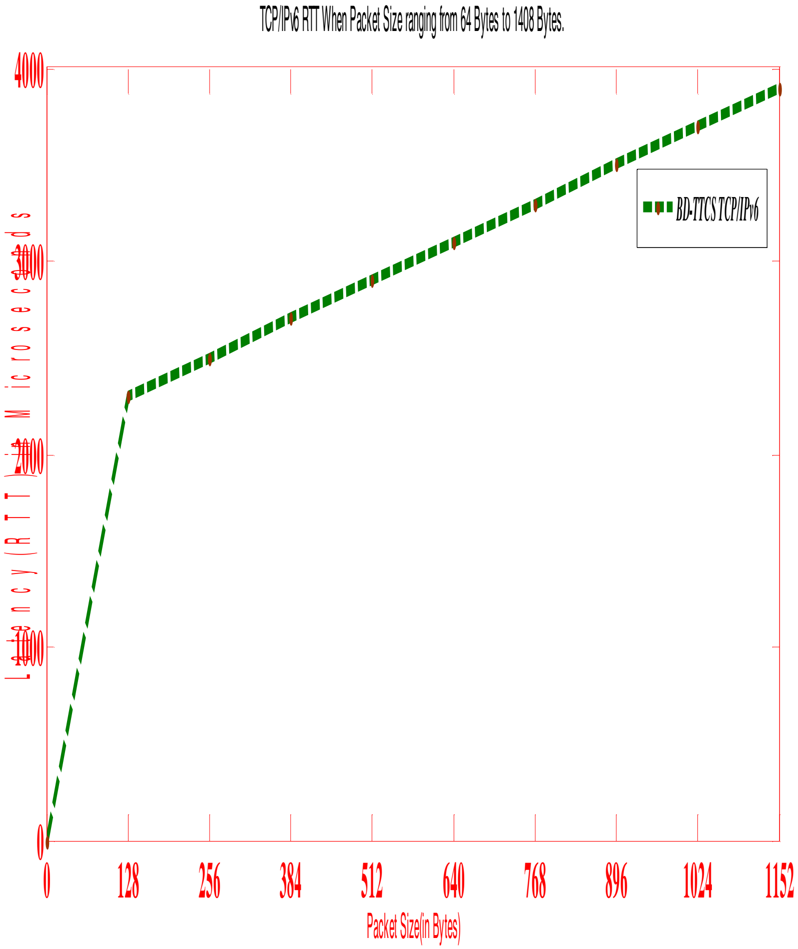


Fig. 10. TCP RTT when a Packet size varies from 64 Bytes to 1408 Kbytes

5 Conclusions

In this paper we have proposed an efficient BD-TTCS translator with the use of parallel computing based list ranking technique using Diminution tree model. Suitable IPv6 performance issues namely Throughput, Round trip time, End-to-end delay(EED),CPU utilization are explored for the purpose of transition of IPv4/IPv6. It is observe that using the proposed IPv6 performance issues one can achieve relatively a good BD-TTCS translator when compared any other IPv4 to IPv6 translators. We have designed our own BD-TTCS translator for IPv4 to IPv6 transition with the use of Parallel computing based diminution tree and we studied the various important performance issues on classification of translators. The experimental results have shown that BD-TTCS translator performs better than any other translators.

Acknowledgements. The authors would like to thank UGC for providing funding for this project with an entitled “Design Tool of IPv6 Mobility for 4G-Networks”, under eleventh plan of Major Research Project scheme (Ref.No F.No 36-167/2008(SR) dated .26.03.2009).

References

[1] Tshis, G., Srisuresh, P.: Network Address Translation-Protocol Translation (NAT-PT)

[2] Afifi, H., Toutain, L.: Methods for IPv4-IPv6 Transition, pp. 478–484. IEEE (July 1999)

[3] Raicu, I., Zeadally, S.: Evaluating IPv4 to IPv6 Transition Mechanisms. In: IEEE International Conference on Telecommunications, ICT 2003, Tahiti Papeete, French Polynesia (February 2003)

[4] Raicu, I., Zeadally, S.: Impact of IPv6 on End user Applications. To appear at 10th Intemational Conference on Telecommunications, ICT 2003, February 23, Tahiti Papeete, French Polynesia (2003)

[5] Raicu, I.: An Empirical Analysis of Internet Protocol version 6(IPv6), Master Thesis, Request for Comments 2766, Internet Engineering Task Force. Wayne State University (February 2002)

[6] Chen, J., Chang, Y., Lin, C.: Performance Investigation of IPv4/IPv6 Transition Mechanisms (2004)

[7] Govil, J., Govil, J., Kaur, N., Kaur, H.: IEEE Region 3 Huntsville Section(IEEE Southeast Con 2008): An examination of IPv4 and IPv6 Networks: Constraints, and Various Transition Mechanisms, April 3-6, Huntsville, Alabama, USA (2008)

[8] Govil, J., Govil, J.: EIT 2007: On the Investigation of Transactional and Interoperability Issues between IPv4 and IPv6 (2007)

[9] Jun, B., Wu, J., Leng, X.: IPv4/IPv6 Transition Technologies and Univer6 Architecture. International Journal of Computer Science and Network Security 7(1) (January 2007)

[10] Deering, S., Hinden, R.: Internet Protocol,Version 6(IPv6), Specification RFC 1883 (December 1995)

Nash Equilibrium in Weighted Concurrent Timed Games with Reachability Objectives

Shankara Narayanan Krishna, G. Lakshmi Manasa, and Ashish Chiplunkar

Department of Computer Science and Engineering,
IIT Bombay, Powai, Mumbai-76, India
{krishnas,manasa,ashishc}@cse.iitb.ac.in

Abstract. Timed automata [1] are a well accepted formalism in modelling real time systems. In this paper, we study concurrent games with two players on timed automata with costs attached to the locations and edges and try to answer the question of the existence of Nash Equilibrium (NE). Considering memoryless strategies, we show that with one clock it is decidable whether there exists a NE where player 1 has a cost bounded by a constant B, while with 3 clocks, it is not. The case of 2 clocks is an interesting open question.

1 Introduction

The concept of games on automata has been introduced with the central idea of multiple players making the automaton run in order to fulfil their interests. These games are classified into competetive and non-competitive games. In competetive games, one player wins the game while the others lose the game. In non-competetive games, there is no notion of winning or losing; each player plays the game in a way so that she gets a favourable outcome. Recently, a variety of *untimed* graph games have been studied in the literature [10], [13], [14] and [15] with discussions on existence/decidability of Nash equilibria. In the timed sense, [5], [6] has discussed techniques to compute Nash equilibria for *non-competetive* games played on *unweighted timed automata* with reachability objectives. Coming to *competetive* games in the timed sense, we mention [11],[9] and [8]. [11] and [9] consider games on *unweighted timed automata* with parity objectives. An algorithm for determining the set of states from which player 1 has a winning strategy is given in [11]; [9] improves the complexity of this algorithm by providing a reduction from timed parity games to untimed parity games. [8] discusses competetive games on *weighted timed automata* with cost bounded reachability objectives and 2 players. [8] shows that with six clocks, it is undecidable whether player 1 has a winning strategy with a cost bounded by a given constant c, while with only one clock, and costs $\in \{0, d\}$ for a fixed $d \in \mathbb{N}$, it is decidable. [4] gives a proof of undecidability of modelchecking logic WCTL against weighted timed automata using 3 clocks and one cost; from this proof, a sketch is shown how this possibly can be used to improve Brihaye's result of [8]. The proof sketch in [4] claims that it is undecidable whether player 1 has a winning strategy with a

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 117–128, 2012.

cost bounded by a given constant c in a competitive weighted timed game with 3 clocks.

The games we consider in this paper are *non-competetive*, with 2 players on *weighted timed automata*. Costs are attached to the locations as well as edges. The number of costs equal the number of players in the game. Costs are used to obtain payoffs for the players at the end of the game. We are not aware of any literature dealing with the existence of Nash Equilibrium (NE) for non-competitive games on weighted timed automata; in this paper, we initiate this study. We show that 3 clocks are sufficient to obtain undecidability for the existence of NE while with one clock, the existence of NE is decidable. Our results are extendible to multiplayer games.

2 Preliminaries

For any set S, S^* denotes the set of all strings over S. We consider as time domain $\mathbb{T}$ the set $\mathbb{Q}_+$ or $\mathbb{R}_+$ of non-negative rationals or reals, and Σ a set of actions. A time sequence over $\mathbb{T}$ is a non-decreasing sequence $\tau = (t_i)_{i\geq 1}$; for simplicity t_0 is taken to be zero always. Let X be a finite set of clocks. A clock valuation over X is a mapping $\nu : X \to \mathbb{R}_+$. We denote by $\mathbb{R}_+^X$ (or $\mathbb{T}^X$) the set of clock valuations over X. If $\nu \in \mathbb{T}^X$ and $\tau \in \mathbb{T}$, then $\nu + \tau$ is the clock valuation defined by $(\nu + \tau)(x) = \nu(x) + \tau$, for $x \in X$. A constraint or guard over X is a conjunction of expressions of the form $x \sim c$ where $x \in X$, $c \in \mathbb{N}$ and $\sim \in \{<, \leq, >, \geq, =\}$. We denote by $\mathcal{C}(X)$ the set of constraints over X. The satisfaction relation for constraints over clock valuations is denoted as $\nu \models \varphi$ whenever valuation ν satisfies constraint φ. $\nu \models \varphi$ if the constraint φ is satisfied when the clocks in X take on values specified by ν. Clock constraints allow us to test the values of clocks. To change the value of a clock x we use clock resets. $U_0(X)$ denotes the set of clock resets. A clock reset $\phi \in U_0(X)$ is defined by simply specifying the subset of clocks that get reset. Let ν be a valuation and ϕ be a clock reset. We use the notation $\nu' = \nu[\phi := 0]$ to denote $\nu'(z) = \nu(z)$ for all $z \in X \backslash \phi$ and $\nu'(y) = 0$ for all $y \in \phi$.

2.1 Timed Automata (TA)

A *timed automaton* [1] is a tuple $\mathcal{A} = (L, L_0, \Sigma, X, E, \eta, F)$ where L is a finite set of locations; $L_0 \subseteq L$ is a set of initial locations; Σ is a set of symbols; X is a finite set of clocks; $E \subseteq L \times L \times \Sigma \times \mathcal{C}(X) \times U_0(X)$ is the set of transitions; $\eta : L \to \mathcal{C}(X)$ defines the invariants of each location and $F \subseteq L$ is a set of final locations. $\mathcal{C}(X)$ and $U_0(X)$ are the set of clock constraints and clock resets as described above. An edge $e = (l, l', a, \varphi, \phi)$ represents a transition from l to l' on symbol a, with the valuation $\nu \in \mathbb{T}^X$ satisfying the constraint φ, and then ϕ gives the resets of certain clocks.

The semantics of a timed automaton $\mathcal{A}$ is given by a labeled timed transition system $\mathcal{T}_\mathcal{A} = (S, \to)$ where $S = L \times \mathbb{T}^X$. We refer to an element $l \in L$ of $\mathcal{A}$ as a *location* while we refer to an element $(l, \nu) \in S$ of $\mathcal{T}_\mathcal{A}$ as a *state*. The terms transition and edge are used interchangeably. $\rightarrow$ is composed of two kinds of transitions (i)delay transitions: In a state (l, ν), on elapsing time t, we reach the state $(l, \nu + t)$ and (ii) discrete transitions $(l, \nu) \xrightarrow{(\sigma_1, \varphi_1, \phi_1)} (l', \nu')$. Here, ν satisfies the constraint φ as well as the invariant $\eta(l)$, ν' is obtained from ν by resetting the clocks specified in ϕ and ν' satisfies $\eta(l')$, the invariant of location l'. A path is a finite sequence of consecutive transitions. The path is said to be accepting if it starts in an initial location ($l_0 \in L_0$) and ends in a final location. A run through a path from a valuation ν'_0 (with $\nu'_0(x) = 0$ for all x) is a sequence $(l_0, \nu'_0) \xrightarrow{t_1} (l_0, \nu_1) \xrightarrow{(\sigma_1, \varphi_1, \phi_1)} (l_1, \nu'_1) \xrightarrow{t_2} (l_1, \nu_2) \xrightarrow{(\sigma_2, \varphi_2, \phi_2)} (l_2, \nu'_2) \cdots (l_n, \nu'_n)$. The timed word corresponding to the run is $\sigma = (\sigma_1, t_1)(\sigma_2, t_2)(\sigma_3, t_3) \cdots (\sigma_n, t_n)$. Note that $\nu_i = \nu'_{i-1} + (t_i - t_{i-1}), \nu_i \models \varphi_i, \eta(l_{i-1})$, and that $\nu'_i = \nu_i[\phi_i := 0], i \geq 1, \nu'_i \models \eta(l_i)$. A timed word σ is accepted by $\mathcal{A}$ iff there exists an accepting run (through an accepting path) over $\mathcal{A}$, the word corresponding to which is σ. The timed language $L(\mathcal{A})$ accepted by $\mathcal{A}$ is defined as the set of all timed words accepted by $\mathcal{A}$.

Clock Intervals. Let $c_m \in \mathbb{N}$ be the maximum constant occurring in the guards $\mathcal{C}(X)$ of the TA $\mathcal{A}$. For every clock $x \in X$, define a set of *clock intervals* $\mathcal{I}$, as

$$\mathcal{I} = \{[c] | 0 \leq c \leq c_m\} \cup \{(c, c+1) | 0 \leq c < c_m\} \cup \{(c_m, \infty)\}$$

A clock region is a tuple $((I_x)_{x \in X}, \prec)$ where $I_x \in \mathcal{I}$ and $\prec$ is a total preorder on the set of clocks with non-integral intervals. The set of regions $\mathcal{R}$ partitions the set of clock vaulations $\mathbb{T}^X$. For a timed automaton, [1] shows the existence of such a set of regions consistent with time elapse and uniform with respect to constraints and resets. A region automaton [1] can be constructed given $\mathcal{A}$ and $\mathcal{R}$ which accepts the untimed equivalent of $L(\mathcal{A})$.

Let $\sigma = (\sigma_1, t_1)(\sigma_2, t_2)(\sigma_3, t_3) \cdots (\sigma_n, t_n)$ be a timed word. Define $\text{Untime}(\sigma) = \sigma_1 \sigma_2 \cdots \sigma_n$. $\text{Untime}(L(\mathcal{A})) = \{\text{Untime}(\sigma) \mid \sigma \in L(\mathcal{A})\}$. The region automaton is an abstraction of the timed automaton accepting $\text{Untime}(L(\mathcal{A}))$ [1].

Theorem 1. *Let $\mathcal{A}$ be a timed automaton. Then the problem of checking emptiness of $L(\mathcal{A})$ is decidable [1].*

2.2 Weighted Timed Automata (WTA)

An extension of timed automata useful in applications like scheduling problems and controller synthesis is weighted timed automata (WTA) introduced in [2], [3]. In this, cost variables are attached to locations and edges. The costs are used only as observers, they cannot be compared or evaluated in the automata. The behaviour of the automata is not influenced by these costs. This has led to several decidability results for optimization problems like minimum cost reachability, cost optimal schedules and so on. We recall the definition of WTAs as in [7].

A *weighted timed automaton* is a tuple $\mathcal{A} = (L, L_0, X, Z, E, \eta, C)$ where L is a finite set of locations, $L_0 \subseteq L$ is a set of initial locations, X is a finite set of clocks, Z is a finite set of costs (let $|Z| = m$), $E \subseteq L \times \mathcal{C}(X) \times U_0(X) \times L$ is the set of transitions. A transition $e = (l, \varphi, \phi, l') \in E$ is a transition from l to l' with valuation $\nu \in \mathbb{T}^X$ satisfying the constraint φ, and ϕ gives the set of clocks to be reset. $\eta : L \to \mathcal{C}(X)$ defines the invariants of each location. $C : L \cup E \to \mathbb{N}^m$ is the cost function which gives the rate of growth of each cost. Note that the costs are called *stopwatches* if $C : L \cup E \to \{0, 1\}^m$. From the nature of the costs and stopwatches, it is clear that stopwatches are restricted costs. WTA with stopwatches form a subclass of WTA with costs.

The semantics of a WTA $\mathcal{A} = (L, L_0, X, Z, E, \eta, C)$ is given by a labelled timed transition system $\mathcal{T}_{\mathcal{A}} = (S, \to)$ where $S = L \times \mathbb{T}^X \times \mathbb{T}^Z$. We refer to an element $l \in L$ of a WTA $\mathcal{A}$ as a *location* while we refer to an element $(l, \nu, \mu) \in S$ of $\mathcal{T}_{\mathcal{A}}$ as a *state*. The terms transition and edge are used interchangeably. $\to$ is composed of transitions

- Time elapse t in l: A state (l, ν, μ) after time elapse t evolves to (l', ν', μ'), where $l' = l$, $\nu' = \nu + t$, $\mu' = \mu + C(l) * t$ and for all $0 \leq t' \leq t$, $\nu + t' \models \eta(l)$.
- Location switch: $(l, \nu, \mu) \xrightarrow{(\varphi,\phi)} (l', \nu', \mu')$ if there exists $e = (l, \varphi, \phi, l') \in E$, such that $\nu \models \varphi$, $\nu' = \nu[\phi := 0]$ and $\mu' = \mu + C(e)$. Here, $\nu \models \eta(l)$, $\nu' \models \eta(l')$.

A path is a sequence of consecutive transitions in the transition system $\mathcal{T}_{\mathcal{A}}$. A path ρ starting at (l_0, ν'_0, μ'_0) is denoted as $\rho = (l_0, \nu'_0, \mu'_0) \xrightarrow{t_1} (l_0, \nu_1, \mu_1) \xrightarrow{(\varphi_1,\phi_1)} (l_1, \nu'_1, \mu'_1) \xrightarrow{t_2} (l_1, \nu_2, \mu_2) \xrightarrow{(\varphi_2,\phi_2)} (l_2, \nu'_2, \mu'_2) \cdots (l_n, \nu'_n, \mu'_n)$. Note that $\nu_i = \nu'_{i-1} + (t_i - t_{i-1}), \nu_i \models \varphi_i$, $\nu'_i = \nu_i[\phi := 0]$ and $\mu_i = \mu'_{i-1} + C(l_{i-1}) * (t_i - t_{i-1})$, $\mu'_i = \mu_i + C(l_{i-1}, \varphi_i, \phi_i, l_i)$.

2.3 Deterministic Two Counter Machines

A deterministic 2-counter machine $\mathcal{M}$ with counters c_1 and c_2 is described by a program formed by five basic instructions:

- l_m : goto l_j;
- l_m : if $c_i = 0$ then goto l_j else goto l_k; (check for zero)
- l_m : $c_i := c_i + 1$, goto l_j;(increment counter c_i)
- l_m : $c_i := c_i - 1$, goto l_j; (decrement counter c_i. A decrement instruction for c_i is always preceded by a check for zero for c_i so that $\mathcal{M}$ does not get stuck)
- l_m : HALT;

Without loss of generality, assume that the instructions are labelled $l_1, \ldots, l_n$ where $l_n = HALT$ (a special instruction) and that to begin with, both counters have value zero. Let $L = \{l_i \mid 1 \leq i \leq n\}$ be the labels of the instructions in $\mathcal{M}$. A configuration of the two counter machine is a tuple $(l, n_1, n_2) \subseteq L \times \mathbb{N} \times \mathbb{N}$. A configuration tells us the current instruction of $\mathcal{M}$ as well as the values of c_1, c_2. The behavior of the machine is described by a possibly infinite sequence of configurations $(l_0, 0, 0), (l_1, C_1^1, C_2^1), \ldots (l_k, C_1^k, C_2^k) \ldots$ where C_1^k and C_2^k are the respective counter values and l_k is the label of the kth instruction. The halting problem of such a machine is known to be undecidable [12].

Let $V_{\mathcal{M}} = \{(c_1, c_2) \mid \exists l$ such that $\mathcal{M}$ visits $(l, c_1, c_2)\}$ be the set of all pairs of values of counters c_1, c_2 which result from $\mathcal{M}$. Two configurations (l, a, b) and

(l', a', b') are distinct if $l \neq l'$ or $a \neq a'$ or $b \neq b'$. Clearly, $V_{\mathcal{M}}$ is finite iff $\mathcal{M}$ visits finitely many distinct configurations.

3 Timed Games

In this section, we introduce the timed games we consider in this paper. A timed game structure is a tuple $\mathcal{G} = (L, L_0, X, Z, P_1, P_2, E, \eta, C, F)$ where L is a finite set of locations, $L_0 \subseteq L$ is a set of initial locations, F is a set of final or target locations and X is a finite set of clocks; there are two cost variables - $|Z| = 2$ corresponding to the two players P_1, P_2; $E \subseteq L \times L \times C(X) \times 2^X$ is the set of transitions; $\eta : L \to C(X)$ is a function assigning an invariant to each location and $C : L \cup E \to \mathbb{N}^2$ is a function associating costs to the two players on locations and edges. A timed game is defined on this structure as follows: The set of edges is partitioned as $E_1 \cup E_2$ corresponding to the two players. At each location, each player i proposes either (1) a time delay t_i and an edge $e_i \in E_i$, or (2) a time delay ∞. The semantics of a timed game structure follow from that of a weighted timed automaton in Section 2.2. If P_i chooses t_i, e_i from a state $q = (l, \nu, \mu)$, then there must exist states q', q'' such that $q = (l, \nu, \mu) \xrightarrow{t_i} q' = (l, \nu + t_i, \mu') \xrightarrow{e_i} q'' = (l'', \nu'', \mu'')$. The transitions $q \xrightarrow{t_1} q' \xrightarrow{e_1} q''$ are executed if $t_1 < t_2$. If $t_2 < t_1$, then player 2's choice (t_2, e_2) is selected. When $t_1 = t_2$, player 2's choice is selected.

A strategy for player i is a function $\lambda_i : S \to (R^+ \times E_i) \cup \{\infty\}$ such that if $\lambda_i(q) = (t_i, e_i)$, then it is possible to take a delay t_i at state q followed by the discrete transition e_i. (S is as defined in section 2.2). A strategy profile is a pair (λ_1, λ_2) of strategies such that λ_i is a strategy for player i. A run $\rho = (l_0, \nu_0', \mu_0') \xrightarrow{\tau_1} (l_0, \nu_1, \mu_1) \xrightarrow{f_1} (l_1, \nu_1', \mu_1') \dots \xrightarrow{\tau_{n+1}} (l_n, \nu_{n+1}, \mu_{n+1}) \xrightarrow{f_{n+1}} (l_{n+1}, \nu_{n+1}', \mu_{n+1}')$ is said to be played according to the strategy profile (λ_1, λ_2) where $\lambda_1(l_i, \nu_i', \mu_i') = (t_{1i}, e_{1i})$ and $\lambda_2(l_i, \nu_i', \mu_i') = (t_{2i}, e_{2i})$. $\tau_{i+1} - \tau_i$=min$\{t_{1i}, t_{2i}\}$, for $i \geq 0$ and $f_i = e_{2i}$ whenever $t_{2i} < t_{1i}$ and $f_i = e_{1i}$ if $t_{1i} < t_{2i}$. If $t_{1i} = t_{2i}$ then $f_i = e_{2i}$. The game is thus biased in favour of player 2. A strategy profile (λ_1, λ_2) gives rise to a unique run : based on the delays proposed, a unique move is chosen. Such a run ρ is called the outcome of (λ_1, λ_2) and is denoted $outcome(\lambda_1, \lambda_2)$. The strategies we consider are *memoryless*, since we only look at the current state to decide the next move. A *terminal history* is a run ρ starting from the initial location, ending in a target location and never passing through a target location in between. Given a terminal history ρ, the payoff of player i is $-u_i(\rho)$ where $u_i(\rho)$ is the cost accumulated along ρ. For a run ρ that does not end in a target location, the payoff for both players is ∞. The *objective* of each player is to reach a target location accumulating as small a cost as possible. Our games are therefore non-competetive, since neither player aims to increase the others' cost.

3.1 Nash Equilibrium (NE)

Consider the example game $\mathcal{G}_1$ in Figure 1. Assume that (λ_1, λ_2) is a profile where player 1 incurs delay $t < 1$ at l and player 2 incurs $1 - t$ at m. The payoff of (λ_1, λ_2) is $(t, 1)$. Let λ_1' be a strategy for player 1 by which the delay at l is $t' < t$. If we keep λ_2 fixed, then the payoff of (λ_1', λ_2) is (∞, ∞) since the target location cannot be reached. Likewise, if we keep λ_1 fixed and consider a strategy λ_2' where λ_2' incurs a shorter delay at m than λ_2, then (λ_1, λ_2') will never reach a target location. Thus, either player deviating from the a profile with terminal history yields them a payoff of (∞, ∞). To take care of this issue, we henceforth look only at strategy profiles that end in a target location. We call such profiles *terminal strategy profiles (tsp).*

Now consider game $\mathcal{G}_2$. Here, for every strategy profile we consider, there exists another profile (reaching the target location) which gives lower payoff to both players.

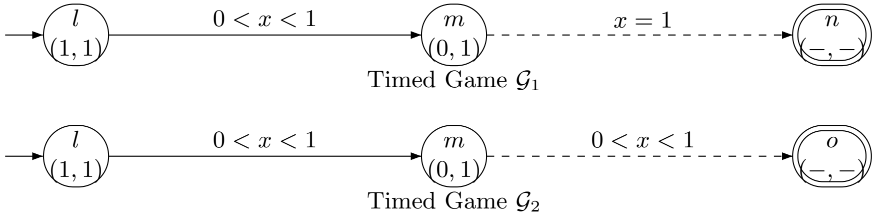


Fig. 1. $(-, -)$ indicates that cost of that location does not affect payoff of the players

Inspired by the standard notion of Nash Equilibrium in untimed games and taking into consideration the nature of time in games discussed above, we define Nash Equilibrium (NE) in timed games as follows:
A terminal strategy profile $(\lambda_1^*, \lambda_2^*)$ is a *Nash Equilibrium* iff

$$u_1(outcome(\lambda_1^*, \lambda_2^*)) \leq u_1(outcome(\lambda_1, \lambda_2^*)) \text{ for every strategy } \lambda_1 \text{ of } P_1 \text{ and}$$

$$u_2(outcome(\lambda_1^*, \lambda_2^*)) \leq u_2(outcome(\lambda_1^*, \lambda_2)) \text{ for every strategy } \lambda_2 \text{ of } P_2$$

such that both (λ_1, λ_2^*) and (λ_1^*, λ_2) are terminal strategy profiles. Game $\mathcal{G}_1$ will have all tsp's as NEs, while no tsp of game $\mathcal{G}_2$ is an NE.

4 Existence of Nash Equilibrium

In this section, we show that the existence of a NE such that P_1 has a payoff bounded by a constant B is undecidable. First, we show that the following question regarding two counter machines is undecidable, and then use it to prove our result.

Q1: Given a two counter machine $\mathcal{M}$, is it decidable whether, starting from a configuration $(q_0, 0, 0)$, $\mathcal{M}$ will visit finitely many distinct configurations?

Proposition 1. *Question Q1 is undecidable.*

Theorem 2. *Given a timed game structure $\mathcal{G}$ with three clocks and stopwatch costs, it is undecidable if there exists an NE (λ_1, λ_2) for the corresponding game, in the outcome of which P_1 incurs a cost bounded by a constant B; that is, $u_1(outcome(\lambda_1, \lambda_2)) < B$.*

Proof. We construct a timed game structure $\mathcal{G}$ with 3 clocks x_1, x_2, x_3 that simulates a two counter machine $\mathcal{M}$. The clocks x_1, x_2, x_3 encode the counter values c_1, c_2 as follows: At the end of each module in $\mathcal{G}$,

$$\nu(x_1) = \frac{1}{2^{c_1}3^{c_2}}, \nu(x_2) = 0.$$

The clock x_3 is used to do calculations and is used for rough work. We show that $\mathcal{G}$ has an NE (λ_1, λ_2) such that $u_1(outcome(\lambda_1, \lambda_2)) < 6$ iff $\mathcal{M}$ visits only finitely many distinct configurations. The detailed proof is long and can be seen in the technical report *www.cse.iitb.ac.in/∼krishnas/icdcit12.pdf.* We just illustrate how we can increment a counter.

In all locations of $\mathcal{G}$, we add a loop with constraint $x_i = 1, i \in \{1, 2, 3\}$ and reset it when x_i reaches 1. For convenience, we will not draw this in any of the locations in the various modules. This loop ensures that the values of x_1, x_2 are always related to each other in one of the following ways:

$$\nu(x_1) - \nu(x_2) = \frac{1}{2^{c_1}3^{c_2}}, 0 \leq x_2 \leq x_1 \leq 1, \text{ or}$$
$$\nu(x_2) - \nu(x_1) = 1 - \frac{1}{2^{c_1}3^{c_2}}, \ 0 \leq x_1 \leq x_2 \leq 1.$$

$\mathcal{G}$ is constructed by connecting the modules simulating the various increment, decrement and zero check instructions according to $\mathcal{M}$. For example, if $\mathcal{M}$ contains the instructions l_1: Increment c_1, goto l_2, l_2: if $c_1 > 0$, goto l_3, else HALT, and l_3: decrement c_1, goto l_1, then $\mathcal{G}$ is obtained by connecting the modules for incrementing c_1, checking if c_1 is zero, then decrementing c_1 in a round robin fashion. For ease of notation, we have used in the locations costs $\in \mathbb{N}$ instead of just 0 and 1. It must be noted that a location which uses a cost other than 0 and 1 can be replaced with a sequence of locations, each of which have costs over $\{0, 1\}^2$. To see this, look at the technical report provided by the link.

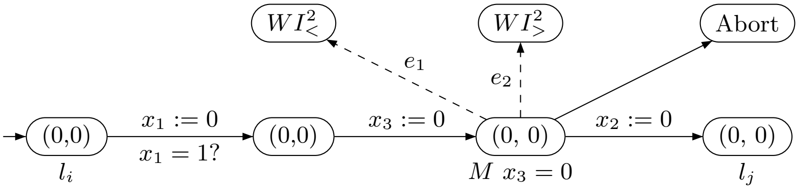


Fig. 2. Module for incrementing c_1

The widget $WI^2_>$ is obtained by switching the costs in all locations of $WI^2_<$.
□

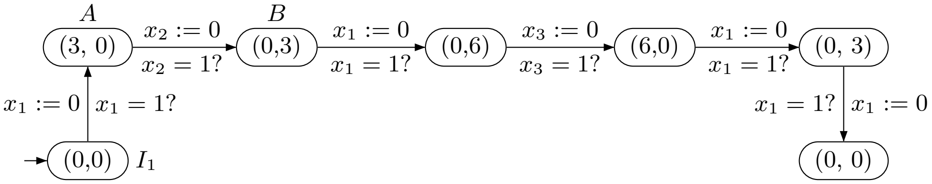


Fig. 3. Widget $WI^2_<$

5 Decidability of NE Existence with 1 Clock

In this section, we show that the existence of bounded NE is decidable in timed game structures having a single clock. From now on, when we mention $\mathcal{G}$, it must be understood that $\mathcal{G}$ has a single clock. Since there is only one clock, we assume that the constraints on the edges of $\mathcal{G}$ are clock intervals of the form $x \in [c]$ or $x \in (c, c+1)$ or $x \in (c, \infty)$. It is easy to see that the clock constraints can always be expressed in this form, for instance if we had an edge with the clock constraint $a < x < b$ can be replaced with edges $x \in (a, a+1), x = [a+1], x \in (a+1, a+2), \ldots, x \in (b-1, b)$. We shall prove that in the case of a single clock, it is decidable to check for the existence of an NE where player 1's payoff is bounded by a constant B.

5.1 Simple Profiles

Consider a timed game $\mathcal{G}$ where the constraints on the edges are regions of the form $(d, d+1), [h]$ or (c_m, ∞), where $d < c_m, h \leq c_m$ and c_m is the maximum constant used in the constraints. Let the edges be numbered e_i, $1 \leq i \leq p$. We simplify the notion of strategies (from that in section 3) and define it as a function $\overline{\lambda_i} : S \to E_i$ where S is the set of states of the form (l, ν, μ), and E_i is the set of player i's edges; each edge is annotated with a constraint $x \in (d, d+1)$ or $x = [h]$ or $x \in (c_m, \infty)$ and a possible reset. The implicit understanding when player i selects an edge e with constraint φ (and reset ϕ) from a location l with clock valuation ν to a location l' is that (i) If player i spends time t in l before selecting e, then $\nu + t \models \eta(l)$, (ii) $e \in E_i$, (iii)$\nu + t \models \varphi$ and $[\nu + t][\phi] \models \eta(l')$. These strategies are called *simple strategies*, and a *simple profile* is a pair $(\overline{\lambda_1}, \overline{\lambda_2})$ of simple strategies.

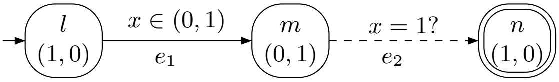


Fig. 4.

We represent simple strategies of player i as the set of edges chosen. In Figure 4, the simple strategies $\overline{\lambda_1} = \{e_1\}$ and $\overline{\lambda_2} = \{e_2\}$ gives rise to the simple profile $(\overline{\lambda_1}, \overline{\lambda_2})$. Note that there are several strategy profiles (as defined in Section 3) that correspond to $(\overline{\lambda_1}, \overline{\lambda_2})$. For example in Figure 4, $\lambda_{1t}(l) = (t, e_1)$ for $0 < t < 1$ are strategies of player 1 that correspond to $\overline{\lambda_1}$. We omit the clock valuation ν as well as the cost μ in the notation since player 1 can choose (t, e_1) for all possible (l, ν, μ). The rest of the paper, we will continue to use this notation. $\lambda_{2t'}(m) = (t', e_2)$ for $0 < t' < 1$ are strategies of player 2 that correspond to $\overline{\lambda_2}$. Thus, a simple profile represents a collection of strategy profiles (strategy profiles are defined in Section 3). If there are edges $e \in \overline{\lambda_1}$ and $e' \in \overline{\lambda_2}$ such that e and e' are edges from the same location l and having the same constraint, then split the simple profile $(\overline{\lambda_1}, \overline{\lambda_2})$ into two: one where e is chosen and the other where e' is chosen. This way, given a simple profile, there is a unique outcome associated to it, formed by following the appropriate sequence of edges.

The payoff of a player i given an outcome ρ of a simple profile $(\overline{\lambda_1}, \overline{\lambda_2})$ is calculated as follows: Let e be an edge with constraint $x \in (a, b)$, from a location l having cost c for player 1. Let player 1 incur no cost along edge e. Let t_e be the time spent at location l. If x was reset while entering l, then $c.t_e$ with $a < t_e < b$ is the payoff after taking edge e from location l. Now consider an edge e with constraint $x = 1?$ from a location l. See Figure 5. Let f be the last edge before e along ρ which had a reset. Let $e_1, e_2, \ldots, e_k$ be the edges chosen in between from locations $l_1, \ldots, l_k$. Let c_i be player 1 costs in locations l_i, $1 \leq i \leq k$, and let c'_i be player 1 edge costs along e_i, $1 \leq i \leq k$. Let c be the cost of player 1 in location l and let c' be her edge cost along e.

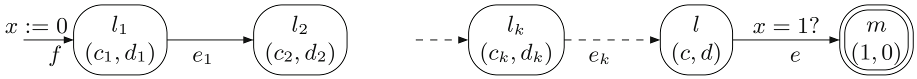


Fig. 5.

Let t_{e_i} be the variables representing the time spent at locations l_i associated to edges e_i, $1 \leq i \leq k$. Let t_e be the variable representing time spent at location l. The payoff between l_1 and m is $(c_1 t_{e_1} + \cdots + c_k t_{e_k} + c t_e) + (c'_1 + \cdots + c'_k + c')$ with the condition that $t_{e_1} + \cdots + t_{e_k} + t_e = 1$. If the edge e had a constraint $x \in (a, b)$, we would have written $a < t_{e_1} + \cdots + t_{e_k} + t_e < b$. Solving this, we obtain the payoff of player 1. Each assignment of values to t_{e_i}, t_e satisfying $t_{e_1} + \cdots + t_{e_k} + t_e = 1$ (or $a < t_{e_1} + \cdots + t_{e_k} + t_e < b$ as is the case) corresponds to the payoff of a profile $(\lambda_1, \lambda_2) \in (\overline{\lambda_1}, \overline{\lambda_2})$. In Figure 4, corresponding to the simple profile $(\overline{\lambda_1}, \overline{\lambda_2})$ generated by simple strategies $\overline{\lambda_1} = \{e_1\}$ and $\overline{\lambda_2} = \{e_2\}$, we have $0 < t_{e_1} < 1, 0 < t_{e_2} < 1$ and $t_{e_1} + t_{e_2} = 1$. The payoff is (t_{e_1}, t_{e_2}). Thus, $\lambda_1(l) = (0.7, e_1), \lambda_2(m) = (0.3, e_2)$ is a profile in $(\overline{\lambda_1}, \overline{\lambda_2})$ with payoff 0.7 to player 1 while $\lambda'_1(l) = (0.32, e_1), \lambda_2(m) = (0.68, e_2)$ is another profile in $(\overline{\lambda_1}, \overline{\lambda_2})$ with payoff 0.32 to player 1. All profiles in $(\overline{\lambda_1}, \overline{\lambda_2})$ will be such that player 1 payoff is bounded above by 1.

5.2 The Number of Simple Profiles

Let $\mathcal{G}$ be a timed game where the maximum cost assigned along any edge and any location for a player is $m \in \mathbb{N}$. Then, at each location, a player can have atmost $\Gamma = (2 * c_m + 1) * 2 * (m + 1)^2$ number of outgoing edges considering constraints, resets and edge costs. Thus, the total number of outgoing edges from a location is $2 * \Gamma$. In any run generated by a simple profile $(\overline{\lambda_1}, \overline{\lambda_2})$, we can assume that there are no cycles or loops since the objective of each player is to minimize her cost; this restriction is easily implementable while looking at a run generated by a simple profile. The total number of outcomes possible to one target location from the initial location is $[2 * \Gamma]^{|L|-1}$. The *outcome* of each simple profile is a unique run from an initial location to a final location. Therefore, the number of simple profiles we need to look is same as the number of runs ending in a final location which is $[2 * \Gamma]^{|L|-1} \times |F|$.

We call a simple profile $(\overline{\lambda_1}, \overline{\lambda_2})$ a *simple NE* iff all the terminal strategy profiles $(\lambda_1, \lambda_2) \in (\overline{\lambda_1}, \overline{\lambda_2})$ are NEs.

5.3 Algorithm to Calculate Simple NE

In this section, we describe an algorithm that outputs all the simple NEs of player 1 which are bounded by a constant $B \in \mathbb{N}$. We first give a result which is easy to observe:

Lemma 1. *Let $\mathcal{G}$ be a timed game. It is possible to construct from $\mathcal{G}$ a timed game $\mathcal{G}'$ with the same set of locations as $\mathcal{G}$, but where every edge with an integral constraint $x = d?$ for $d \in \mathbb{N}$ has a reset $x := 0$ such that the following holds: for every outcome ρ in $\mathcal{G}$ of a strategy profile (λ_1, λ_2) with payoff (u_1, u_2), there exists an outcome ρ' in $\mathcal{G}'$ of a strategy profile (λ'_1, λ'_2) passing through exactly the same locations as ρ, with the same payoff (u_1, u_2). The same observation holds if we start with an outcome ρ' in $\mathcal{G}'$.*

Given an outcome $\rho = (l_0, \nu_0, \mu_0) \xrightarrow{e_1} (l_1, \nu_1, \mu_1) \ldots \xrightarrow{e_n} (l_n, \nu_{n+1}, \mu_{n+1})$ of a simple profile $(\overline{\lambda_1}, \overline{\lambda_2})$, let A_i denote the set of locations in ρ where player i incurred a non-zero cost. Along a path ρ of length n, incrementally construct the sets B_{ji} for player i where $0 \leq j \leq n$ and $i = 1, 2$ as follows: Initially, $B_{ji} = \emptyset$ for all $0 \leq j \leq n$. Along ρ, when $e \in E_i$, update $B_{j+1i} = B_{ji} \cup \{l\}$ where e is the edge chosen from location l; otherwise, $B_{j+1i} = B_{ji}$. In the example given in Fig 4, $A_1 = \{l, n\}, A_2 = \{m\}$. $B_{11} = \{l\} = B_{21}$ while $B_{12} = \emptyset, B_{22} = \{m\}$.

Algorithm Simple NE

Input: The game structure $\mathcal{G} = (L, L_0, X, Z, P_1, P_2, E, \eta, C, F)$ and a bound B.
Output : The set of simple NEs where player 1 has a cost bounded by B.

1. For each player P_i, find the set of all simple strategies. A simple strategy is written by specifying the edge choices which get selected at the various locations. So, we write $\overline{\lambda} = (e_{p_1}, e_{p_2}, \ldots, e_{p_m})$ where $p_1, \ldots, p_m \in L$ and $e_{p_j} \in E_i$.
2. For each simple profile $(\overline{\lambda_1}, \overline{\lambda_2})$, determine its unique outcome and its payoff (z_1, z_2). If necessary, split the profiles as mentioned in Section 5.1.
3. For any simple profile, the cost z_i accumulated for player P_i is calculated as mentioned in Section 5.1. Construct the best response table of all simple profiles by marking a $*$ for a profile $(\overline{\lambda_1}, \overline{\lambda_2})$ if $\overline{\lambda_1}$ is player 1's best response (the one with least cost) to $\overline{\lambda_2}$. Similarly mark a ○ for player 2's best response.

4. Let C_i be a boolean variable which tells us whether player i can improve her payoff. For each simple profile $(\overline{\lambda_1}, \overline{\lambda_2})$ with payoff (z_1, z_2) which has a $\circledast$ in the best responses table, do $C_i = reduce\_payoff((\overline{\lambda_1}, \overline{\lambda_2}), P_i)$. If $C_1 \vee C_2$ is FALSE, then declare the pair $(\overline{\lambda_1}, \overline{\lambda_2})$ as a simple NE by putting a $\Box$ around the $\circledast$.
5. Return the set of all simple NEs (having $\boxed{\circledast}$) generated by the best responses table such that payoff for P_1 is less than B.

Algorithm Reduce_payoff

Input: A strategy profile $(\overline{\lambda_1}, \overline{\lambda_2})$ and a player P_i.
Output: TRUE(FALSE) if the payoff of P_i can (cannot) be reduced in any terminal strategy profile belonging to the given simple profile.

1. Let ρ be the outcome of $(\overline{\lambda_1}, \overline{\lambda_2})$.
2. Partition ρ into reset free paths $\rho_1, \rho_2, \ldots, \rho_k$ such that the last location of ρ_p is the first location of ρ_{p+1} for $1 \leq p \leq k-1$. Also, the last transition of each ρ_p $1 \leq p \leq k-1$ has a reset.
3. For $1 \leq p \leq k$, do the following on each ρ_p:
4. Determine A_i and set $B_{ji} = \emptyset$, for $0 \leq j \leq |\rho_p|$, $C = FALSE$.
5. Let e be the first edge of ρ_p. If $|\rho_p| = 1$, then there is a reset on e.
   - If the constraint ψ on e is of the form $x = a$?, then increment p, and proceed to the next path ρ_{p+1}.
   - If the constraint ψ on e is of the form $x \in (d, d+1)$ or (c, ∞), and $B_{1i} \cap A_i \neq \emptyset$, then set $C = TRUE$ and goto step 7.
6. If $|\rho_p| = r > 1$.
   - Let e be the first edge of ρ_p. e has a constraint of the form $x \in (d, d+1)$ or (c, ∞). Compute B_{ji} for $j = 1$
   - Proceed with the next transition. If there are no resets, compute $B_{j+1,i}$ from B_{ji} for $1 < j < r$. Repeat this step till the last transition of ρ_p is reached.
   - Consider the last edge f of ρ_p.
     - If the constraint on f is of the form $x \in (d, d+1)$ or (c, ∞) and $B_{ri} \cap A_i \neq \emptyset$, then set $C = TRUE$ and goto step 7.
     - If the constraint on f is of the form $x = a$ and $B_{ri} \cap A_i \neq \emptyset$ and $B_{ri} - A_i \neq \emptyset$, then set $C = TRUE$ and goto step 7.
7. If $C = FALSE$, then z_i cannot be reduced so that the constraints in ρ hold good and a target is reached. If $C = TRUE$, then z_i can be reduced by reducing one of the intervals contributing to z_i. Return C.

Lemma 2. *Consider a simple profile $(\overline{\lambda_1}, \overline{\lambda_2})$ such that $\overline{\lambda_1}$ and $\overline{\lambda_2}$, are best responses to each other. Algorithm reduce_payoff returns $C_1 = TRUE$ iff for some strategy $\lambda_1 \in \overline{\lambda_1}$, there exists an alternate strategy $\lambda_1' \in \overline{\lambda_1}$ such that (λ_1', λ_2) has a lower payoff for player 1 compared to that of (λ_1, λ_2). Here, $\lambda_2 \in \overline{\lambda_2}$.*

Lemma 3. *Algorithm Simple NE is correct : Given a timed game $\mathcal{G}$, algorithm Simple NE declares a simple profile $(\overline{\lambda_1}, \overline{\lambda_2})$ to be a simple NE iff it is. Moreover, if $(\overline{\lambda_1}, \overline{\lambda_2})$ is a simple profile with player 1 cost bounded by B, then for all profiles $(\lambda_1, \lambda_2) \in (\overline{\lambda_1}, \overline{\lambda_2})$, player 1 cost will be bounded by B.*

Theorem 3. *Given a timed game structure $\mathcal{G}$ with one clock, it is decidable if there exists a NE (λ_1, λ_2) for the corresponding game, in the outcome of which P_1 incurs a cost bounded by a constant B; that is, $u_1(outcome(\lambda_1, \lambda_2)) < B$.*

Proofs of all results as well as an example for the algorithm can be seen in *www.cse.iitb.ac.in/∼krishnas/icdcit12.pdf.*

References

1. Alur, R., Dill, D.L.: A Theory of Timed Automata. Theoretical Computer Science 126(2), 183–235 (1994)
2. Alur, R., La Torre, S., Pappas, G.J.: Optimal Paths in Weighted Timed Automata. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 49–62. Springer, Heidelberg (2001)
3. Behrmann, G., Fehnker, A., Hune, T., Larsen, K.G., Pettersson, P., Romijn, J., Vaandrager, F.: Minimum-Cost Reachability for Priced Timed Automata. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 147–161. Springer, Heidelberg (2001)
4. Bouyer, P., Brihaye, T., Markey, N.: Improved Undecidability Results on Weighted Timed Automata. Information Processing Letters 98(5), 188–194 (2006)
5. Bouyer, P., Brenguier, R., Markey, N.: Computing Equilibria in Two-Player Timed Games *via* Turn-Based Finite Games. In: Chatterjee, K., Henzinger, T.A. (eds.) FORMATS 2010. LNCS, vol. 6246, pp. 62–76. Springer, Heidelberg (2010)
6. Bouyer, P., Brenguier, R., Markey, N.: Nash Equilibria for Reachability Objectives in Multi-player Timed Games. In: Gastin, P., Laroussinie, F. (eds.) CONCUR 2010. LNCS, vol. 6269, pp. 192–206. Springer, Heidelberg (2010)
7. Brihaye, T., Bruyère, V., Raskin, J.: Model-Checking for Weighted Timed Automata. In: Lakhnech, Y., Yovine, S. (eds.) FORMATS 2004 and FTRTFT 2004. LNCS, vol. 3253, pp. 277–292. Springer, Heidelberg (2004)
8. Brihaye, T., Bruyère, V., Raskin, J.: On Optimal Timed Strategies. In: Pettersson, P., Yi, W. (eds.) FORMATS 2005. LNCS, vol. 3829, pp. 49–64. Springer, Heidelberg (2005)
9. Chatterjee, K., Henzinger, T.A., Prabhu, V.S.: Timed Parity Games: Complexity and Robustness. In: Cassez, F., Jard, C. (eds.) FORMATS 2008. LNCS, vol. 5215, pp. 124–140. Springer, Heidelberg (2008)
10. Daskalakis, C., Schoenebeck, G., Valiant, G., Valiant, P.: On the complexity of Nash equilibria of action-graph games. In: Proceedings of SODA 2009, pp. 710–719 (2009)
11. de Alfaro, L., Faella, M., Henzinger, T.A., Majumdar, R., Stoelinga, M.: The Element of Surprise in Timed Games. In: Amadio, R.M., Lugiez, D. (eds.) CONCUR 2003. LNCS, vol. 2761, pp. 144–158. Springer, Heidelberg (2003)
12. Minsky, M.L.: Computation: finite and infinite machines. Prentice-Hall Inc., USA (1967)
13. Ummels, M.: The Complexity of Nash Equilibria in Infinite Multiplayer Games. In: Amadio, R.M. (ed.) FOSSACS 2008. LNCS, vol. 4962, pp. 20–34. Springer, Heidelberg (2008)
14. Ummels, M., Wojtczak, D.: The Complexity of Nash Equilibria in Simple Stochastic Multiplayer Games. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5556, pp. 297–308. Springer, Heidelberg (2009)
15. Ummels, M., Wojtczak, D.: Decision Problems for Nash Equilibria in Stochastic Games. In: Grädel, E., Kahle, R. (eds.) CSL 2009. LNCS, vol. 5771, pp. 515–529. Springer, Heidelberg (2009)

Parallelization of PageRank on Multicore Processors

Tarun Kumar[1], Parikshit Sondhi[2], and Ankush Mittal[3,*]

[1] Samsung Noida Mobile Center, Noida
tarun.krgtm2002@gmail.com
[2] Department of Computer Science University of Illinois at Urbana Champaign
sondhi1.uiuc@gmail.com
[3] College of Engineering Roorkee
dr.ankush.mittal@gmail.com

Abstract. PageRank is a prominent metric used by search engines for ranking of search results. Page rank of a particular web page is a function of page ranks of all the web pages pointing to this page. The algorithm works on a large number of web pages and is thus computational intensive. The need of hardware is currently served by connecting thousands of computers in cluster. But faster and less complex alternatives to this system can be found in multicore processors. In this paper, we identify major issues involved in porting PageRank algorithm on Cell BE Processor and CUDA, and their possible solutions. The work is evaluated on three input graphs of different sizes ranging from 0.35 million nodes to 1.3 million. Our results show that PageRank algorithm runs 2.8 times fast on CUDA compared to Xeon dual core 3.0 GHz.

Keywords: Cell BE Processor, CUDA, Multicore Processor, PageRank, Web Graph.

1 Introduction

PAGERANK of a webpage, first introduced by Google is a prominent characteristic used by search engines while ranking of search results [1]. PageRank first introduced in [2], exploits the link structure of web. It assigns a relative importance measure called rank of the page, to each web page.

Rank of a particular page depends upon the rank of the web pages linking to this page. Higher the page rank more important is the page. PageRank algorithm itself is computational intensive and it has to work upon billions of web pages. It takes time in order of days [3] to solve the PageRank. Web pages are updated, added, removed to and from WWW continually, therefore the frequent computation of rank of pages is required. Besides this, some applications of PageRank like topic sensitive search and personalized web search require large number of page rank scores recomputed to reflect the user preferences [4]. Thus, some new ways to calculate rank of web pages in minimum possible time are always sought. A recent breakthrough with introduction of the Multicore Processors has provided a new alternative for solving computational intensive algorithms in efficient ways in terms of time.

[*] The work presented in this paper was done while author was in IIT Roorkee.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 129–140, 2012.

In this paper, we identify the issues and their possible solutions of porting PageRank algorithm on cell BE Processor followed by the implementation of PageRank algorithm on Cell BE. We also provide an implementation of PageRank algorithm on CUDA. A comparison of Cell BE and CUDA is presented on the bases of time taken to compute PageRank algorithm for standard web graphs.

The rest of the paper is organized as follows: Section 2 describes the related work on PageRank algorithm. Section 3 provides the background information on multicore architecture and PageRank algorithm. Section 4 provides implementation of PageRank on Cell BE Processor and CUDA. Section 5 shows the results. Section VI concludes the work and suggests the future work.

2 Related Work

There has been a sincere effort to reduce the time of computation of PageRank algorithm. Chen et. al. [5] has proposed some I/O efficient technique to reduce the disk reads and writes. They analyzed the link structure of the web in detail and perform the preprocessing of the web graph and propose IO efficient algorithm. Their approach shows significant benefits over original PageRank algorithm when main memory of the system to be worked upon is very small of the order of MBs. But in real scenario main memory size has been increased very much therefore their approach becomes of no use.

Another technique for solving rank of web pages which exploits block structure of web was presented by Kamvar et. al. [6]. Web graph has majority of hyperlinks which link pages on a host to other pages on the same host, many of those that do not link pages to within the same domain. They exploited this structure of web and achieved a speedup of 2 times with this approach. Manaskasemsak et. al. [7] presented a parallel PageRank Computation on a Gigabit PC cluster and showed significant improvement.

PageRank is a highly computational intensive and Cell BE Processor is also designed for computational intensive algorithm. With this idea, Buehrer et. al. [8] implemented PageRank algorithm on Cell BE. But, because of large number of random memory writes, and data transfer between PPE and SPE required by PageRank algorithm, implementation took more time than on single processor Xeon. They also presented a comparison of time taken by different processors to calculate ranks of pages for a particular graph and found that their implementation on Cell BE is 22 times slow in comparison to Xeon processor.

3 Background

3.1 Multicore Processors

A multi-core processor is an integrated circuit to which two or more processors are attached for enhanced performance, reduced power consumption, and more efficient simultaneous processing of multiple tasks. In this section we describe two multi-core architectures- STI Cell BE and CUDA.

The Cell BE [9] is a heterogeneous multi-core chip that is significantly different from conventional multiprocessors. It consists of a central microprocessor called the

Power processing element (PPE), eight SIMD co-processing units called synergistic processor elements (SPE), a high speed memory controller, and a high bandwidth bus interface, all integrated on a single chip. It has a 128 registers of 128 bits. Serje et. al. [10] and Kurzak et. al. [11] show a significant improvement in their implementations.

General purpose computing on the GPU (Graphics Processing Unit) is an active area of research. The GPU contains hundreds of cores that work great for parallel implementation. CUDA (Compute Unified Device Architecture) allows GPUs to be programmed using a variation of C language. GPUs have been proved very efficient for highly computational algorithms [12, 13].

3.2 PageRank

PageRank is an algorithm to determine the relative ranking of web pages. The concept of PageRank is based on an idea that if a page v of interest has many other pages u with pointing to , then the pages u are implicitly conferring some importance to page v. Let C(u) be the number of links which page u points out, and let PR(u) be the rank of page u, then hyperlink $u \rightarrow v$ confers $PR(u)/C(u)$ units of rank to page v.

$$PR_i(A) = (1-d) + d * \left(\frac{PR_{i-1}(T_1)}{C(T_1)} + - - - - - - - - + \frac{PR_{i-1}(T_n)}{C(T_n)} \right) \quad (1)$$

Where, $PR_i(A)$ is the PageRank of page A calculated in ith iteration. $PR_{i-1}(Ti)$ is the PageRank of page Ti which link to page A, calculated in i-1th iteration. C(Ti) is the number of outbound links on page Ti and d is a damping factor which can be set between 0 and 1.

(dest_id)	(in_degree)	(Source_nodes)
1	2	3 7
2	4	4 5 7 9
3	3	2 7 9
4	1	1
-	-	-----

Fig. 1. Structure of file containing Web graph

WWW can be considered as a directed graph where each web page is treated as a node of graph and hyperlinks as edges of graph which is known as web graph. Every node of web graph has some number of forward and backward links. A web graph is the input to the PageRank algorithm and stored into a text file as shown in figure 1.

4 Implementation Method

4.1 Implementation of PageRank on Cell BE Processor

Implementation Issues

1. PageRank operates on a huge amount of data. To achieve a better performance gain, all calculations should be done on SPEs. Since SPE's local store is small

(256 KB) and data to be worked upon is large and available at PPE, therefore a large number of DMA transfer need to be done between PPE and SPE producing a bottleneck in performance.

2. Rank of a particular node depends upon any number of nodes in the complete range of nodes. That means data to be worked upon is not continuous (rather scattered in memory arbitrarily). So on the direct input, data level parallelism is not possible. To achieve data level parallelism some modification are required.
3. Since DMA is done on sequential data while requirement in PageRank is of any random node. Thus many DMA operations may be required for smaller data.

Design

We provide data structures followed by algorithm. The data structure design is follows:

1. The web graph is read on PPE and stored into two arrays such that,
   a. Array1 (referred as Node array) contains
      i. First node followed by its in-degree, followed by the source nodes, then
      ii. Second node followed by its in-degree, followed by its source nodes then
      iii. Third node and so on.

      n1| deg1| s1|s2|s3.....|n2|deg2|s4|s5|s6|.....

      here n1, n2 are nodes.

      deg1, deg2 are in-degree of n1, n2 respectively and s1, s2 represent the source node to node n1.

   b. Array2 (referred as Degree array) contains the out-degree of nodes in the indices corresponding to source nodes in Array1

      n1|deg1|d1|d2|d3|....|n2|deg2|d4|d5|d6......

      here n1 and n2 are same as in array1, deg1 and deg2 are same as in array1 but di represents the out-degree of node si (present in array1)
2. Two arrays V1 and V2 are maintained to keep rank of nodes at ith and (i+1)th. V1 is used as a reference array containing the page ranks as calculated from the previous iterations and used in calculation of V2.

Size of V1 is equal to the size of array1 and array2 while size of V2 is equal to number of nodes. Here thing to be noted is that V1 contains rank of all nodes in the sequence same to the sequence of nodes of array1. That means there is redundancy of rank value of a particular node several times in V1. This is because, a particular node may be the source node of multiple nodes and hence present multiple times in array1. Algorithm proceeds in following way,

1. Array V1 is initialized to 1.
2. for each iteration,

i. Array2 (or Degree array) and V1 are equally divided among number of SPEs.

ii. Since number of nodes dedicated to an SPE is large so SPE reads array2 and V1 in parts. In one time SPE reads as many elements of arry2 and V1 as it can accommodate in its local store. Since a particular node, its in-degree, source nodes and their out-degree all are present in array2 and V1 so rank of node can be calculated easily, and same section of V1 need not be read again for calculations of two nodes. As soon as the rank of nodes is calculated in one time it is sent back to the PPE where it is stored in V2.

iii. As soon as all SPEs calculate the rank of all nodes dedicated to them and V2 is updated at PPE, V1 is updated from V2.

Implementation Details

PPE Operations

Processing starts with PPE by reading the web graph and preparing data structures. PPE spawns pthreads equal in number to SPEs. Each pthread spawns an SPE thread. As soon as SPE thread is created, it starts calculating rank of nodes assigned to it. During this time PPE's pthread goes into a blocking wait giving control to other pthreads of PPE while waiting for a signal by SPE. Figure 2 shows the overall working of PPE and SPEs for one iteration of PageRank algorithm.

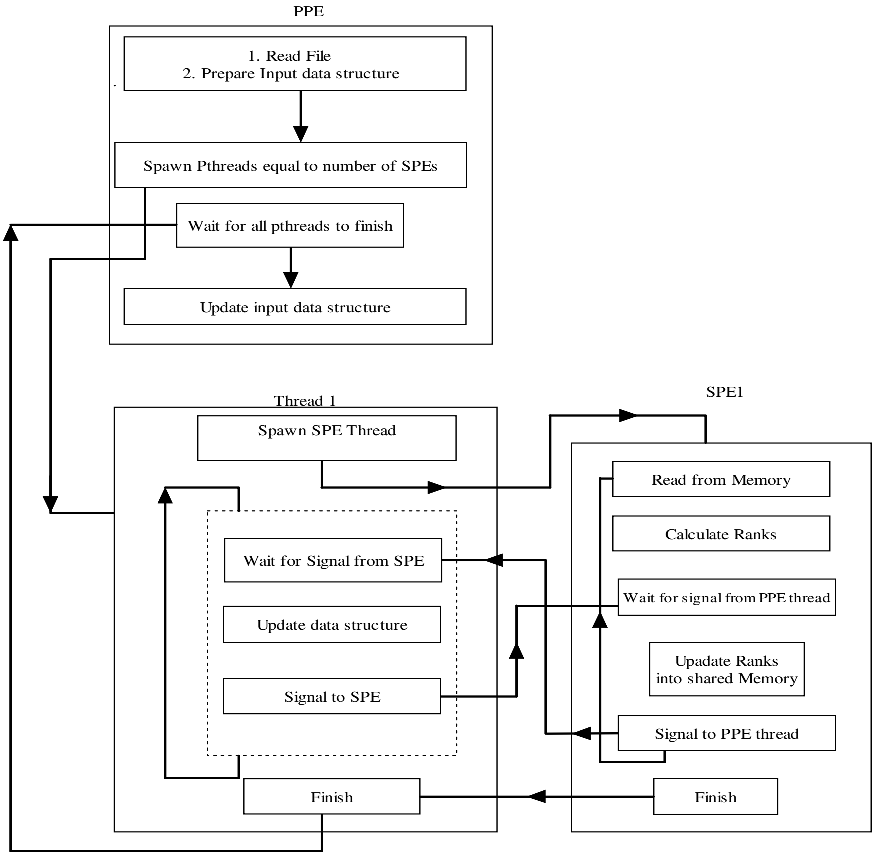


Fig. 2. Overall working of Cell BE processor for PageRank Algorithm

Pthreads update their data structures from the shared memory which is updated by corresponding SPE. Shared memory synchronization is required between pthread and SPE.

SPE Operations

As soon as the SPE thread is created by PPE, SPE starts reading two arrays of degree and rank. Since the task is equally divided among all SPEs so each SPE reads from a particular array location which is determined by the number of SPE. Each SPE starts reading at (SIZE/N)*i location where size of arrays is given by SIZE, N represents total number of SPEs and i is between 0 to N-1 for different SPEs. SPE reads data from memory through DMA operation. Since DMA transfers are limited by 16KB per transfer, therefore only 4096 integer elements can be brought at one time. Thus 4096 elements of degree array and 4096 elements of rank array are brought by two successive DMA transfer. The calculation of PageRank is done with SIMD operation. New rank of nodes present in these 4096 elements is calculated and sent back to PPE by writing into the shared memory. Before writing into shared memory SPE waits for a signal by PPE. After writing into the shared memory SPE sends a signal to PPE about the update of memory and start reading next data from input arrays.

Synchronization

Communication and shared memory synchronization between PPE thread and SPE is achieved with mailbox. The mailbox used is SPE write outbound mailbox. The SPE informs PPE each time after updating shared memory. The status of mailbox at PPE is 1 when mailbox is full and 0 when mailbox has been read by PPE while at SPE its value is 1 when there is no data in it and 0 when SPE writes data in mailbox. The communication synchronization between PPE and SPE is shown is figure 3.

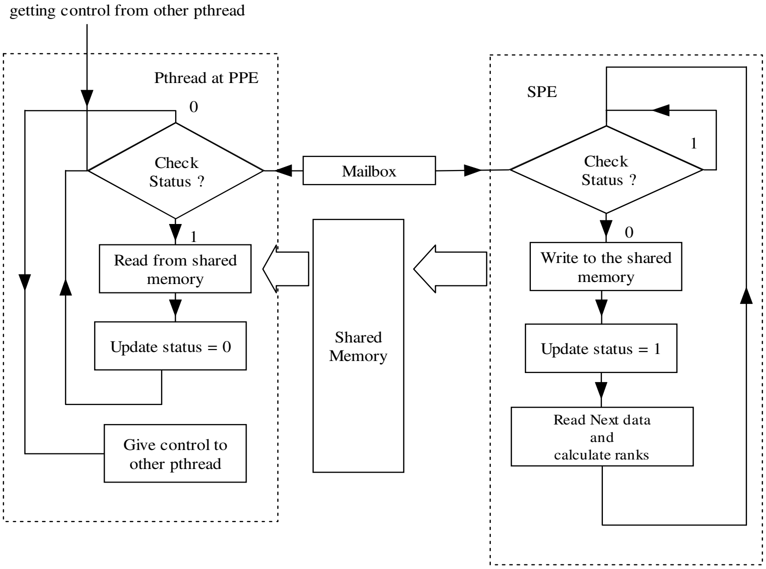


Fig. 3. Synchronization between PPE thread and SPE

4.2 Implementation of PageRank on CUDA

Implementation Issues

1. Architecture of CUDA requires threads of same code path to be running in parallel on a multiprocessor. Execution on CUDA takes place in form of warps. Warps are 32 thread units that are executed on a multiprocessor. CUDA stops all divergent threads within a warp. So if any branch statement is encountered the amount of parallelization gets reduced as divergent threads are no longer running in parallel. Real world scenario web graphs have varying number of in-degrees for nodes. Now processing in the PageRank algorithm works on every node. We have initiated one thread for every node. Since these nodes have varying number of in-degrees the amount of iterations performed by every thread is different which creates a number of divergent threads.
2. Memory synchronization constructs for global memory are not available in CUDA.
3. CUDA does not provide atomic statements for floating point values while PageRank works totally on floating point values.
4. PageRank calculation performs read operations from a wide range of memory area with very less localization of reference. This generates a lot of page faults.

Design

CUDA provides the facility of generating tens of thousands of threads at a time with no generation time. These threads can run on multiprocessors of GPU in parallel. CUDA threads run on different multiprocessors simultaneously, therefore there should not be any data dependency among threads.

Design 1

PageRank algorithm calculates rank of nodes such that rank of a particular node depends on rank of other nodes which have a link with that node. There is no data dependency present between any two nodes of web graph. We create threads equal to the number of nodes. CUDA's limitations disallow this direct approach of solving PageRank algorithm efficiently. The limitations of CUDA with respect to PageRank are as follows:

Number of nodes in a web graph is very large (of the order of billion) and such a large number of threads cannot be generated on GPU (limited by hardware).

Rank of a node may depend upon any number of nodes; therefore a loop to calculate rank of different nodes runs for different number of iterations and hence causing different code paths for threads. This means a large number of threads diverge and they cannot run in parallel.

The above stated problems were further eliminated in the following manner.

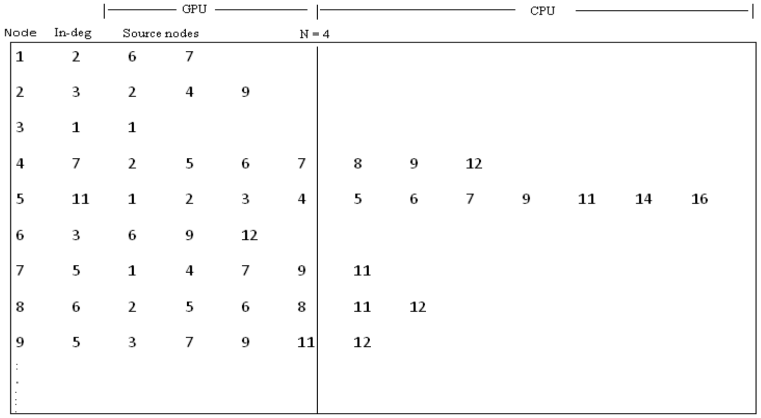
Threads corresponding to all nodes should not be spawned at the same time; therefore threads are created in multiple passes in a loop.

To avoid the problem of divergence an extra level of parallelism is added. Instead of calculating rank of a node on a single thread, rank of one node is calculated by as many threads as the in degree of node. We create threads equal to the total in-degree of all the nodes. For each node, threads equal to its in-degree calculate parallely their respective shares of rank and add that share to the rank of node (which is kept 0 initially). This causes threads to have equal amount of work to be done and hence the code path is same for each thread. This approach requires the rank of a node modified by several threads running in parallel which causes the problem of synchronization among the threads. CUDA does not provide synchronization tools for global memory. Though it provides atomic operations (means once a thread is using a particular memory location no other thread can use that location) for integers only but PageRank requires floating point values. Thus this approach could not be used.

Design 2

The main problem with design 1 is that it hinders the performance because of the variable loop length of each thread. In order to avoid this problem, we run a fixed length loop (say N) on GPU for all threads. Value of N depends upon the web graph to be worked upon. Ideally we want most of the computations to be performed on GPU. GPU prefers threads of similar amount of computation. We select N in such a way that more GPU threads are similar in computation. The idea is to run GPU and CPU parallely such that while GPU is running loop of length N for all threads, CPU calculates partial rank of those nodes which have in-degree more than N by running a loop from N to in-degree of the node.

Figure 4 shows an example of small web graph. Value of N is kept 4. Rank of all nodes is calculated with 4 (or less) source nodes at GPU. Host at the same time calculates partial rank of nodes 4, 5, 7, 8, 11 with those source nodes participating that have index more than N (= 4) . For example for node 4 partial ranks with source nodes 2, 5, 6, 7 is calculated on GPU and partial rank with source nodes 8, 9, 12 is calculated at Host.



		GPU				CPU						
Node	In-deg	Source nodes			N = 4							
1	2	6	7									
2	3	2	4	9								
3	1	1										
4	7	2	5	6	7	8	9	12				
5	11	1	2	3	4	5	6	7	9	11	14	16
6	3	6	9	12								
7	5	1	4	7	9	11						
8	6	2	5	6	8	11	12					
9	5	3	7	9	11	12						
⋮												

Fig. 4. An example showing the division of work between GPU and CPU

The work of PageRank calculation is divided onto GPU and Host in such a way that when GPU is calculating partial rank of all nodes with the help of N (or less) source nodes, Host at same time calculates partial rank of all nodes with remaining source nodes (other than N nodes if any). GPU calculates the share of N source nodes which point to a destination node by running loop N times for each thread. Host calculates share of rest of the nodes by running a loop from N to their corresponding in degree. In this way host and GPU calculate partial rank of nodes. These partial ranks are then added and used for next iteration.

Implementation Details

This section presents implementation details of design 2 described in previous section. Program for calculating PageRank consists of mainly two parts, host and kernel. Figure 5 shows the overall control flow of PageRank calculation on CUDA. Execution of PageRank algorithm starts with host program.

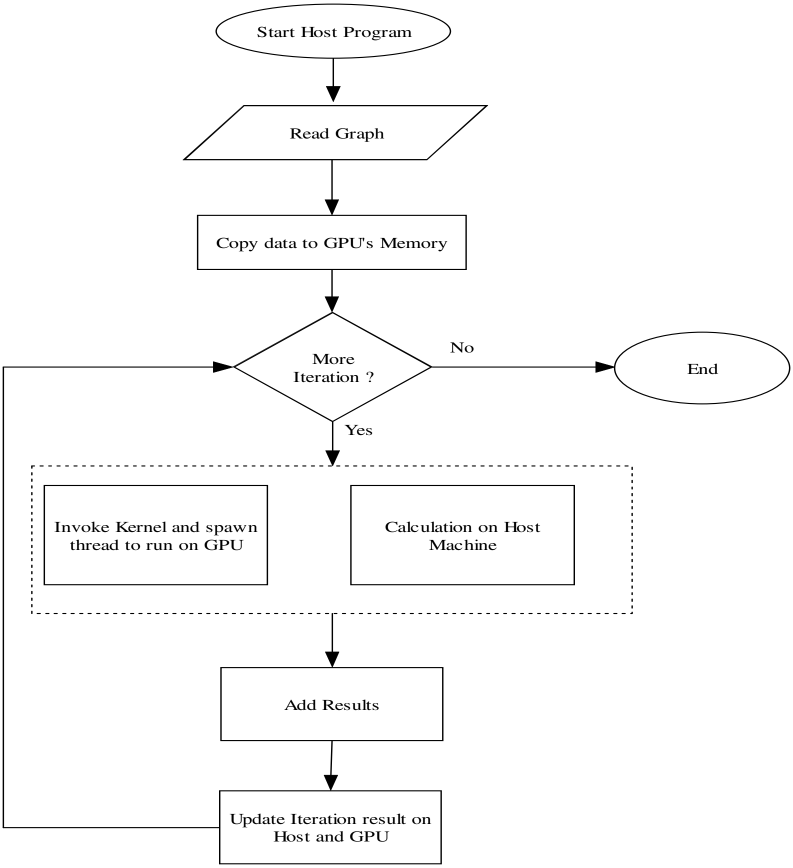

Start Host Program
Read Graph
Copy data to GPU's Memory
More Iteration ?
No
End
Yes
Invoke Kernel and spawn thread to run on GPU
Calculation on Host Machine
Add Results
Update Iteration result on Host and GPU

Fig. 5. Program flow of PageRank calculation on CUDA

Host program reads the web graph and copies it to the GPU's memory. Host invokes the kernel to be run on GPU for calculating partial rank with N source nodes. GPU starts processing. Since kernel calls are non-blocking for host, therefore Host also starts rank calculation of nodes having in-degree more than N. As soon as calculation on both GPU and Host are finished, partial rank of all nodes from Host is brought into GPU memory and added with partial rank calculated at GPU. Thus the new rank of all nodes is calculated and input vectors for next iteration is updated both at GPU and Host.

5 Results

The PageRank algorithm works upon a large web graph in practice therefore; the web graphs which are used for experiments are EU – 2005, CNR-2000 and In-2004 [14]. EU-2005 graph contains 862664 nodes and 19235140 links. CNR-2000 contains 325557 nodes and 3216152 links. Graph In-2004 contains 1382908 nodes and 18534900 links. These graphs are prepared with ubi-Crawler [15].The Cell processor used for execution and testing results is Cellbuzz provided by Georgia Tech. University [16]. GPU used for experiments is GTX 280. Figure 6 shows the comparison of execution times between XEON dual core 2.0 GHz and CELL BE for graphs EU-2005 and In-2004.

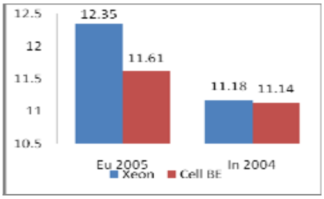


Fig. 6. Comparison of time between XEON and CELL BE processor for graph EU-2005 and In-2004

The speed up obtained by Cell BE over Xeon does not show a marked improvement but when compared to Brehrer et. al. [8]'s implementation of PageRank algorithm on Cell BE, our implementation of algorithm is 22 times faster. We also compare the time taken by Xeon dual core 3.0 GHz and CUDA. It is observed that implementation on CUDA is nearly 2.8 times fast for graph EU-2005. Figure 7 shows the comparison of timing on Xeon and CUDA for graph EU-2005 and CNR-2000.

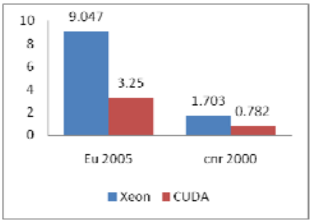


Fig. 7. Comparison of timing on Xeon and CUDA processors for graph EU 2005 and CNR 2000

We also compare the implementation on CUDA with Cell BE, it is found that CUDA performs much better. Figure 8 shows a comparison of timing on Xeon dual core 2.0 GHz, Cell BE and CUDA processor. It shows that implementation on CUDA is 2.6 times faster than implementation on Cell BE.

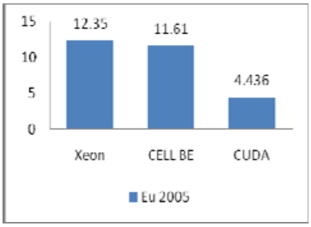


Fig. 8. Comparison of timing on Xeon and CUDA and Cell BE processors on graph EU 2005

6 Conclusion and Future Work

In this paper, we identified the major issues of porting PageRank algorithm on Cell Processor. Possible solutions to these issues were presented. Previous implementation of PageRank on Cell BE resulted in poor performance because of the high data transfer operation between PPE and SPE. A new approach is implemented which reduces the data transfer between PPE and SPE drastically and leads to a better performance. We also presented issues of porting PageRank algorithm on CUDA followed by its implementation on CUDA. It was found that implementation of PageRank on CUDA is performing much better than on Cell.

A better performance in current implementation can be found by dividing the graph in small blocks and then determine the value of N (number of iterations to be run on GPU for a thread) for each block. Performance of PageRank algorithm on multicore processor can be improved by analyzing the web graph in detail and preprocess it according to the restrictions and features of multicore architecture. Another possible solution that can be used for improving performance is to sort the web graph on the bases of in-degree of nodes.

References

1. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia, pp. 107–117 (April 1998)
2. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web. Stanford Digital Library Working Paper (1998)
3. PageRank Google's Original Sin, `http://www.google-watch.org/pagerank.html`
4. Haveliwala, T.H.: Topic Sensitive PageRank. IEEE Transactions on Knowledge and Data Engineering 15(4), 784–796 (2003)

5. Chen, Y.Y., Gan, Q., Suel, T.: I/O-efficient techniques for computing PageRank. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, McLean, Virginia, USA, pp. 549–557 (2002)
6. Kamvar, S.D., Haveliwala, T.H., Manning, C.D., Golub, G.H.: Exploting the block Structure of the Web for Computing PageRank. Technical Report CSSM-03-02, Computer Science Department, Stanford University (2003)
7. Manaskasemsak, B., Rungsawang, A.: Parallel PageRank Computation on gigabit PC Cluster. In: Proceedings of 18th International Conference on Advanced Information Networking and Applications AINA, Fukuoka Japan, vol. 1, pp. 273–277 (March 2004)
8. Buehrer, G., Parthasarathy, S., Goyder, M.: Data mining on the cell broadband engine. In: Proceedings of the 22nd Annual International Conference on Supercomputing, Island of Kos, Greece, pp. 26–35 (June 2008)
9. Cell Broadband Engine – An introduction. Cell Programming Workshop, IBM System and Technology Group, April 14–18 (2007)
10. Sarje, A., Aluru, S.: Parallel Genomic Alignments on the Cell Broadband Engine. IEEE Transactions on Parallel and Distributed Systems, December 09 (2008)
11. Kurzak, J., Buttari, A., Dongarra, J.: Solving Systems of Linear Equations on the CELL Processor Using Cholesky Factorization. IEEE Transactions on Parallel and Distributed Systems 19(9), 1175–1186 (2008)
12. Liu, W., Schmidt, B., Voss, G., Muller-Wittig, W.: Streaming Algorithms for Biological Sequence Alignment on GPUs. IEEE Transactions on Parallel and Distributed Systems 18(9), 1270–1281 (2007)
13. Garland, M., Le Grand, S., Nickolls, J., Anderson, J., Hardwick, J., Morton, S., Phillips, E., Zhang, Y., Volkov, V.: Parallel Computing Experiences with CUDA. IEEE Micro 28(4), 13–27 (2008)
14. Laboratory for Web Algorithmics, `http://law.dsi.unimi.it/index.php?option=com_include&Itemid=65`
15. Boldi, P., Codenotti, B., Santini, M., Vigna, S.: UbiCrawler: A Scalable Fully Distributed Web Crawler. Journal of Software: Practice & Experience 34, 711–726 (2004)
16. User guide, Cell buzz, `http://wiki.cc.gatech.edu/cellbuzz/index.php/User_Guide`

Cryptanalysis and Improvement of Sood et al.'s Dynamic ID-Based Authentication Scheme

Chun-Guang Ma[1], Ding Wang[1,2,*], and Qi-Ming Zhang[1]

[1] College of Computer Science and Technology, Harbin Engineering University
145 Nantong Street, Harbin City 150001, China
wangdingg@mail.nankai.edu.cn
[2] Automobile Management Institute of PLA, Bengbu City 233011, China

Abstract. Anonymity is one of the important properties of remote authentication schemes to preserve user privacy. Recently, Sood et al. showed that Wang et al.'s dynamic ID-based remote user authentication scheme fails to preserve user anonymity and is vulnerable to various attacks if the smart card is non-tamper resistant. Consequently, an improved version of dynamic ID-based authentication scheme was proposed and claimed that it is efficient and secure. In this paper, however, we will show that Sood et al.'s scheme still cannot preserve user anonymity under their assumption. In addition, their scheme is also vulnerable to the offline password guessing attack and the stolen verifier attack. To remedy these security flaws, we propose an enhanced authentication scheme, which covers all the identified weaknesses of Sood et al.'s scheme and is more secure and efficient for practical application environment.

Keywords: Dynamic ID, Authentication protocol, Non-tamper resistant, Smart card, Cryptanalysis, Anonymity.

1 Introduction

With the significant advances in communication networks over the last couple of decades, smart cards have been widely used in many ecommerce applications and network security protocols due to their low cost, portability, efficiency and cryptographic properties. Smart card authentication is based on different techniques such as passwords, digital certificates, digital signature and biometric technology. Among these techniques, password is the most commonly used authentication technique to authenticate users on the server due to its simplicity and convenience. Except efficiency and convenience, there are also many other desirable properties of a secure remote authentication scheme, such as freedom of choosing passwords, mutual authentication, session key generation, forward secrecy and user anonymity.

Recently, because of the advantages of smart cards, a number of password-based authentication schemes with smart cards have been proposed [1-6]. Most of the proposed schemes assume that the smart card is tamper-resistant, i.e., the secret information stored in the smart card cannot be revealed. However, recent research

[*] Corresponding author.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 141–152, 2012.

results have shown that the secret information stored in the smart card could be extracted by some means, such as monitoring the power consumption [7] or analyzing the leaked information [8]. Therefore, such schemes based on the tamper resistance assumption of the smart card are prone to some types of attacks, such as user impersonation attacks, server masquerading attacks, and offline password guessing attacks, etc., once an adversary has obtained the secret information stored in a user's smart card and/or just some intermediate computational results in the smart card.

A common feature among most of the published schemes is that the user's identity is static in all the transaction sessions, which may leak the identity of the logging user once the login messages were eavesdropped, hence user anonymity is not preserved. The leakage of the user identity may also cause an unauthorized entity to track the user's login history and current location. Therefore, assuring anonymity does not only preserve user privacy but also make remote user authentication protocols more secure. One of the solutions to preserve user anonymity is to employ dynamic ID in different login requests. In 2004, Das et al. [9] first introduced the concept of dynamic ID authentication scheme to resist ID-theft and thus to achieve user anonymity. However, in 2005, Chien and Chen [10] pointed out that Das et al.'s scheme fails to protect the user's anonymity, so they proposed a new one. In 2009, to overcome the security pitfalls of Das et al.'s scheme, Wang et al. [11] also proposed a dynamic ID-based authentication scheme, and claimed that their scheme is more efficient and secure while keeping the merits of Das et al.'s scheme.

All of the above three dynamic ID authentication schemes are based on the tamper-resistant assumption of the smart card. However, it is a challenge that the smart card is non-tamper resistant while preserving user anonymity. In 2007, Hu et al. [12] showed that Chien-Chen's scheme is vulnerable to the strong masquerading server/user attack, if the smart card is no longer tamper-resistant, and then they proposed an improved scheme. Later on, Horng et al. [13] showed that Hu et al.'s scheme is still vulnerable to the strong masquerading server/user attack and the offline password guessing attack. Therefore, Horng et al. proposed an improvement over Hu et al.'s scheme to remedy their drawbacks. In 2010, Yeh et al. [14] pointed out that Wang et al.'s scheme is insecure against replay attack, impersonation attack, man-in-the-middle attack and password guessing attacks. In 2011, Khan et al. [15] found Wang et al.'s scheme also does not provide user anonymity, session key agreement and revocation of lost smart card.

In 2011, Sood et al. [16] also identified that Wang et al.'s scheme cannot withstand various attacks stated above and further proposed an enhanced remote authentication scheme. They claimed their scheme is efficient and can overcome all the identified security drawbacks of Wang et al.'s scheme even if the smart card is non-tamper resistant. Unfortunately, in this paper, however, we will demonstrate that Sood et al.'s scheme cannot withstand stolen verifier attack and is still vulnerable to offline password guessing attack. And to our surprise, user anonymity, which is the most essential security feature a dynamic ID authentication scheme is designed to support, cannot be preserved. To conquer the aforementioned weaknesses, an enhancement of Sood et al.'s scheme is presented.

The remainder of this paper is organized as follows: in Section 2, we review Sood et al.'s authentication scheme. Section 3 describes the weaknesses of Sood et al.'s scheme. Our improved scheme is presented in Section 4, and its security analysis is given in Section 5. The comparison of the performance of our scheme with the other related schemes is shown in Section 6. Section 7 concludes the paper.

2 Review of Sood et al.'s scheme

In this section, we examine the dynamic ID authentication scheme using smart cards proposed by Sood et al. [16] in 2011. Sood et al.'s scheme consists of four phases: the registration phase, the login phase, the verification and session key agreement phase and the password change phase. For ease of presentation, we employ some intuitive abbreviations and notations listed in Table 1.

Table 1. Notations

Symbol	Description
U_i	i^{th} user
S	remote server
ID_i	identity of user U_i
P_i	password of user U_i
x	master secret of remote server S
y_i	a random value corresponding to user U_i
p,q,n	p and p are two large prime numbers, and n=pq
e,d	e is a prime number and d is an integer, where ed=1mod(p-1)(q-1)
$H(\cdot)$	collision free one-way hash function
$\oplus$	the bitwise XOR operation
$\|\|$	the string concatenation operation
$A \Rightarrow B:M$	Message M is transferred through a secure channel from A to B
$A \rightarrow B:M$	Message M is transferred through a common channel from A to B

2.1 Registration Phase

The registration phase involves the following operations:

Step R1. Server S authenticates itself to the user U_i using its public key certificate. Then U_i generates and encrypts the session key (SS) with the public key (PK) of the server S as $(SS)_{PK}$.

Step R2. $U_i \rightarrow S:(SS)_{PK}, (ID_i)_{SS}, (P_i)_{SS}$.

Step R3. On receiving the registration message from U_i, the server S decrypts the session key (SS) using its private key. Thereafter, the server S decrypt the identity $(ID_i)_{SS}$ and password $(P_i)_{SS}$. Then server S chooses random value y_i and computes $N_i=H(ID_i||P_i)\oplus H(x)$, $A_i=H(ID_i||P_i)\oplus P_i\oplus H(y_i)$, $Bi=y_i\oplus ID_i \oplus P_i$ and $D_i=H(H(ID_i||y_i)\oplus x)$. Server S chooses the value of y_i corresponding to each user to make sure D_i is unique for each user. The server S stores $y_i\oplus x$ and $ID_i\oplus H(x||y_i)$ corresponding to D_i in its database.

Step R4. $S \Rightarrow U_i$: A smart card containing security parameters (N_i, A_i, B_i, $H(\cdot)$).

2.2 Login Phase

When U_i wants to login to S, the following operations will be performed:

Step L1. U_i inserts his/her smart card into the card reader and inputs ID_i^* and P_i^*.

Step L2. The smart card computes $y_i^* = B_i \oplus ID_i^* \oplus P_i^*$, $A_i^* = H(ID_i^* || P_i^*) \oplus P_i^* \oplus H(y_i^*)$. Smart card verifies the validity of A_i^* by checking whether A_i^* equals the stored A_i. If the verification holds, the smart card computes $H(x) = N_i \oplus H(ID_i || P_i)$, $CID_i = H(ID_i || y_i) \oplus H(H(x) || T)$ and $M_i = H(ID_i || H(x) || y_i || T)$, where T is current date and time. Otherwise, the session is terminated.

Step L3. $U_i \rightarrow S$: CID_i, M_i, T.

2.3 Verification and Session Key Agreement Phase

After receiving the login request message from user U_i, server S performs the following operations:

Step A1. The server S checks the validity of timestamp T by checking (T' – T) <=δT, where T' is current date and time of the server S and δT is permissible time interval for a transmission delay. The server S computes $D_i^* = H(CID_i \oplus H(H(x) || T) \oplus x)$ and finds D_i corresponding to D_i^* in its database and then extracts $y_i \oplus x$ and $ID_i \oplus H(x || y_i)$ corresponding to D_i^* from its database. Now the server S computes y_i from $y_i \oplus x$ and ID_i from $ID_i \oplus H(x || y_i)$ because the server S knows the value of x.

Step A2. The server S computes $M_i^* = H(ID_i || H(x) || y_i || T)$ and compares M_i^* with the received value of M_i. This equivalency authenticates the legitimacy of the user U_i and the login request is accepted else the connection is terminated.

Step A3. The user U_i and the server S agree on the common session key $SK = H(H(x) || ID_i || T || y_i)$ for securing future data communications.

2.4 Password Change Phase

When U_i wants to change the password, the following operations will be performed:

Step P1. U_i inserts his/her smart card into the card reader and inputs ID_i^* and P_i^*.

Step P2. The smart card computes $y_i^* = B_i \oplus ID_i^* \oplus P_i^*$, $A_i^* = H(ID_i^* || P_i^*) \oplus P_i^* \oplus H(y_i^*)$. Smart card verifies the validity of A_i^* by checking whether A_i^* equals to the stored A_i. If it holds, ID_i^* will be equal to ID_i and P_i^* will be equal to P_i, otherwise, the smart card rejects the password change request.

Step P3. The smart card asks the cardholder to resubmit a new password P_i^{new} and computes $N_i^{new} = N_i \oplus H(ID_i || P_i) \oplus H(ID_i || P_i^{new})$, $A_i^{new} = H(ID_i || P_i^{new}) \oplus P_i^{new} \oplus H(y_i)$ and $B_i^{new} = y_i \oplus ID_i \oplus P_i^{new}$. Thereafter, smart card updates the values of N_i, A_i and B_i stored in its memory with N_i^{new}, A_i^{new} and B_i^{new} respectively.

3 Cryptanalysis of Sood et al.'s Scheme

In this section we will show that Sood et al.'s scheme fails to preserve user anonymity and is vulnerable to offline password guessing attack and stolen verifier attack. Although tamper resistant smart card is widely assumed in most of the authentication schemes, but such an assumption is difficult in practice. Many researchers have shown that the secret stored in a smartcard can be breached by analyzing the leaked information or by monitoring the power consumption [7,8]. Be aware of this threat, Sood et al. intentionally based their scheme on the assumption of non-tamper resistance of the smart card. However, Sood et al.'s scheme fails to serve its purposes.

3.1 Failure of Protecting the User's Anonymity

Let us consider the following scenarios. A malicious privileged user U_i having his own smart card can gather information $N_i = H(P_i||ID_i)\oplus H(x)$ from his own smart card. Then he can find out the value of H(x) as $H(x)=N_i\oplus H(P_i||ID_i)$ because the malicious user U_i knows his own identity ID_i and password P_i corresponding to his smart card. Then the attacker can successfully learn some sensitive user-specific information about any legitimate client through the following steps:

Step 1. Eavesdrops and intercepts a login request message (CID_k, M_k, T) of user U_k from the public communication channel.
Step 2. Computes $L_1= H\ (H\ (x)||T)$, where H (x) and T are known.
Step 3. Computes $L_2= CID_k\oplus L_1$.

Its obvious that L_2 is unconditionally equal to $H(ID_k||y_k)$, while the value of $H(ID_k||y_k)$ is kept the same for all the login requests of user U_k and is specific to user U_k. This value $H(ID_k||y_k)$ can be seen as user U_k's identification, and an attacker can, therefore, use this information to trace and identify the user U_k's requests. From the above attack, any legal user who logins to the remote server would be exposed to attacker U_i, and thus the scheme fails to achieve user anonymity, which is the most essential security feature a dynamic ID authentication scheme is designed to support.

3.2 Offline Password Guessing Attack

In Sood et al.'s scheme, a user is allowed to choose his/her own password at will during the registration and password change phases; the user usually tends to select a password, i.e., his phone number, which is easily remembered for his convenience. Hence, these easy-to-remember passwords, called weak passwords, have low entropy and thus are potentially vulnerable to offline password guessing attack. Inevitably, user's ID, chose by the user at will as described in the scheme, suffers from the same threat. Thus, the result of $ID \oplus P$ shall not be of high entropy if both user's ID and password P are human memorable and of low entropy. Therefore, the result of $ID \oplus P$ also is exposed to the same threat.

Let us consider the following scenarios. A malicious privileged user U_i having his own smart card can gather information $N_i = H(P_i||ID_i)\oplus H(x)$ from his own smart card. Then he can find out the value of H(x) as $H(x)=N_i\oplus H(P_i||ID_i)$ because the malicious user U_i knows his own identity ID_i and password P_i corresponding to his smart card. In case another user U_k's smart card is stolen by this malicious user, he can perform offline password guessing attack in the following steps:

Step 1. Extracts the information N_k, A_k and B_k in U_k's smart card.
Step 2. Computes $T_1=P_k\oplus H(y_k)=N_k\oplus A_k\oplus H(x)$, as N_k, A_k and H(x) are known.
Step 3. Computes $T_2= N_k \oplus H(x)$, as N_k and H(x) are known.
Step 4. Assumes $R= P_k\oplus ID_k$.

Step 5. Guesses the value of R to be R^* from a uniformly distributed dictionary.
Step 6. Computes $T_3=H((R^*\oplus T_1\oplus H(B_k\oplus R^*)) \| (T_1\oplus H(B_k\oplus R^*)))$.
Step 7. Verifies the correctness of R^* by checking if T_3 is equal to T_2.
Step 8. Repeats steps 4, 5, and 6 of this phase until the correct value of R is found.
Step 9. Computes $P_k= T_1\oplus H(B_k\oplus R)$, $ID_k=R\oplus P_k$.

Because $y_k= B_k\oplus R$, it is obvious that the following relationships hold true: $P_k = T_1\oplus H(y_k) =T_1\oplus H(B_k\oplus R)$, $ID_k= R\oplus P_k= R\oplus T_1\oplus H(B_k\oplus R)$ and $T_3 =H(ID_k^*\|P_k^*)$. As $N_k= H(ID_k\|P_k)\oplus H(x)$ is predefined by the authentication system, the equality of $H(ID_k\|P_k) =N_k\oplus H(x)$ will always hold. Therefore, the attacker U_i can confirm the correctness of the guessed R^* by the verification in Step 7. Once the correct value of R is obtained, the correct value of password P_k and identity ID_k can be computed in step 9. Thus, Sood et al.'s scheme cannot withstand offline password guessing attack.

After guessing the correct value of P_k and ID_k, an attacker can compute $y_k= B_k\oplus ID_k\oplus P_k$. Then the attacker can fabricate and send a valid login request message (CID_k^*, M_k^*, T_u) to the service provider server S, where T_u is the current timestamp of the attacker U_i. Hence the malicious user can successfully make a valid login request to masquerade as a legitimate user U_k.

Moreover, once the adversary obtains the correct value of P_k and ID_k, he/she can easily change the password to a new one, this causes the password change phase becoming insecure. Even if the adversary returns the changed smart card to the original user U_k, U_k will not be able to login to the remote server S. This leads to a denial of service attack.

3.3 Stolen Verifier Attack

Let us consider the following scenarios. A malicious privileged user U_i having his own smart card can gather information $B_i = y_i \oplus ID_i \oplus P_i$ from his own smart card. Then he can find out the value of y_i as $y_i = B_i \oplus ID_i \oplus P_i$ because the malicious user U_i knows his own identity ID_i and password P_i corresponding to his smart card. In case the verifier table in the database of the server S is leaked out or stolen by this malicious user, he can compute the private key x of the server S as $x= (y_i \oplus x) \oplus y_i$ because the value of y_i is known. With this x, the malicious user can compute any y_k corresponding to user U_k from the item $y_k \oplus x$ stored in the verifier table, then the malicious user can launch user/server impersonation attacks successfully. As a result, the entire authentication scheme will be compromised.

4 Our Proposed Scheme

The abbreviations and notations used in the following sections are listed in Table 1.

4.1 Registration Phase

The server S generates two large primes p and q and computes n=pq, then chooses a prime number e and an integer d, such that ed=1 mod(p-l)(q-1). Finally, the server S makes the values of n and e public, while p, q and d are only known to server S. The registration phase involves the following operations:

Step R1. Server S authenticates itself to the user U_i using its public key certificate. Then U_i generates and encrypts the session key (SS) with the public key (PK) of the server S as $(SS)_{PK}$.

Step R2. $U_i \rightarrow S$: $(SS)_{PK}$, $(ID_i)_{SS}$, $(P_i)_{SS}$.

Step R3. On receiving the registration message from U_i, the server S decrypts the session key (SS) using its private key. Thereafter, the server S decrypts the identity $(ID_i)_{SS}$ and password $(P_i)_{SS}$. Then server S chooses random value y_i and computes $N_i=H(ID_i \| P_i) \oplus H(d)$, $A_i=H(P_i \| ID_i) \oplus H(y_i)$, $B_i=y_i \oplus ID_i \oplus P_i$ and $D_i=H(H(ID_i \| y_i) \oplus d)$. Server S chooses the value of y_i corresponding to each user to make sure D_i is unique for each user. The server S stores $y_i \oplus H(H(d) \| d)$ and $ID_i \oplus H(d \| y_i)$ corresponding to D_i in its database.

Step R4. $S \Rightarrow U_i$: A smart card containing security parameters (N_i, A_i, B_i, n, e, $H(\cdot)$).

4.2 Login Phase

When U_i wants to login the system, the following operations will perform:

Step L1. U_i inserts his/her smart card into the card reader and inputs ID_i^* and P_i^*.

Step L2. The smart card computes $y_i^*= B_i \oplus ID_i^* \oplus P_i^*$, $A_i^*=H(P_i^* \| ID_i^*) \oplus H(y_i^*)$. Smart card verifies the validity of A_i^* by checking whether A_i^* equals to the stored A_i. If the verification holds, the smart card chose a random number N_u and computes $H(d)=N_i \oplus H(ID_i \| P_i)$, $CID_i=H(ID_i \| y_i) \oplus H(H(d) \| N_u \| T)$, $C_i=N_u^e \bmod n$, and $M_i=H(ID_i \| H(d) \| y_i \| T \| N_u)$, where T is current date and time. Otherwise, the session is terminated.

Step L3. $U_i \rightarrow S$: CID_i, $C_{i,}$ M_i, T.

4.3 Verification and Session Key Agreement Phase

After receiving the login request message from user U_i, server S performs the following operations:

Step A1. The server S checks the validity of timestamp T by checking $(T' - T) <= \delta T$, where T' is current date and time of the server S and δT is permissible time interval for a transmission delay. The server S decrypts the random number N_u from C_i using its private key d, then computes $D_i^* =H(CID_i \oplus H(H(d) \| N_u \| T) \oplus d)$ and finds D_i corresponding to D_i^* in its database, then extracts $y_i \oplus H(H(d) \| d)$ and $ID_i \oplus H(d \| y_i)$ corresponding to D_i^* from its database. Now the server S computes y_i from $y_i \oplus H(H(d) \| d)$ and ID_i from $ID_i \oplus H(d \| y_i)$ because the server S knows the value of d.

Step A2. The server S computes $M_i^*= H(ID_i \| H(d) \| y_i \| T)$ and compares $M_i^*$with the received value of M_i. This equivalency authenticates the legitimacy of the user U_i and the login request is accepted else the connection is terminated.

Step A3. The user U_i and the server S agree on the common session key $SK=H(H(d) \| ID_i \| T \| y_i)$ for securing future data communications.

4.4 Password Change Phase

In this phase, we argue that the user's smart card must have the ability to detect the failure times. Once the number of login failure exceeds a predefined system value, the smart card must be locked immediately to prevent the exhaustive password guessing behavior. The other parts of this phase are the same with that of Sood et al.'s scheme.

5 Security Analysis

The security of our proposed authentication scheme is based on the secure hash function and the difficulty of the large integer factorization problem. In this section, we analyze the security features provided by our scheme under the assumption that the secret information stored in the smart card can be revealed, i.e., H(d) can be obtained by a malicious privileged user.

(1) User anonymity: Suppose that the attacker has intercepted U_i's login request message (CID_i, C_i, M_i, T). Then, the adversary may try to retrieve any static parameter from the login message, but CID_i, C_i and M_i are all session-variant and indeed random strings due to the randomness of N_u. Accordingly, Without knowing the random number N_u, the adversary will face to solve the large integer factorization problem to retrieve the correct value of $H(ID_i||y_i)$ from $CID_{i,}$, while $H(ID_i||y_i)$ is the only static element in the login request. Hence, the proposed scheme can overcome the security flaw of user anonymity breach which is inherent in Sood et al.'s scheme.

(2) Offline password guessing attack: Suppose that a malicious privileged user U_i has intercepted U_k's login request message (CID_k, C_k, M_k, T) and also has got U_k's smart card. With these harsh terms and under our assumption of non-tamper resistant smart card, the secret information N_i, A_i and B_i can also be revealed. Even after gathering this information and obtaining $H(d)=N_k \oplus H(P_k||ID_k)$, the attacker has to at least guess both ID_i and P_i correctly at the same time. It impossible to guess these two parameters correctly at the same time in real polynomial time.

(3) Stolen verifier attack: In the proposed protocol, only the server S knows private secret d and stores $y_i \oplus H(H(d)||d)$ and $ID_i \oplus H(d||y_i)$ corresponding to D_i in its database. Although a malicious privileged user can compute H(d) in the way described in Section 3.1, he/she does not have any technique to find out the value of d, nor can he/she calculates y_i corresponding to other legitimate user. Therefore, the proposed protocol is secure against stolen verifier attack.

(4) User impersonation attack: As both CID_i and M_i are protected by secure one-way hash function, any modification to these two parameters of the legitimate user U_i's login request message will be detected by the server S if the attacker cannot fabricate the valid CID_i^* and M_i^*. Because the attacker has no way of obtaining the values of ID_i, P_i and y_i corresponding to user U_i, he/she can not fabricate the valid CID_i^* and M_i^*. Therefore, the proposed protocol is secure against user impersonation attack.

(5) Server masquerading attack: In the proposed protocol, a malicious server cannot compute the session key $SK = H(H(d)||ID_i||T||y_i||N_u)$ because the malicious server does not know the values of N_u, ID_i and y_i corresponding to user U_i. Moreover, the session key is session-variant for the same user U_i. Therefore, the proposed protocol is secure against server masquerading attack.

(6) Replay attack and parallel session attack: Our scheme can withstand replay attack because the authenticity of login request message (CID_i, C_i, M_i, T) is verified by checking the freshness of timestamp T. On the other hand, the presented scheme resists parallel session attack, in which an adversary may masquerade as legitimate user U_i by replaying a login request message within the valid time frame window. The attacker cannot compute the agreed session key SK between user U_i and server S because he does not know the values of N_u, ID_i and y_i corresponding to user U_i. Therefore, the resistance to replay attack and parallel session attack can be guaranteed in our protocol.

(7) Mutual authentication: In our dynamic ID-based scheme, the server authenticates the user by checking the M_i in the login request. We have shown that our scheme can preserve user anonymity, so user ID_i is only known to the server S and the user U_i itself. We have proved that our scheme can resist user impersonation attack. Therefore, it is impossible for an adversary to forge messages to masquerade as U_i in our scheme. To pass the authentication of server S, the smart card first needs U_i's identity ID_i and password P_i to get through the verification in Step L2 of the login phase. In this Section, we have shown that our scheme can resist offline password guessing attack. Therefore, only the legal user U_i who owns correct ID_i and P_i can pass the authentication of server S. On the other hand, the user U_i authenticates server S implicitly by checking whether the other party communicating with can obtain the correct session key SK = $H(H(d)||ID_i||T||y_i||N_u)$ and decrypt the encrypted messages successfully or not. Since the malicious server does not know the values of N_u, ID_i and y_i corresponding to user U_i, only the legitimate server can compute the correct session key SK. From the above analysis, we conclude that our scheme can achieve mutual authentication.

(8) Denial of service attack: Assume that an adversary has got a legitimate user U_i's smart card. However, in our scheme, smart card checks the validity of user identity ID_i and password P_i before the password update procedure. Since the smart card computes $A_i^* = H\ (P_i^*||ID_i^*))\oplus H(y_i^*)$ and compares it with the stored value of A_i in its memory to verify the legality of the user before the smart card accepts the password update request, it is not possible for the adversary to guess out identity ID_i and password P_i correctly at the same time in real polynomial time. Moreover, once the number of login failure exceeds the predefined system value, the smart card will be locked immediately. Therefore, the proposed protocol is secure against denial of service attack.

(9) Online password guessing attack: In this type of attack, the attacker pretends to be a legitimate client and attempts to login to the server by guessing different words as password from a dictionary. In the proposed scheme, the attacker first has to get the valid smart card and then has to guess the identity ID_i and password P_i corresponding to user U_i. It is not possible to guess out identity ID_i and password P_i correctly at the same time in real polynomial time. Therefore, the proposed protocol is secure against online password guessing attack.

(10) **Forward secrecy:** An authentication scheme with forward secrecy assures that even if the server S's long time private key d is leaked out by accident or is stolen by an adversary, it is still impossible for an adversary to obtain the session keys generated before, nor can the adversary launch user/server impersonation attack successfully. In our scheme, the session key SK and login message M_i are generated with the contribution of y_i, which can not be computed without knowing the correct value of the identity ID_i and password P_i corresponding to user U_i, even the attacker knows the server S's long time private key d and has got user U_i's smart card. As a result, our scheme provides the property of forward secrecy.

6 Performance Analysis

We compare the performance and security features among the relevant dynamic ID-based authentication schemes and our proposed scheme in this section. The comparison results are depicted in Table 2 and 3, respectively.

Table 2. Performance comparison among relevant dynamic ID-based schemes

	Our scheme	Sood et al. [16]	Khan et al. [15]	Hu et al. [12]	Horng et al. [13]
Total computation cost	$2T_E+12T_H$	$12T_H$	$10T_H$	$4T_E+4T_S+6T_H$	$7T_E+4T_S+8T_H$
Communication overhead	1408 bits	384 bits	768 bits	3456 bits	2432 bits
Storage cost	2432 bits	384 bits	384 bits *	2816 bits	3328 bits

* It's likely that a parameter was missed out when Khan et al. design the registration phase [17].

Table 3. Security features comparison among relevant dynamic ID-based schemes

	Our scheme	Sood et al. [16]	Khan et al. [15]	Hu et al. [12]	Horng et al. [13]
Preserving user anonymity	Yes	No	No	No	Yes
Resistance to offline password guessing attack	Yes	No	No	No	Yes
Resistance to stolen verifier attack	Yes	No	Yes	Yes	Yes
Resistance to user impersonation attack	Yes	Yes	Yes	No	Yes
Resistance to server masquerading attack	Yes	Yes	Yes	No	Yes
Resistance to replay attack	Yes	Yes	Yes	Yes	Yes
Resistance to parallel session attack	Yes	Yes	Yes	Yes	Yes
Resistance to denial of service attack	Yes	Yes	Yes	Yes	No
Resistance to online password guessing attack	Yes	Yes	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes	Yes
Forward secrecy	Yes	Yes	No	Yes	Yes

An efficient authentication scheme must take computation cost, communication overhead and storage cost into consideration. We mainly focus on the efficiency of login and verification phases since these two phases are the main body of an authentication scheme. Note that the identity ID_i, password P_i, timestamp values and output of secure one-way hash function are all 128-bit long, while n, e and d are all 1024-bit long. Let T_H, T_E, T_S and T_X denote the time complexity for hash function, exponential operation, symmetric cryptographic operation and XOR operation respectively. Since the time complexity of XOR operation is negligible as compared to the other three operations, we do not take T_X into account. Typically, time complexity associated with these operations can be roughly expressed as $T_E > T_S > T_H >> T_X$.

In our scheme, the parameters N_i, A_i, B_i, n and e are stored in the smart card, thus the storage cost is 2432(= 3 * 128+2 * 1024) bits. The communication overhead includes the capacity of transmitting message involved in the authentication scheme, which is 1408 (= 3 * 128+1024) bits. During the login, verification and session key agreement phase, the total computation cost of the user and server is $2T_E+12T_H$. The proposed scheme is more efficient than Hu et al.'s scheme [12] and Horng et al.'s scheme [13], and requires more computation, communication and storage than that of Sood et al.'s scheme [16] and Khan et al.'s scheme [15], but it is highly secure as compared to the related schemes.

Table 3 gives a comparison of the security features of the proposed scheme with the other relevant dynamic ID-based authentication schemes. The proposed scheme provides user anonymity and resists offline password guessing attack, while the latest schemes proposed by [12], [15] and [16] suffer from these attacks. The proposed scheme can withstand denial of service attack, while the scheme presented by [13] is vulnerable to this attack. It is clear that our scheme is more secure as compared to other relevant dynamic ID-based schemes.

7 Conclusion

More recently, Sood et al. showed that Wang et al.'s dynamic ID-based remote user authentication scheme cannot defend against various attacks and then proposed an improved scheme. However, in this paper, we argue that Sood et al.'s scheme fails to preserve user anonymity and is vulnerable to offline password guessing attack and stolen verifier attack under the assumption of non-tamper resistance of the smart card. As to our main contribution, an improved dynamic ID-based authentication scheme was proposed to remedy these security flaws, the security and performance analysis demonstrated that the improved scheme is more secure and practical. In future work, we will give a formal security proof of our proposed scheme.

Acknowledgements. This research was supported by the National Natural Science Foundation of China (NSFC) under Grants No. 61170241 and No. 61073042, and the open program of State Key Laboratory of Networking and Switching Technology under Grant No. SKLNST-2009-1-10. The authors are grateful to the four anonymous reviewers for their valuable suggestions and comments that highly improve the readability and completeness of the paper.

Reference

1. Ku, W.C., Chen, S.M.: Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics 50(1), 204–207 (2004)
2. Chen, Y.C., Yeh, L.Y.: An efficient nonce-based authentication scheme with key agreement. Applied Mathematics and Computation 169(2), 982–994 (2005)
3. Shieh, W.G., Wang, J.M.: Efficient Remote Mutual Authentication and Key Agreement. Computers and Security 25(1), 72–77 (2006)
4. Hsiang, H.C., Shih, W.K.: Weaknesses and Improvements of the Yoon-Ryu-Yoo Remote User Authentication Scheme using Smart Cards. Computer Communications 32(4), 649–652 (2009)
5. Kumar, M.: A new secure remote user authentication scheme with smart cards. International Journal of Network Security 11, 88–93 (2010)
6. Sood, S.K., Sarje, A.K., Singh, K.: Secure Dynamic Identity-Based Remote User Authentication Scheme. In: Janowski, T., Mohanty, H. (eds.) ICDCIT 2010. LNCS, vol. 5966, pp. 224–235. Springer, Heidelberg (2010)
7. Kocher, P., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
8. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Examining Smart-Card Security under the Threat of Power Analysis Attacks. IEEE Transactions on Computers 51(5), 541–552 (2002)
9. Das, M.L., Saxena, A., Gulati, V.P.: A dynamic ID-based remote user authentication scheme. IEEE Transactions on Consumer Electronics 50(2), 629–631 (2004)
10. Chien, H.Y., Chen, C.H.: A remote authentication scheme preserving user anonymity. In: IEEE AINA 2005, pp. 245–248. IEEE Computer Society, Los Alamitos (2005)
11. Wang, Y.Y., Liu, J.Y., Xiao, F.X., Dan, J.: A more efficient and secure dynamic ID-based remote user authentication scheme. Computer Communications 32(4), 583–585 (2009)
12. Hu, L.L., Yang, Y.X., Niu, X.Y.: Improved remote user authentication scheme preserving user anonymity. In: Fifth Annual Conference on Communication Networks and Services Research, pp. 323–328. IEEE Computer Society, Los Alamitos (2007)
13. Horng, W.B., Lee, C.P., Peng, J.: A secure remote authentication scheme preserving user anonymity with non-tamper resistant smart cards. WSEAS Transactions on Information Science and Applications 7(5), 619–628 (2010)
14. Yeh, K.H., Su, C.H., Lo, N.W.: Two robust remote user authentication protocols using smart cards. Journal of Systems and Software 83(12), 2556–2565 (2010)
15. Khan, M.K., Kim, S.K., Alghathbar, K.: Cryptanalysis and security enhancement of a 'more efficient & secure dynamic ID-based remote user authentication scheme'. Computer Communications 34(3), 305–309 (2011)
16. Sood, S.K.: Secure Dynamic Identity-Based Authentication Scheme Using Smart Cards. Information Security Journal: A Global Perspective 20(2), 67–77 (2011)
17. He, D.B., Chen, J.H., Zhang, R.: Weaknesses of a dynamic ID-based remote user authentication scheme. International Journal of Electronic Security and Digital Forensics 3(4), 355–362 (2010)

An Algebra of Social Distance

Hrushikesha Mohanty

Department of Computer and Information Sciences
University of Hyderabad
India
mohanty.hcu@gmail.com

Abstract. This paper continues with the idea of social distance, proposed in [2] and introduces an algebra with two operators for *group social distance* and *lead distance*. It essentially extends the concept of social distance from individual level to group level, formally specifying how a group is socially distanced from an opportunity that intends development. It also proposes a measure to find how one influences others in a group resulting social improvement measured by a metric called *lead distance*. The paper analyzes the effects due to joining of different types of people where types are found based on their social privileges in accessing a social resource. Social difference between two is computed from their social distances, and the difference leads to social entropy of a group. Increase in entropy leads to group fragmentation. It is shown that group fragmentation can be controlled to give rise to desired sub groups. Our aim is to develop a computing methods for social engineering leading towards inclusive society.

Keywords: Social Distance, lead distance, groups and splits -social engineering and social inclusion.

1 Introduction

Human is born to an environment that provides both opportunities as well as constraints for existence.In [3] we have modelled an habitat that models a premise of a person,explains its dynamic behaviour and defines a process of reasoning using the model. Alike human, every entity (natural or man-made) has a premise of its existence. The distance between two premises is termed here as *social distance*. The term distance, is generally used to measure geographical distance. In a seminal work [4] person-to-person communication is categorized in terms of geographical distance and these categories are termed into several types of social distance. A theory on social communication is built on the basis of this concept of social distance. Later, social distance is rather used as a measure between two social vectors that represent their respective social environments. In this paper we have reviewed works on social distance that cover a wide spectrum of research areas ranging from sociology, economics, psychology, legal study to computer science. As of now, computers have more and more roles to play in our

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 153–165, 2012.

daily activities, and so these gadgets need to behave understandably to human needs and sensibilities for making itself socially responsive.

Following this line of reasoning in [2] we have proposed a protocol that is socially responsive by scheduling social resource requests based on social distance. Here, social distance is a measure to quantify the limitation a person possess (for its environment) to access a social resource. In this paper, we have further worked on the concept of social distance and have proposed an algebra with two operators *join*: τ and *lead*: ℓ . Here, a study on these operators has been carried out. It is shown that based on social distance, between a person and a social entity, groupings of different granularity can be made for a particular purpose like accessing a social resource. Study on social dynamics for delivery of governance as well as social welfare programs is of importance. And we wish this piece of work is a small step in that direction.

The paper has seven sections. The second section introduces the concept of social distance. Third section contains the core idea defining the algebra for joining and leading a group. We have defined two operators *join*: τ and *lead*: ℓ and studied their properties. In the fourth section there has been a study on social dynamics in terms of group formation and splits. The questions, like how many under-privileged can be accommodated in a group of a given group distance; and how many privileged and philanthropists should join a group to achieve a desired social uplift, are asked in the framework of an algebra of social distance. The same section also has a study on social fragmentation based on social entropy. Considering, the preference of homogeneity, for joining a group a person selects the group based on its social proximity. Further, we have discussed how does a person select a group to join and on formation of hierarchy of groups of different granularity. In the next section, we describe a framework demonstrating a possible implementation of the concept developed here, for delivery of a social welfare program. Some related works are reviewed in the sixth section. I did not find works exactly related to the work presented here. I have widened scope of my survey to include study on similar concepts in different areas of research ranging from social science to computing science. The paper concludes in the following section identifying further study on this concept.

2 Social Distance

The term *social distance* here characterizes how much a person is constrained to participate in a state program or to avail a state resource. People encounter limitations that arise due to unabridged gap between person and a resource. Social distance between a person u and a resource r is computed by

$$SD_u^r = w_1 * LD_u^r + w_2 * ID_u^r + w_3 * CD_u^r \tag{1}$$

where LD_u^r, ID_u^r and CD_u^r correspond to location distance, invest distance and constraint distance respectively for resource r and user u. More, w_is are the weights assigned to each of the three constituents of the social distance metric. A user is specified by its location, financial capability and facts around it. Below, there is an example of user Varsa located at (Bangalpur, Balasore, Odisha), can invest 1500 rupees. She has five dependants and three lakhs (3L) rupees annual income. The information about $Varsa$ is reported as:

```
User:: Varsa
     {
       Location: (Bangalpur, Balasore, Odisha)
       Invest: 1500
       Fact: (Dependants 5) & (AnnlIncome 3L)
     }
```

Similarly a resource, HBF (House Building Fund) is represented as below:

```
Resource: HBF
     {
         Location: (Welfare Deptartment, Bhubaneshwar,Odisha)
         Cost: 1000 Rs.
         Constraint: AnnlIncomeL4() &  FamilyDepndant4()  &
                                     CompletionCert()
       }
```

The resource HBF is being distributed by Welfare department. Bhubaneshwar, Odisha (gives a geographical location of the resource); the cost to apply for fund is 1000 Rs. The fund is available for those who have less than four lakh rupees annual income i.e. *AnnlIncomeL4()* and the dependants are four or more i.e. *FamilyDepndant4()* . On availing the fund, a user needs to submit a house completion certificate *CompletionCert()*. These are the constraints the resource has.

LD_u^r computes absolute geographical distance between a resource r and a user u. Based on the distance, it is categorized as far, near or at average distance. And each of it assumes value in 10 scale e.g. 10 for *far*, four for *near* and six for *average distance*. Similarly $ID_u^r = r.Cost - u.Invest$. This factor contributes to social distance only if $ID_u^r > 0$. Semantically, $ID_u^r > 0$ is categorized *prohibitive*, *expensive* and *costly* based on slabs defined on invest distance; let's say the three assume values ten, eight and six respectively in a scale of 10. Of course, this categorization has to be person specific for example based on one's economic conditions and liability, the ID slabs are to be fixed. The constraint distance indicates a person's social limitations in availing a resource. A person is disadvantaged if $u.Fact$ fails to satisfy the conditions stated at $r.Constraint$. If no condition is satisfied then CD_u^r is the maximum and zero for vice-versa. The ratio of the unsatisfied condition with respect to the total number of conditions propose a constraint distance. For details, reader may refer to [2].

3 Social Distance Algebra

In this section we present an algebra defining certain operations on social distance. The purpose of such operation is to group people on the basis of social distance. Such a grouping is useful for an agency to deliver a social resource targeting a set of marginalised people defined by their social distances. People registered for a resource make a group. And the algebra is applied on the group to compute both group social distance as wells as lead distance. Also, entropy of the group is computed. Based on entropy, group dynamics including group formation, splitting and hierarchy among groups, are studied.

3.1 Joining a Group

A program like NREGP: National Rural Employment Guarantee Program, adopted in India for mitigating poverty wishes maximal participation of underprivileged people. In order to monitor effective participation, social distance can be considered as a measure. Suppose, the programme has solicited participation of people and in response to the call people join to make a group. As a person p with social distance sd_p joins a group with social distance sd_g, the resultant social distance of the new group with p is defined as:

$$sd_p \ \tau \ sd_g \ = \ (sd_g \ \mid \ sd_g \ = \ max(sd_p, sd_g)) \tag{2}$$

The operator τ returns social distance of the most under-privileged as the social distance of the group. Thus the operator reflects *social inclusiveness.* The other properties of the operator τ when applied to sd_1 and sd_2 are as follows

1. $sd_1 \ \tau \ sd_2 \ = \ sd_2 \ \tau \ sd_1$
2. $(sd_1 \ \tau \ sd_2) \ \tau \ sd_3 = sd_1 \ \tau \ (sd_2 \ \tau \ sd_3)$
3. $(sd_1 \ \tau \ 0) \ = \ sd_1$
4. a. $sd_1 \ \tau \ (-sd_1) \ = \ sd_1$,
   b. $sd_1 \ \tau \ (-sd_i) \ = \ sd_1$

The above first property indicates the social distance of a group is not sensitive to the order people join the group. And the first and the second respectively show *commutative* and *associative* property of the operator τ. The next two rules show the social distance of a group is also not sensitive to additive identity and additive inverse. The joining of a person either with zero or negative social distance does not change the group social distance. Negative social distance is a measure of social privilege, a person may have for its social habitat. However, the idea needs further study for valid calibration. For this paper, the notion of negative social distance is considered for people of privilege who can lead a group positively for resolving some of its social constraints, so that the group attains a new distance called lead distance, that's less than or equal to group's social distance.

3.2 Leading a Group

Self-help has been an affective tool for leading social development programs towards poverty elevation. People in community collaborate to abridge the limitations they have. And, in practice effort is made to elevate people those who are in the most disadvantaged position. This means people at the farthest social distance are to be helped with an intention that, group social distance can be reduced to a distance called *lead distance*. The difference between lead distance and social distance is called *bridge distance of group:* bd_g i.e. the distance that has been covered by leadership of a man on joining the community. A person with social distance sd_i on joining a group with social distance sd_g makes group lead distance ld_g as

$$sd_i \ \ell \ sd_g = (ld_g \ \mid \ ld_g = ld_g + \frac{sd_i - sd_g}{sd_g}); \ sd_g = max(sd_g, sd_i) \tag{3}$$

sd_g is the new social distance of the group. Lead distance computed by ℓ operator, is a measure of a distance obtained for a society being lead by a person for its joining the group. The measure accomplished social development for joining of a person is quantified by the term $(bd_g \ = \ \frac{sd_i - sd_g}{sd_g})$. The distance computation shows different characteristics based on following cases of sd_i and sd_g value combinations. The cases are :

1. $sd_g \ < \ sd_i$
2. $sd_g \ > \ sd_i$
3. $sd_g \ = \ sd_i$
4. $sd_i \ = \ 0$
5. $sd_i \ = \ -sd_g$

Case-1 $sd_g \ < \ sd_i$: A person with social distance sd_i joins a group of social distance sd_g and lead distance ld_g. Then the evolved distances $sd_g^{'}$ and $ld_g^{'}$ are as follows:

$$\begin{aligned} sd_g^{'} &= sd_i \\ ld_g^{'} &= ld_g + \tfrac{sd_i - sd_g}{sd_g} \end{aligned}$$

The case talks of joining of the most under-privileged (than rest of the group) to a group. Though, there is increase in social and lead distances, still the evolved lead distance $ld_g^{'}$ is less than the evolved social distance $sd_g^{'}$. This projects the very nature of community collaboration and self-adjustment leading to improvement by $(sd_i - ld_g^{'})$ - a metric to measure community collaboration.

Case-2: $sd_g \ > \ sd_i$: As a person with social distance sd_i that is less to the group social distance sd_g joins, then it remains unchanged but lead distance decreases by $\frac{sd_i - sd_g}{sd\_g}$. Thus the following relation

$$ld_g^{'} < ld_g < sd_g$$

where ld_g is the lead distance of a group before the person with sd_i joins the group. And $ld_g{'}$ is the evolved lead distance after the person joins. This tells a person with lesser social distance (in comparison to group social distance) does positive contribution to the group enabling some people in overcoming some of their limitations.

Case-3: $sd_g \ = \ sd_i$: In this case, i.e. for sd_i being equal to sd_g, the bridge ratio turns zero thus does not contribute to the computation of group lead distance. And, of-course, the group distance remains the same. Mathematically, it shows the state of helplessness for people with extreme limitations that is equal to sd_g. As all of them are equally disadvantaged, they are not in position to help each other and so the lead distance does not change.

Case-4: $sd_i \ = \ 0$: A person with zero social distance means, the person does not have any limitation in accessing the social resource. As and when such a person joins a group, the social distance of the group remains the same whereas lead distance is reduced by one for bridge ratio assuming -1. Though it's mathematically correct, but misleading for suggesting annihilation of lead distance of a group by joining of ld_g number of people with zero distance, which is a very unrealistic suggestion. In order to rationalize this absurdity we would like to redefine ℓ operator as:

$$sd_i \ \ell \ sd_g = (ld_g \ \mid \ ld_g = ld_g + \frac{sd_i - sd_g}{n \ * \ sd_g}); \ sd_g \ = \ max(sd_g, sd_i) \qquad (4)$$

We have introduced a parameter n in denominator to reduce the bridge ratio so that the mallard of zero distance can be avoided. A simple idea to decide on n is the total number of people present in the group. Having this, we can sensitize bridge ratio to the size of the group stating that the bridge distance is inversely proportional to the size of a group; that is a convincing proposition ($bd_g \propto \frac{1}{n}$).

Case-5: $sd_i = -sd_g$: The negation of social distance indicates one's advantageous position. When a person at advantage with social distance $-sd_g$ joins a group the bridge distance assumes value ($bd_g \ = \ \frac{2}{n}$). This shows the rate of *poverty elevation* is twice now than that of *Case-4*.

From the above cases we have come across the following properties for ℓ operator:

1. $sd_1 \ \ell \ sd_2 \ \neq \ sd_2 \ \ell \ sd_1$
2. $sd_1 \ \ell \ (sd_2 \ \ell \ sd_3) \ \neq \ (sd_1 \ \ell \ sd_2) \ \ell \ sd_3$
3. $sd_1 \ \ell \ (sd_2 \ \tau \ sd_3) \ \neq \ (sd_1 \ \ell \ sd_2) \ \tau \ (sd_1 \ \ell \ sd_3)$

This shows the ℓ operator is neither commutative nor associative; not even distributive. Thus it shows, every lead is different; probably so for uniqueness of a leader!

3.3 Asymmetric Lead Association

The above rules on ℓ operator indicates the lead distance of a group is sensitive to the order of people join the group. Here we will further detail on asymmetry in lead association and deal with its impacts in group formation. In order to appreciate social aspects of lead association we categorize people based on their social distances and study the scenarios that their associations generate. Based on social distances we categorize people to two sections i.e. *LoG*: Lesser of God with farther social distances and *BoG*(Better of God) for people at lesser social distance. One can think of choosing of threshold of social distance, say sd^t to categorize people and this categorization depends on social habitat, the subjects reside on. The cases we list here use these two terms to paint understandably social scenario in formation of a group. The scenarios are :

1. *LoG* earlier (under-privileged join in priority)
2. *BoG* earlier (privileged join in priority)
3. *LoG* and *BoG* join in random
4. *BoG* biased (random but with certain bias for privileged to join)
5. *LoG* biased (random but with certain bias for under-privileged to join)
6. *BoG* followed by *LoG* ($BoG - LoG$) (First privileged ones join followed by under privileged)
7. *LoG* followed by *BoG* ($LoG - BoG$) (First under-privileged form a group and then they allow to privileged ones to join)

Scenario 1: LoG earlier, allows people to join such that their social distances are in increasing order. Say the social distances of those n people make the order $sd_1 > sd_2 > ... > sd_i > ... > sd_n$. Then the association formation also follows an increasing order in lead distances as: $ld_1 > ld_2 > ... > ld_i > ... > ld_n$, $sd_i > ld_i$, and i gives an index of joining a group. This means though the leadership of associating people reduces social distance to a lead distance still the group effectively becomes more and more under-privileged for more and more constrained people join the group. However, when lately less under-privileged keep joining, the group inches forward to better status.

Scenario 2: BoG earlier,unlike Scenario 1 allows the more privileged early to join i.e the order of social distances of the people joining is $sd_1 < sd_2 < ... < sd_i < ... < sd_n$. and that of lead distances is $ld_1 < ld_2 < ... < ld_i < ... < ld_n$. This shows, a privileged society is in process of taking responsibility of under-privileged at the risk of its lowering societal comforts.

Scenario 3,4,5: The natural order in group formation is random that is without enforcing an order as described in the first two cases. This will not exhibit any definite patterns in change of lead distance and social distance of the group as seen before. The bias to one (either privileged or under-privileged) can be made to tune the changes of lead and social distances during group formation.

Scenario 6: In this case, the order of social distances of the people joining a group is $sd_1 < sd_2 < ... < sd_i; sd_{i+1} > sd_{i+2} > ... > sd_n$. A social distance sd^t divides two stages of group formation such that $sd_i < sd^t < sd_n$ and a boundary of two stages demarcated by ';'. The first stage is alike to the second scenario followed by the group formation as done in the first scenario.

Scenario 7: In case of the seventh scenario, the order of group formation is exactly opposite. The usability of analysing group formation in terms of social distance helps observers (e.g. sociologists or program implementation agencies) to monitor group formation and assess its inclusiveness. For example, say in case of the seventh scenario, allowing *LoG*s to join in group, so that a desired ld_g is obtained. And then *BoG*s are invited to join the group so that the group is led (i.e the distance is reduced) to desired level.

This pattern matches with performance of self-help-group made by people of certain social distances and then being equally helped by people of a philanthropic group. A group of people of certain social bracket (a range of social distance) form a group and then the group invites participation of (could be investment or donation) from privileged people to reach a targeted lead distance. This kind of group formation ushers a *controlled social engineering.*This idea leads to investigate the affect of social distances in group formation, that we study in the next section.

4 Group Engineering

In civil society, sometimes it is necessary to drive a community to form pressure groups for achieving objectives. Heterogeneity of a group leads to its split while homogeneous people form a group. Social engineering including both grouping and splitting needs to be tailor made for achieving desired inclusiveness specified in terms of lead distances. Here, group formation and then splitting are studied.

Group formation: Suppose, for a given group with n number of people,social distance sd_g and lead distance ld_g we need to find inclusiveness towards a desired lead distance $ld_g^{'}$. The types of inclusiveness we desire to study are reflected by these questions: How many *LoG*s / *BoG*s / MoGs can join a group? MoGs i.e. *Missionary of God* are kind of persons with social distance $\leq\ -sd_g$. These are the philanthropists who can help a group in overcoming its constraints. In section 3.2 we have seen the role of bridge ratio bd_g in group formation. It either adds to or reduces a group lead distance. The join of *LoG*s increments the distance whereas that of *BoG*s decreases. But, the join of MoGs to a group helps to reduce its lead distance at fast rate. Say, the joining of m number of people makes desired lead distance $ld_g^{'}$, then.

$$ld_g^{'} = \sum_{i=1}^{m}(ld_g^{i-1} + \frac{sd_i - sd_g}{sd_g}); sd_g \ = \ max(sd_g, sd_i), ld_g^0 = ld_g, \qquad (5)$$

sd_i is the social distance of the i^{th} person. For all the above three cases one can simulate group formation and get an idea of the number of people required to join a group to achieve the desired group lead distance.

On Splits: A group may need to split on certain criteria, in order to make it manageable for the purpose it is created. For example, for a poverty elevation program the population of a region may be grouped on the basis of group size or social distance or social difference ratio. We have favoured the last one as it reflects the lack of homogeneity in a group with respect to its mass (size). Before dealing with group split, we will discuss on a concept, group *social entropy* se_g a measure to count social disorder with respect to group size of m. Thus $se_g = \sum_{i=0}^{m} \frac{sf_i}{m}$. One can compute social difference of an entrant with respect to the latest entrant or the best of *BoG*s or the average social distance of the group. While people join for an objective, the group formation being guided by social entropy, form smaller sub-groups where each is assigned to a desired range of social entropy. Based on this idea, below we present an algorithm-*joining a group*. A group can be ordered in a hierarchy of sub-groups of different granularity at different levels; where granularity at the bottom level is the lowest and the vice-versa at the root level. As the idea is very simple, for space limit, we will not detail it.

Algorithm 1. Alg.1 Joining a Group

```
Add_Person($sd_i, S$)
// $sd_i$- social distance of person $i$, $S$ - set of sub-groups.
begin
1. Choose a sub-group that is at the most proximity of $sd_i$. In case there are more such, then choose one with the least size.
2. If the chosen group is in-growth i.e has not reached its entropy threshold
then join, update social entropy of the sub-group and Terminate.
3. else (sub-group is at-growth i.e. has reached its entropy threshold) find the person $sd'$ contributing the most to social entropy.
4. if ($Sd_i < sd'$) then remove $sd'$; $sd_i$ joins the subgroup, Add_Person($sd', S$) else Add_Person($sd_i, S$)
5. if $sd_i$ fails to find a subgroup to join then new group is made with $sd_i$.
end.
```

5 A Framework

Here, we take up a hypothetical example to demonstrate the possible implementation of a system that uses the proposed algebra. Suppose, a government has proposed a scheme - Small Scale Business Promotion (SSBP) with a preference to under-privileged strata of a society. A person P is considered under-privileged based on its distance from SSBP (i.e. eligibility criteria SSBP has for its client P). A resource directory ResD keeps a list of such schemes offered to citizens. And there is an entity, ResM, a Resource Manager that monitors social dynamics e.g.fixing social distance ranges for sub-grouping, finding the number of LoGs, MoGs or BoGs inviting to join a group. In a digital society P, R (Resource-SSBP here is an instantiation of R) ResD and ResM are connected on Internet as shown in Fig.1. The relation among entities are depicted in Fig.2.

An Internet based resource management system as here, is used for resource registration, querying a resource, requesting a resource, group formation and management.

First, a resource is required to register at ResD. In the arena of governance one can think of existence of several such directories on Internet, each associated to a geographical area. A person wishing to avail a resource; for example needing to take part in a SSBP scheme, can make a query on its ResD to avail the information. Information dissemination can follow either push or pull mode. In

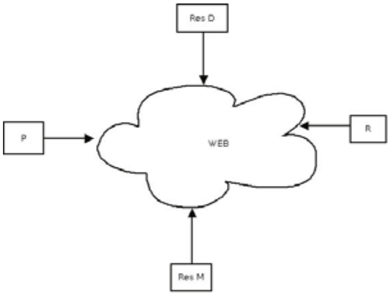


Fig. 1. Person and Resources on Web

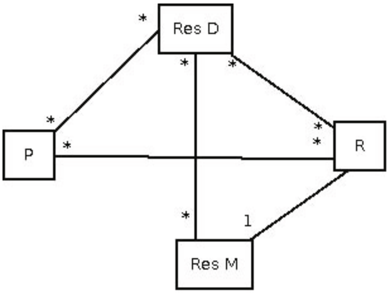


Fig. 2. Class diagrams: P, R, ResD and ResM

case of push mode, people wishing to avail such information needs to register at its ResD, so that on publication of a resource it can pass the information to a person on its cell. The person now can directly interact (for service) with the resource. And as stated in [2], the request can be scheduled as per its social distance. The resource R, may group the clients (say in case of SSBP, citizens) in groups for administrative reason like targeting to serve people of certain social status that is defined by social distance of a group. A resource manager ResM periodically gets group information and reason over it for social engineering. The repositories ResourceList, GroupsList (refer to XML schema) provide description of resource and group used in the proposed resource management system. A collaboration diagram visualizing interactions required for basic purposes like resource and citizen registrations at ResD, group management by ResM and citizen resource service session initiation are presented in Fig.3. In Table 1, all interaction messages are listed and the purpose of the interactions are also stated. This provides a blue print of possible implementation of the proposed system.

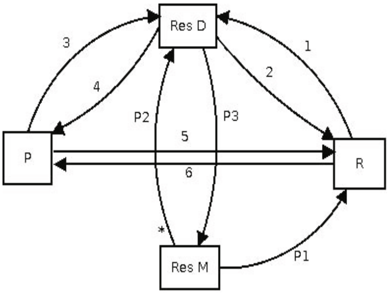


Fig. 3. Interactions among P,R,ResD and ResM

```
<!-- List of all Resources -->    <!List groups >

<ResourceList>                    <GroupsList>
<resource  name="String">         <group name="string1"id="string2">
```

```
 <location>
 <DoorNo></DoorNo>
 <Street></Street>
 <City></City>
 <State></State>
 </location>
<cost>amt</cost>
<constraint> string
</constraint>
<WorkingGroups>
<groupID>String</GroupID>
</WorkingGroups>
</resource>
</ResourceList>
```

```
<!List group members>
<groupMembers>
<groupMemberInfo id="String/null">
<client name="string">
<location>
<DoorNo></DoorNo>
 <Street></Street>
  <City></City>
  <State></State>
  </location>
  <invest>amt</invest>
  <fact>String</fact>
 </client>
   </groupMemberInfo>
   </groupMembers>
    <groupDistance>"number"
    </groupDistance>
    <groupEntropy>"RealNumber"
   </groupEntropy>
    <parentGroupID>ID
    </parentGroupID>
    <sibblingGroupID>ID
    </sibblingGroupID>
    <nhbrGroupID>ID
      </nhbrGroup>
  </group>
 </GroupsList>
```

Table 1. Collaboration Messages

IntId	IntMessage	Purpose
1	RegisterRes	Resource registration
2	AckRegister	Acknowledging resource registration
3	EnquireRes	Enquiring a resource
4	EnquiryRest	Enquiry result
5	ServReq	Client request for resource service
6	InitServReq	Initiation of service request
p1	GetGroup	Resource manager polls group information
p2	GroupEngg	Group Engineering intervention
p3	GroupEnggReq	Requests for group engineering

6 Related Work

A way back, in 1959 Edward T Hall while describing people's world view, outlined how culture control people's lives and make one different from another. He introduced proxemics, use of space in his seminal works *The Silent Language* [4] and proposed several distances viz. Intimate Distance, Personal Distance, Public Distance and Social Distance to categorize social communication. He said, social distance is the distance for impersonal business that shows the degree on involvement and formality. Further, the notion of distance is being used to analyze social phenomena. A social entity is represented by several social dimensions and the distance between two provides a measure of social distance. But, the point is, social distance is a metric unlike to physical distance, as the former deals with points at different types of spaces. So, the usual Euclidean metric is not applicable. So, the paper [1] prophesies that social distance is not a mathematical metric; and if exists at all is a pseudo metric. It proposes a pseudo-metric on multi-dimensional space R^n that embeds a set of points, where each is a $n-tuple$ of real numbers giving measures of characteristics of social entities the space embeds in it. The pseudo metric has two constraints: ordinal invariance and non-constancy. **a.** $d(x,y) < d(x,z)$ iff $d[T(x),T(y)] < d[T(x),T(z)]$ **b.** $d(x,y) \neq d(x,z)$ where z is in open neighbourhood of y, $d()$ returns a distance measure between $x, y \in R^n$ and $T()$ is a transform function on $x \in R^n$. The paper considers the issue of transformation of measures of an attribute taken in different scales. It is sceptic of defining social distance metric for natural complexity of defining measure of societal entities. For example, there could be zero distance between two people with distinct socio-economic vectors or two countries with distinct development vectors on some issues (but not so for another).

In a paper [7] on coalition, interpersonal relationship is used to define interpersonal distance, that is termed here as social distance. Relationship, intimacy, frequency of interactions, reciprocity and similar attributes are considered to measure interpersonal relationship. The stronger a tie the lesser is its social distance. The paper follows small-world notion introduced in [8] to categorize interpersonal relationship depending on social distance. It investigates the usages of long-range connections [5] in finding a target in a social network.

Here in this work, we define distance as quantification of difficulty one faces to access service of a social resource. A person of a social habitat [3] is endowed with certain enabling factors. A social resource requires its clients to satisfy constraints associated (to the resource). The degree of inability of the entity to satisfy the constraints gives a measure of its social distance with the respect to the resource. In [2] we have shown the use of social distance in routing social resource request. This work is in continuation to our work on social distance and its usages in social engineering.

7 Conclusion

An entity in society differs to another for its unique existence and the difference between their habitats makes a social distance. Social distance generates social dynamics - huge social phenomena that are to be understood for the prosperity of tomorrow. The study made here is a small step in that direction.

We have defined here an algebra of social distance and demonstrated its usage in engineering of social groups. An algebra of two operators *join* (τ) and *lead* (ℓ) is defined to calculate social distance as well as lead distance of a group. The usage of this algebra is demonstrated in reduction of social distance by participation of socially privileged ones.

Role of social distance in social fragmentation as well as social amalgamation is also studied. A framework for Internet based system for creation of social distance based e-society is presented showing the possible implementation of the idea.

Determining social distance of a person in a society is complex and requires a study to devise a generic framework to compute it for a purpose, in making of an e-society. This requires a formalisation of description of a social entity, its environment, deduction of its constraints (to achieve a goal).

In this paper, we have taken a gross view of social distance at macro-level. Further, it requires a micro-level study to understand what structure a group holds with individuals, to project a gross view on social distance. Understanding such a structure will be of immense usages for targeting a group for reduction of its social distance. In other-words, a program can be manufactured for a structure such that a desired result is possible. Likewise, a further study on lead distance needs to be carried out to model collaboration in a group leading to generate desirable leading distance. Further, study can be both at macro and micro levels for identification of a group leader. Above all, there is a pressing need to investigate on utility of the concept with experiments at different social environments.

Acknowledgement. This work was carried out during my sabbatical in the academic year 2010-11. I'd like to thank reviewers for their insightful comments.

References

1. Osborne, D.K.: Social Distance. Quality and Quantity 9, 339–348 (1975)
2. Mohanty, H.: Socially Responsive Resource Usage: A Protocol. In: Natarajan, R., Ojo, A. (eds.) ICDCIT 2011. LNCS, vol. 6536, pp. 243–254. Springer, Heidelberg (2011)
3. Mohanty, H.: Person in Habitat and its Actions: A Model. Accepted in IEEE Conference Indicon 2011, Hyderabad, India (2011)
4. Hall, E.T.: The Silent Language. Anchor Books, NY (1990)
5. Granovetter, M.S.: The strength of weak ties. American Journal of Sociology 6, 1360–1380 (1973)
6. Dutta, B., Mutuswami, S.: Stable networks. Journal of Economic Theory 76, 322–344 (1997)
7. Inaltekin, H., Chiang, M., Vincent Poor, H.: Average Message Delivery Time for Small-World Networks in the Continuum Limit. IEEE Transactions on Information Theory 56(9), 4447–4470 (2010)
8. Watts, D., Strogatz, S.H.: Collective dynamics of small-world networks. Nature 393(6684), 440–442 (1998)

Detecting Flaws in Dynamic Hierarchical Key Management Schemes Using Specification Animation

Anil Mundra, Anish Mathuria, and Manik Lal Das

DA-IICT Gandhinagar, India
{anil_mundra,anish_mathuria,maniklal_das}@daiict.ac.in

Abstract. In key assignment schemes for hierarchical access control systems, each access class has a key associated with it that can be used to derive the keys associated with every descendant of that class. Many recently proposed key assignment schemes support updates to the hierarchy such as addition and deletion of classes and class relationships. The dynamic changes entail a change to the hierarchy as well as re-computing of public and secret information. In this paper, we describe a software tool that supports the animation of specifications of dynamic schemes. The specification of a scheme, written in Prolog, corresponds to a symbolic model of the algorithms used by the scheme for key generation and for handling dynamic changes. The tool allows us to generate a test hierarchy, generate keys for the classes in the hierarchy, and simulate various dynamic operations. The animation search using the tool has shown to be useful in finding previously unreported attacks on several existing dynamic schemes.

1 Introduction

In hierarchical access control systems, the higher privileged users are entitled to have access to the information held by lower privileged users; the latter are not entitled to have access to the information held by higher privileged users. A hierarchical key assignment scheme is a cryptographic mechanism for enforcing access control in hierarchies. It provides a method for assigning encryption keys and optional private information to each class in the hierarchy in such a way that it is feasible for a class (also known as node) to derive the keys of its descendants in the hierarchy whereas it is infeasible to derive the the keys of its ancestors. The best known example of such a scheme is the Akl-Taylor scheme [1] introduced in 1983.

A dynamic key assignment scheme supports changes to the hierarchy such as addition and deletion of classes and class relationships. Many proposals for dynamic key assignment schemes [2,3,4,5] have been published in the literature. The algorithms for handling dynamic changes introduce additional challenges to the security of such schemes. Indeed, several dynamic schemes have been shown to suffer from the so-called *ex-member problem*, cf. [6,7,8]. In some schemes a

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 166–176, 2012.

user who is removed from a class and thus unauthorized to access that class is still able to obtain the new class key.

The work of Atallah, Frikken, and Blanton [9] provides a formal framework for defining and proving the security of hierarchical key assignment schemes. However, a proof of security in their framework is limited to static hierarchies; it does not address security of dynamic schemes. A number of formal methods with tool support have been proposed for analysis of key management protocols. The Interrogator [10] tool of Millen, Clark and Freedman is a Prolog program that searches for security vulnerabilities in protocols. Longley and Rigby [11] proposed a rule-based approach to search for potential attacks on protocols. The ProVerif [12] tool, developed by Blanchet, is an automatic cryptographic protocol verifier based on a representation of the protocol and the attacker by Prolog rules.

To the best of our knowledge, none of the previous protocol analysis tools can be directly applied to the security analysis of dynamic schemes. We use an alternative approach for analysis of dynamic schemes: specification animation. Our approach is motivated by Boyd and Kearney [13] who showed that specification animation provides a useful technique for finding flaws in fair exchange protocols.

In this paper, we describe a Prolog-based tool for analyzing dynamic key assignment schemes using specification animation. Using our tool we were able to find attacks on several published schemes. We have used the tool to rediscover all the known attacks on the scheme proposed by Yang and Li [8]. We also found previously unreported flaws in the schemes proposed by Tang [14], Chen and Huang [15], and He and Li [16].

The paper is organized as follows. In Section 2 we introduce some notations and definitions related to key assignment schemes. Section 3 describes the schemes we analyzed using our tool. In Section 4 we describe our approach to specifying key assignment schemes in Prolog. Section 5 contains descriptions of attacks against the schemes described earlier. Section 6 concludes the paper.

2 Preliminaries

A partially ordered set is a pair $(L, \leq)$, where $\leq$ is a reflexive, anti-symmetric, transitive binary relation on a set L. We say y is the immediate successor of x (and x is the immediate predecessor of y), denoted $y \leq x$, if $y \leq x$ and there does not exist $z \in L$ such that $y \leq z \leq x$. If there exist such z then y is said to be successor of x (and x is the predecessor of y). A node with no predecessor is called a *root* node. The notations $Desc(x)$ and $Anc(x)$, where x is a node in hierarchy, represents sets of successors and predecessors of x (including x) respectively. Similarly, $IPre(x)$ and $ISuc(x)$, represent sets of immediate predecessors and immediate successors of x (excluding x) respectively.

Following Crampton, Martin and Wild [17], a key assignment scheme for an information flow policy $(L, \leq)$ defines two algorithms:

- *Key generation algorithm* returns a labeled set of encryption keys $(k(x) : x \in L)$ and secret values $(\sigma(x) : x \in L)$, which we denote by $k(L)$ and $\sigma(L)$ respectively and some public data, *Pub*;
- *Key derivation algorithm* takes $x, y \in L, \sigma(x)$ and *Pub*, and returns $k(y)$ whenever $y \leq x$.

Existing schemes can be classified into different categories depending on the type of key derivation and key selection [17]. An *indirect* scheme is based on the idea of iterative key derivation, that is, to derive the key of its descendant v, a node u has to derive the key of each class between u and v. A scheme has *dependent keys* if the keys cannot be chosen independently.

2.1 Security Definition

Let $M \subseteq L$ and let $\sigma(M)$ denote $(\sigma(x) : x \in M)$. We say that a key assignment scheme S is collusion secure with respect to M (or M-secure) if for all $y \in L$, it is feasible to derive $k(y)$ knowing $\sigma(M)$ and Pub only if $y \leq x$ for some $x \in M$. We say S is node secure if it is x - secure for all $x \in L$, and collusion secure if it is M-secure for all $M \subseteq L$.

2.2 Dynamic Access Control

Usually in large organizations, the access hierarchy evolves with time. Changes to the hierarchy can be divided into two types:

User Revocation. This refers to the addition or deletion of users from classes. It does not modify the structure of hierarchy.

Structural Modification. This refers to the addition or deletion of nodes and relationships between the nodes.

Changes to the hierarchy require rekeying of nodes. A major design goal of dynamic schemes is to accomplish such changes without performing wholesale rekying of the hierarchy.

3 Descriptions of Three Schemes

3.1 Tang Scheme

This is an iterative scheme with dependent keys.

Key Generation

1. For each node $v_i \in V$, pick a randomly chosen public label P_i.
2. For each root node pick a random value as the key associated with that node.
3. To compute the key K_j of node v_j, do the following:
   (1) If v_j has a single parent v_i whose key is K_i, then set $K_j = H(K_i \oplus P_j)$.
   (2) If v_j has multiple parents $v_1, v_2, \ldots, v_n$ whose keys are $K_1, K_2, \ldots, K_n$, then set $K_j = H(H(K_1 \oplus P_j) \oplus H(K_2 \oplus P_j) \oplus \cdots \oplus H(K_n \oplus P_j))$. The values $H(K_1 \oplus P_j)$, $H(K_2 \oplus P_j)$, $\ldots$, $H(K_n \oplus P_j)$ are kept confidential to the nodes $v_1, v_2, \ldots, v_n$.

Insertion of an edge: Suppose the edge (v_i, v_j) is to be inserted. Then we re-compute the key of each node $v_h \in Desc(v_j)$.

Insertion of a new node: Suppose a new node v_i is to be inserted with optional edges coming into and out of v_i. Then we obtain the key K_i of v_i as follows. If v_i has no incoming edges, then set k_i to a newly generated random value, else we compute K_i from the keys of parents of v_i. Finally, we re-compute the key of each $v_h \in Desc(v_i) \setminus \{v_i\}$.

Deletion of an edge: Suppose the edge (v_i, v_j) is to be deleted. Then we obtain the new key K_j of v_j as follows. If v_j becomes a root node after edge removal, then set k_j to a newly generated random value, else we re-compute K_j from the keys of remaining parents of v_j. Finally, we re-compute the key of each $v_h \in Desc(v_j) \setminus \{v_j\}$.

Deletion of a node: Suppose a node v_i is to be deleted. We perform the following steps:

1. Add an edge (v_j, v_k) for each $v_j \in IPred(v_i)$ and $v_k \in ISuc(v_i)$.
2. Delete all edges coming into and out of v_i.
3. For each node v_k for which an incoming edge was added in the first step, re-compute the key of each $v_h \in Desc(v_k)$.

Figure 1 shows an example hierarchy. The key assignment for the hierarchy under Tang's scheme and two other schemes is given in Table 1.

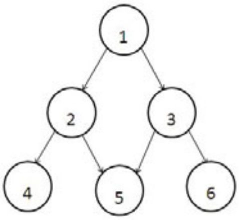


Fig. 1. An example hierarchy

3.2 He-Li Scheme

This is an iterative scheme with independent keys.

Key Generation

1. For each node $v_i \in V$, pick a randomly chosen key K_i.
2. For each edge (v_i, v_j), compute public information $y_{i,j} = K_j \oplus H(K_i)$, where K_i and K_j are the keys of nodes v_i and v_j, respectively.

Table 1. Key Assignments for Reviewed Schemes

Node Id	Label	Shaohua Tang		Chen-Huang		He-Li	
		Key	Secret Info	Key	Edge Info	Key	Edge Info
1	P_1	K_1	-	K_1	$E_{H(P_2 \oplus K_1)}(K_2)$ $E_{H(P_3 \oplus K_1)}(K_3)$ $E_{H(P_4 \oplus K_1)}(K_4)$ $E_{H(P_5 \oplus K_1)}(K_5)$ $E_{H(P_6 \oplus K_1)}(K_6)$	K_1	$K_2 \oplus H(K_1)$ $K_3 \oplus H(K_1)$
2	P_2	$H(K_1 \oplus P_2)$	$H(K_3 \oplus P_5)$	K_2	$E_{H(P_4 \oplus K_2)}(K_4)$ $E_{H(P_5 \oplus K_2)}(K_5)$	K_2	$K_4 \oplus H(K_2)$ $K_5 \oplus H(K_2)$
3	P_3	$H(K_1 \oplus P_3)$	$H(K_2 \oplus P_5)$	K_3	$E_{H(P_5 \oplus K_3)}(K_5)$ $E_{H(P_6 \oplus K_3)}(K_6)$	K_3	$K_5 \oplus H(K_3)$ $K_6 \oplus H(K_3)$
4	P_4	$H(K_2 \oplus P_4)$	–	K_4	-	K_4	-
5	P_5	$H(H(K_2 \oplus P_5)$ $\oplus H(K_3 \oplus P_5))$	–	K_5	-	K_5	-
6	P_6	$H(K_3 \oplus P_6)$	–	K_6	-	K_6	-

Insertion of a new node: Suppose a new node v_i is to be inserted with optional edges coming into and out of v_i. Then we assign a new random key K_i to node v_i and compute public information corresponding to all edges coming into and out of v_i.

Deletion of a node: Suppose a node v_i is to be deleted. We perform the following steps:

1. Add an edge (v_j, v_k) for each $v_j \in \mathit{IPred}(v_i)$ and $v_k \in \mathit{ISuc}(v_i)$.
2. Delete all edges coming into and out of v_i.
3. Now compute public information $y_{j,k}$'s corresponding to all edges that were added in the first step, and remove public information corresponding to all edges that were deleted in the second step.

The above scheme is a variant of the scheme of Atallah *et al.*

3.3 Chen-Huang Scheme

This is a direct scheme with independent keys.

Key Generation: For each node $v_i \in V$, we perform the following steps:

1. Pick a randomly chosen key K_i and a public label P_i.
2. For each descendant $v_j \in \mathit{Desc}(v_i)$, we compute public information $R_{ij} = E_{H(P_j \oplus K_i)}(K_j)$, where P_j and K_j are the public label and secret key associated with the descendant v_j of v_i.

Insertion of an edge: Suppose the edge (v_i, v_j) is to be inserted.

1. Add the edge (v_i, v_j).
2. For each $v_a \in \mathit{Anc}(v_i)$ and $v_b \in \mathit{Desc}(v_j)$, if the corresponding public information R_{ab} does not exist, then compute $R_{ab} = E_{H(P_b \oplus K_a)}(K_b)$.

Insertion of a new node: Suppose a new node v_i is to be inserted with optional edges coming into and out of v_i.

1. Pick a randomly chosen key K_i and a public label P_i.
2. Add the edges one by one, using the above procedure for edge-insertions.

Deletion of an edge: Suppose the edge (v_i, v_j) is to be deleted.

1. Remove the edge (v_i, v_j) and the corresponding public information R_{ij}.
2. For each $v_a \in Anc(v_i)$ such that $v_a \notin Anc(v_j)$, remove the public value R_{ab} for each $v_b \in Desc(v_j)$.

Deletion of a node: Suppose a node v_i is to be deleted. We perform the following steps:

1. Delete all edges coming into and out of v_i, using the above procedure for edge-deletions.
2. Remove v_i.

4 Prolog-Based Modeling

We associate an integer constant to each node when creating the hierarchy. The graph corresponding to a given hierarchy is represented by a list of vertices and a set of edge predicates. For example, the graph in Figure 1 is represented using the following clauses.

```
vertex([1,2,3,4,5,6]).
edge(1,2).
edge(1,3).
...
```

We represent randomly generated secret keys and public labels by disjoint sets of constants. The constants representing keys and labels are stored in two separate lists. Once a constant representing a key or label is allocated to a node, it is removed from the respective list. The labels assigned to various nodes are stored in `node_pub/2` clauses, which have the form `node_pub(NodeId, Label)`.

```
node_pub(1, p1).
node_pub(2, p2).
...
```

In a scheme with independent keys such as the He-Li scheme, we initially extract one key from the initial key list for every node in the hierarchy. In the case of a scheme with dependent keys, we extract one key from the initial key list for every root node in the hierarchy. For such schemes, the derived keys are represented using compound terms. Suppose we assign the key `k1` to the root node of example hierarchy. In the case of Tang's scheme, we represent the key of the left child of root by the term `hash(xored(k1,p2))`. The keys assigned to various nodes are stored in `key/2` clauses, which have the form `key(NodeId, Key)`.

```
key(1, k1).
key(2, hash(xored(k1,p2))).
...
```

In a scheme with dependent keys, the key of each non-root node is derived from the keys of its parents. The following code fragment is used for generating the key of a node with a single parent:

```
key(Parent,Parent_Key),
node_pub(Child, Child_label),
append([Parent_Key],[Child_label],Int_Key),
Fact_val=..[xored | Int_Key],
Key = hash(Fact_val).
```

The following predicate is used for generating the key of a node with multiple parents.

```
create_key(_,[],List,List).
create_key(Child,[Parent|PList],Temp_List,Key_List) :-
          key(Parent,Parent_Key),
          node_pub(Child, Child_label),
          append([Parent_Key],[Child_label],Keys),
          Fact_val=..[xored | Keys],
          NFact = hash(Fact_val),
          list_iparent(Child, Parent_list),
          delete(Parent_list, Parent, Peer_list),
          distr_sec(Child,Peer_list,NFact),
          Temp = [NFact | Temp_list],
          create_key(Child,PList,Temp,Key_list).
```

When a child node has multiple parents then each parent node requires relevant secret information to derive the child's key. The additional secret information is generated using the distr_sec predicate and stored in secret/3 clauses, which have the form secret(Parent, Child, SecretInfo). The following two clauses are generated for the example hierarchy.

```
secret(2,5,hash(xored(hash(xored(k1,p3)),p5))).
secret(3,5,hash(xored(hash(xored(k1,p2)),p5))).
```

In some schemes public information is associated with each edge in the graph. The public edge information is stored in edge_pub/3 clauses, which have the form edge_pub(Parent, Child, PubInfo).

```
edge_pub(1,2,xored(k2,hash(k1)).
edge_pub(1,3,xored(k3,hash(k1)).
...
```

The information known to the adversary includes labels associated with nodes and public edge information. We store such information in known/1 clauses, which have the form `known(Info)`. Each dynamic operation modifies the `edge/2` clauses and vertex list representing the hierarchy. If a dynamic operation requires rekeying then the operation also modifes the `key/2` clauses. The revoked public and secret information is added to the database of `known/1` clauses.

The following predicate checks whether a particular term is known to the adversary.

```
compute_key(Dst_key)   :-
            known(Dst_key),
            write(Dst_key),nl.
compute_key(hash(X))   :-
            compute_key(X).
compute_key(hash(X,Y)) :-
            compute_key(X),
            compute_key(Y).
compute_key(xored(X,Y)):-
            compute_key(X),
            compute_key(Y).
compute_key(X)         :-
            (known(xored(X,Y));
            known(xored(Y,X))),
            compute_key(Y).
compute_key(X)         :-
            known(enc(Hash_val,K)),
            compute_key(Hash_val),
            X == K.
compute_key(Src_node,Dst_node)  :-
            key(Src_node,Src_key),
            key(Dst_node,Dst_key),
            asserta(known(Src_key)),
            compute(Dst_key).
```

The next predicate *attack_simulate* simulates a collusion attack. It takes a challenge node as input. The predicate *list_unauth* returns a list of all unauthorized nodes with respect to the challenge node. The predicate *corrupt_keys* stores the keys of the unauthorized nodes in `known/1` clauses.

```
attack_simulate(Challenge_node) :-
      list_unauth(Challenge_node,Unauth_list),
      corrupt_keys(Unauth_list),
      key(Challenge_node,Ch_key),
      compute_key(Ch_key).
```

5 Attack Simulation

Using the tool, we generated three example hierarchies to demonstrate attacks against the schemes describe earlier.

Tang Scheme

Consider the hierarchy shown in Figure 2.

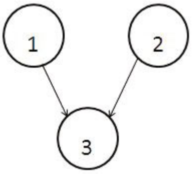


Fig. 2. An example hierarchy for Tang scheme

Here $k1, k2$ and $hash(xored(hash(xored(k1, p3)), hash(xored(k2, p3))))$ are the secret keys assigned to the nodes $1, 2$ and 3 respectively. The nodes 1 and 2 are additionally assigned the secrets $hash(xored(k2, p3))$ and $hash(xored(k1, p3))$, respectively.

Suppose we delete node 2.

```
?-delete_node(2).
  True
?-attack_simulate(3).
  hash(xored(k1,p3)).
  True
```

After node 2 is deleted, the secret information held previously by node 2 is added to the information known to the adversary. The information added includes the value $hash(k1 \oplus p3)$, which is equal to the new key assigned to node 3.

He-Li Scheme

Consider the hierarchy shown in Figure 3. Here $k1, k2$ and $k3$ are the secret keys assigned to the nodes $1, 2$ and 3 respectively. The public information corresponding to the edges $(1, 2)$ and $(1, 3)$ consists of $k2 \oplus hash(k1)$ and $k3 \oplus hash(k1)$.

In the hierarchy under consideration, node 2 is not entitled to access node 3. Similarly, node 3 is not entitled to access node 2. However, the following queries show that the above requirements are not met.

```
?- compute_key(2,3).
   True.
?- compute_key(3,2).
   True.
```

The following calculations show why the first query returns $True$. First, node 2 xors its key $(k2)$ with the public information corresponding to edge $(1, 2)$ to obtain $hash(k1)$. Now node 2 xors this value with the public edge information corresponding to edge $(1, 3)$ to obtain $k3$.

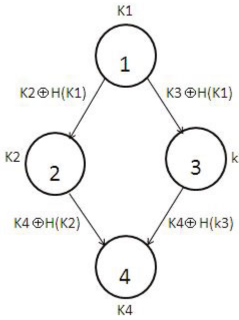


Fig. 3. An example hierarchy for He-Li scheme

Chen-Huang Scheme

Consider the hierarchy shown in Figure 4. The public information corresponding to the edges $(3, 5)$ and $(3, 6)$ is stored in the following clauses.

```
edge_pub(3,5,enc(hash(xored(k3,p5)),k5)).
edge_pub(3,6,enc(hash(xored(k3,p6)),k6)).
```

Suppose we delete node 3. The resulting hierarchy is shown in Figure 5.

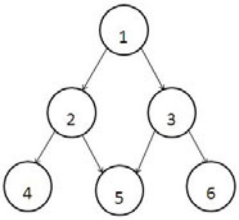


Fig. 4. Original hierarchy for Chen and Huang scheme

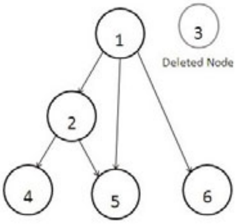


Fig. 5. After deletion of intermediate node in Chen and Huang scheme

```
?- delete_node(3).
   True.
?- attack_simulate(5).
   k3
   True.
```

After node 3 is deleted, it can still derive the key of node 5 using the old public information. Lack of rekeying makes this scheme insecure.

6 Conclusions

We modeled three existing dynamic key assignment schemes using Prolog. Using the tool we have developed, we were able to simulate attacks on all three schemes.

Our approach has several limitations. The most important limitation is that attacks may be found only if the sequence of operations completed by the user lead to an insecure state. In future work, we plan to develop techniques to automate the search for insecure states.

Acknowledgments. This work was supported by a grant from DST (Govt. of India) under the India-Japan co-operative programme. We would also like to thank the three anonymous referees for helpful comments.

References

1. Akl, S.G., Taylor, P.D.: Cryptographic solution to a problem of access control in a hierarchy. ACM Trans. Comput. Syst. 1(3), 239–248 (1983)
2. Kayem, A.V.D.M., Akl, S.G., Martin, P.: On replacing cryptographic keys in hierarchical key management systems. Journal of Computer Security 16(3), 289–309 (2008)
3. Lin, C.-H.: Dynamic key management schemes for access control in a hierarchy. Computer Communications 20(15), 1381–1385 (1997)
4. Lin, C.-H.: Hierarchical key assignment without public-key cryptography. Computers & Security 20(7), 612–619 (2001)
5. Lo, J.-W., Hwang, M.-S., Liu, C.-H.: An efficient key assignment scheme for access control in a large leaf class hierarchy. Inf. Sci. 181(4), 917–925 (2011)
6. Shen, V.R.L., Chen, T.-S.: A novel key management scheme based on discrete logarithms and polynomial interpolations. Computers & Security 21(2), 164–171 (2002)
7. Wu, T.-C., Chang, C.-C.: Cryptographic key assignment scheme for hierarchical access control. Comput. Syst. Sci. Eng. 16(1), 25–28 (2001)
8. Yang, C., Li, C.: Access control in a hierarchy using one-way hash functions. Computers & Security 23(8), 659–664 (2004)
9. Atallah, M.J., Frikken, K.B., Blanton, M.: Dynamic and efficient key management for access hierarchies. In: Atluri, V., Meadows, C., Juels, A. (eds.) ACM Conference on Computer and Communications Security, pp. 190–202. ACM (2005)
10. Millen, J.K., Clark, S.C., Freedman, S.B.: The interrogator: Protocol security analysis. IEEE Trans. Software Eng. 13(2), 274–288 (1987)
11. Longley, D., Rigby, S.: An automatic search for security flaws in key management schemes. Computers & Security 11(1), 75–89 (1992)
12. Blanchet, B.: An efficient cryptographic protocol verifier based on prolog rules. In: CSFW, pp. 82–96. IEEE Computer Society (2001)
13. Boyd, C., Kearney, P.: Exploring Fair Exchange Protocols Using Specification Animation. In: Pieprzyk, J., Okamoto, E., Seberry, J. (eds.) ISW 2000. LNCS, vol. 1975, pp. 209–223. Springer, Heidelberg (2000)
14. Tang, S.: Efficient key assignment for hierarchical access control using one-way hash function. In: Proceedings of the 10th WSEAS International Conference on Computers, ICCOMP 2006, Stevens Point, Wisconsin, USA, pp. 350–354 (2006)
15. Chen, T.-S., Huang, J.-Y.: A novel key management scheme for dynamic access control in a user hierarchy. Applied Mathematics and Computation 162(1), 339–351 (2005)
16. He, Z.H., Li, Y.-S.: Dynamic key management in a user hierarchy. In: 2nd International Conference on Anti-counterfeiting, Security and Identification, ASID 2008, pp. 298–300 (August 2008)
17. Crampton, J., Martin, K.M., Wild, P.R.: On key assignment for hierarchical access control. In: CSFW, pp. 98–111. IEEE Computer Society (2006)

Strong Minimum Energy Minimum Interference Topology in Wireless Sensor Networks

Bhawani S. Panda[1], D. Pushparaj Shetty[1], and Bijaya Kishor Bhatta[2]

[1] Computer Science and Application Group
Department of Mathematics
Indian Institute of Technology Delhi, Hauz Khas New Delhi 110016, India
[2] School of Math.-Stat-Computer Science
Utkal University, Bhubaneswar 751004, India
{bspanda,prajshetty}@maths.iitd.ac.in,
bijaya.kishor@gmail.com

Abstract. Energy minimization and interference minimization are two of the main objectives of topology control problem in wireless sensor networks. Reducing interference lowers energy consumption by reducing the number of collisions and packet retransmissions on the media access layer. Reducing transmission energy increases the lifetime of the network. In order to increase the lifetime of the network it is important to fulfil both these objectives. However, the topology control problem of minimizing interference as well the the topology control problem of minimizing total transmission power are shown to be NP-Complete. Several heuristics have been proposed for minimizing the energy and interferences separately. Only few heuristics are available in the literature which address these two problems simultaneously. In this paper, we propose a local search based heuristic for the problem of assigning transmit power to each of the n sensors such that the total power consumption and interference is minimum along with the constraint that the resulting topology consisting of bidirectional links only is strongly connected. We prove that the transmission power resulting from this heuristic is at most twice the optimal. The simulation result shows that we get significant reduction in interference as compared to the interference of existing heuristics.

Keywords: Wireless Sensor Networks, Topology Control, Minimum Spanning Tree, local search, Interference.

1 Introduction

A wireless sensor network consists of a collection of battery powered sensors each of which is integrated in a single package with low power signal processing, computation, and a wireless transceiver. The *topology control problem* can be considered as the following problem: Given a network connectivity graph, compute a subgraph with specific desired properties, such as connectivity, short stretches, sparsity, low interference or low degree node. The two main objectives of topology control problem are (i) to minimize the power consumption in the

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 177–187, 2012.

network and (*ii*) to minimize the maximum interference of the network. The topology control problem is widely studied (See [6,9,10,11,20,21]). High interference at node increases the probability of packet loss and forces the sender to retransmit the packets, which results in more energy consumption. Reducing the interference lowers energy consumption by reducing the number of collisions and consequently packet retransmissions on the media access layer. In order to increase the lifetime of the network it is important to fulfil both objectives.

We define a new topology named **Strong Minimum Energy Minimum Interference Topology(SMEMIT)** as follows.

Definition 1. *Given a set of sensors in the plane, the* strong minimum energy minimum interference topology (SMEMIT) *problem is to assign transmit power to each sensor such that* (*i*) *the resulting topology containing only the bidirectional links is strongly connected,* (*ii*) *the sum of the transmit powers assigned to all the nodes is minimized, and* (*iii*) *the maximum of the interferences of the nodes is minimized.*

In this paper we propose a local search based heuristic for the SMEMIT problem. We prove that the energy consumption of the resulting topology is at most twice the optimal and show by simulation that the maximum node interference is reduced significantly, as compared to the existing heuristics.

The rest of the paper is organized as follows. In Section 2, we discuss the graph theoretic model for wireless sensor networks and define energy minimization problem and different type of interference minimization problems. In section 3, we give summary of existing work on Energy minimization and interference minimization problems. Section 4 contains the details of proposed heuristic. In Section 5, we compare our heuristic with the existing heuristics. Section 6 concludes the paper.

2 Sensor Network Model and Energy and Interference Minimization Problems

A sensor network can be modeled by a complete weighted directed graph with vertex set V consisting of n sensors and $c : V \times V \to \Re^+ \cup \{\infty\}$ such that $c(u, v)$, the transmission cost of arc (u, v), denotes the power emission necessary for node u to send a message to node v directly. If $c(u, v) = \infty$, then node u cannot communicate to node v directly. In practice, the transmission cost function $c(u, v)$ is taken to be equal to the Euclidean distance between u and v powered by a value α ($\alpha = 2$ usually).

A **power assignment** $r : V \to \Re^+$ for a sensor network induces a **communication digraph** $D_r = (V, A_r)$, where $(u, v) \in A_r$ if and only if $r(u) \geq c(u, v)$. The presence of the arc (u, v) guarantees that the transmit power of node u is sufficiently high so that node u can send a message to node v directly. If both the arcs (u, v) and (v, u) are present in A_r, then these two arcs can be replaced by a **bidirectional arc** uv. Let $G_r = (V, E_r)$, where $E_r = \{uv|(u, v) \in A_r$ and $(v, u) \in A_r\}$. So E_r is the set of all bi-directional arcs in D_r.

The **Strong Minimum Energy Topology (SMET) problem** is to find a power assignment $r : V \to \Re^+$ such that $\sum_{u \in V} r(u)$ is minimum.

There are two important interference models considered in the literature. The **sender-interference model** and the **receiver interference model** (see [2]). In the sender-interference model, the interference of a node v is denoted by $I_S^r(v)$, and is equal to $|N_r^+(v)|$, where $N_r^+(v) = \{u \in V \setminus \{v\} | c(v,u) \leq r(v)\}$. In the receiver-interference model, the interference of a node v is denoted by $I_R^r(v)$, and is equal to $|N_r^-(v)|$, where $N_r^-(v) = \{u \in V \setminus \{v\} | c(u,v) \leq r(u)\}$. Clearly $I_S^r(v)$ gives the number of receiver nodes interfered by the sender v and is equal to the out-degree of node v, and $I_R^r(v)$ is the number of sender nodes that can interfere the receiver node v and is equal to the in-degree of node v.

Let $f_S(r) = \max_{v \in V} I_S^r(v)$, $g_S(r) = \sum_{v \in V} I_S^r(v)$, $f_R(r) = \max_{v \in V} I_R^r(v)$, and $g_R(r) = \sum_{v \in V} I_R^r(v)$.

The interference minimization problems are:

1. **MinMaxSIP :** Given (V, c), a set of nodes and the transmission cost function c, find the transmission assignment function r such that D_r is strongly connected and $f_S(r)$ is minimum.
2. **MinAVGSIP:** Given (V, c), a set of nodes and the transmission cost function c, find the transmission assignment function r such that such that D_r is strongly connected and $g_S(r)$ is minimum.
3. **MinMaxRIP:** Given (V, c), a set of nodes and the transmission cost function c, find the transmission assignment function r such that such that D_r is strongly connected and $f_R(r)$ is minimum.
4. **MinAVGRIP:** Given (V, c), a set of nodes and the transmission cost function c, find the transmission assignment function r such that such that D_r is strongly connected and $g_R(r)$ is minimum.

If follows from the work of Davide *et al.* [2] that the **MinAVGSIP** and **MinAVGRIP** are same. It is commonly believed that among the above defined interference problems, **MinMaxRIP** is most relevant (see [2]). Bidirectional links are preferred in wireless sensor networks because the messages sent over a link must be acknowledged by the receiver. The current MAC layer protocols such as IEEE 802.11 and S-MAC only take bidirectional link into consideration [5].

In view of the above, we consider the following interference minimization problem:

1. **MinMaxRIPB :** Given (V, c), a set of nodes and the transmission cost function c, find the transmission assignment function r such that G_r is connected and $f_R(r)$ is minimum.

In this paper, we study the *Strong Minimum Energy Minimum Interference Topology(SMEMIT)* problem, which can now be defined as follows.

Definition 2. *Given* (V, c), *a set of nodes and the transmission cost function* c, *find the transmission assignment function* r *such that* (i) G_r *is connected,* (ii) $\sum_{u \in V} r(u)$ *is minimum, and* (iii) $f_R(r)$ *is minimum.*

Given (V, c), a set V of n nodes and a symmetric transmission cost function $c : V \times V \rightarrow \Re + \cup\{\infty\}$ (i.e. $c(u, v) = c(v, u)$) and a power assignment function $r : V \rightarrow \Re^+$ we construct a weighted graph $G_r = (V, E)$ where $E = \{uv | r(u) \geq c(u, v)$ and $r(v) \geq c(v, u)\}$ and $w(uv) = c(u, v)$. Note that $E(G_r)$ is nothing but all the bi-directional links of D_r, the complete directed weighted graph induced by r on (V, c). Now (G_r, w) is a weighted connected spanning subgraph of the weighted complete graph (K_n, w'), where $w'(uv) = c(u, v)$.

A power assignment function $r : V \rightarrow \Re^+$ is a feasible solution to the **SMEMIT** if G_r is connected. Given a weighted spanning tree $T = (V, E')$ of (K_n, w'), we define a power assignment function $r^T : V \rightarrow \Re^+$ by $r(u) = \max_{uv \in E'}\{w'(uv)\}$. It is easy to see that G_{r^T} contains T as a subgraph and hence r^T is a feasible power assignment function. In other words, a spanning tree of a weighted complete graph (K_n, w) is a feasible solution for **SMEMIT** problem.

3 Summary of Existing Work

The SMET problem is shown to be NP-hard by Cheng *et al.* [5]. There are several heuristics for SMET problem, namely Minimum spanning tree based approximation [9], Prim-incremental heuristic [5], Valley-free heuristic [1] and Kruskal incremental heuristic [16].

The interference reduction has been the main motivation for topology control. Most of the previous work on interference reduction addresses the interference issue implicitly by constructing topologies featuring sparseness or low node degree. However Burkhart *et al.* [4] reveal that such implicit notion of interference is not sufficient to reduce the interference since message transmissions can affect nodes even if they are not direct neighbors of sending node in the resulting topology. A definition for interference is proposed in [4]. Moaveni *et al.* [14] used this definition and measured the number of nodes affected by the communication of a single communication link and give a optimal result for maximum node interference using minimum spanning tree for their model. They proved that the average node interference given by MST is at most twice the optimal. They also show by simulation that graph spanners do help in reducing the interference for a given network. Rickenbath *et al.* [18] suggested a receiver centric interference model. They proposed an algorithm for a special case when all the nodes are positioned linearly, which is called the *highway model*. They also analyzed and compared the algorithmic complexity of various interference models. Halldorrson *et al.* [7] attained an $O(\sqrt{\Delta})$ bound for the maximum interference where Δ is the interference of uniform radius ad hoc network. Moscibroda *et al.* [15] presented a greedy algorithm that computes an $O(\log n)$ approximation to the interference problem with connectivity requirement, where n is the number of nodes. Rickenbach *et al.* [19] compared both sender-centric and receiver centric models and analyzed their properties and complexities. Locher *et al.* [12] posed the following interference minimization open problem.

Problem 1. Given n nodes in a plane, connect the nodes by a spanning tree. For each node v we construct a disk centering at v with the radius equal to the distance to v's farthest neighbor in the spanning tree. The interference of a node v is then defined as the number of disks that include node v. Find a spanning tree that mini- minimizes the maximum interference.

Buchin [3] proved the posed problem to be NP-hard. Amit *et al.* [22] proposed a spanning tree based and a local search based heuristic for minimizing the interference and analyzed their performance for best and worst cases. Davide *et al.* [2] explained both sender-interference model(SI model) and receiver-interference model (RI model) and gave an optimal algorithm for MinMaxSIP. They proved that the Minimum Total Interference Problem, MTIP is the same for for both SI and RI model.

Zhang *et al.* [13] proposed an interference and energy aware topology control protocol. This is the only one protocol available in the literature that considers minimizing both energy as well as interference.

4 The Proposed Heuristic

In this section, we propose a local search based heuristic for the SMEMIT problem. Local search is an iterative heuristic used to solve many optimization problems. Typically, a local search heuristic starts with any feasible solution, and improves the quality of the solution iteratively. At each step, it considers only local operations to get a better feasible solution, if possible, in the neighborhood of the feasible solution obtained in the previous iteration. The algorithm stops in the r^{th} iteration if it fails to find a better solution in the neighborhood of the solution obtained in the $r-1^{th}$ iteration. The details about local search can be found in [8]. Our algorithm performs local changes and retains those changes that improves the interference keeping the total energy within a certain bound. Thus we retain only those changes which attains a good balance between energy and interference. The local change includes swapping a tree edge with a non-tree edge. This notion can be extended to swapping as many tree edges for an equal number of non tree edges. A local change operation involving at most k tree edges is termed a *k-change*. So *k-change* is costlier to implement than $(k-1)$-*change*. Algorithm using *k-change* generally performs better than an algorithm using $(k-1)$-*change*.

4.1 Locally Optimal Tree and the Heuristic

Let $G=(V,E)$ be a complete graph with n vertices and let $w : E \rightarrow Re^{+}$ be a weight function defined on E. Let T be a spanning tree of G. Note that a feasible solution to the SMEMIT problem is a spanning tree. Let $N_k(T) = \{T' || E(T') \setminus E(T)| = k\}$. So each $T' \in N_k(T)$ can be obtained from T by removing k edges from T and adding suitable k edges from $E(G) \setminus E(T)$. Hence, $N_k(T)$ is the k-neighborhood of T.

Let $\beta = (I(T) - I(T'))/I(T)$ where $I(T) = f_R(r)(T)$, be the ratio representing increase or decrease in interference from previous iteration to the current iteration. Clearly a negative value for β indicates no improvement in interference. It is easy to see that higher the value of β higher is the reduction in interference.

The energy may increase considerably while trying to reduce the interference. To overcome this problem, we use the following lower bound condition for computation of energy given by Aneja *et al.* [1]. Consider a minimum cost spanning tree $T(V, E')$. Then the lower bound for energy, $P(T)$ is $LB = \sum_{ij \in E'} C_{ij} + \max_{ij \in E'} C_{ij}$, where C_{ij} is the cost of $ij \in E'$.

The iteration continues if (i) $\beta > 0$ and (ii) $P(T^i) < 2 \times LB$, where T^i is the spanning obtained in the i^{th} iteration. This condition makes sure that power consumption is not more than twice the optimal. A spanning tree T admits a k-improvement if there exists a spanning tree $T' \in N_k(T)$ such that (i) $\beta > \beta'$ and (ii) $P(T') < 2 \times LB$. A k-locally optimal tree, (k-LOT) of a given graph G, is a spanning tree of G that does not admit any k-*improvement*. We explain our algorithm below which outputs the locally optimal tree k-LOT for $k = 1$.

```
  Input  : $G = (V, E, w)$, where $w$ is the cost function
  Output: Locally optimal spanning tree 1-$LOT$ of $G$.
1 Let $T$ be the minimum spanning tree of $G$;
2 while there exists a 1-improvement $T'$ on $T$ do
3     Let $T = T'$
4 Output $T$ as 1-$LOT$ of $G$.
```

Algorithm 1. SMEMIT-heuristic

Theorem 1. *The running time of SMEMIT-heuristic is polynomial in n.*

Proof. The algorithm starts with a minimum spanning tree T which can be found in $O(n^2)$ time, where n is the number of nodes in G. Then it keeps applying 1-*improvements* to the current spanning tree until it becomes a 1-LOT. The number of distinct 1-*changes* possible for a spanning tree is at most $\left(\begin{array}{c}\binom{n}{2} - (n-1) \\ 1\end{array}\right)\left(\begin{array}{c}n-1 \\ 1\end{array}\right)$, which is $O(n^3)$. Computing $P(T')$ would take at most $O(n)$ time. Computing $I(T')$ requires $O(n^2)$ times. Total computation within the loop is $\max(O(n^2), O(n)) = O(n^2)$. The number of iterations required by while loop is at most n, since in each iteration interference value is decreased by at least one, and maximum value of interference number is n. So the algorithm takes $O(n^6)$. This proves the lemma. □

Next we prove that energy resulting from SMEMIT heuristic is at most twice the optimal. i.e. SMEMIT heuristic produces a spanning tree T such that $P(T) \leq 2 \times OPT$.

Theorem 2. *The power assignment resulting from SMEMIT-heuristic has a performance ratio of* 2*.*

Proof. Let T be the MST constructed in Line 1 of SMEMIT heuristic. Let T_i be the spanning tree obtained in iteration i. The lower bound for the total energy $P(T_i)$ in each iteration is $LB = \sum_{ij \in E(T)} C_{ij} + \max_{ij \in E(T)} C_{ij}$, where C_{ij} is the cost of $ij \in E(T)$. It is clear from the algorithm that $P(T_i) < 2 \times LB$. We know that $OPT \geq LB$. Hence $P(T_i) \leq 2 \times OPT$. Hence SMEMIT heuristic results in a power assignment which is at most twice the optimal. □

5 Experimental Results

In this section, we compare the energy and interference of LMST based heuristic [13], prim-incremental power greedy heuristic for energy minimization [5], Kruskal incremental power greedy heuristic for energy minimization [16] and our proposed SMEMIT heuristic. We assume n sensors are randomly distributed in a 1000×1000 square. The power function used in the simulation study is $f(d) = t.d^{\alpha}$, where α is a constant between 2 and 4. We take $\alpha = 2$ in our simulation study, t is the threshold which is set to 1. The receiver centric interference model explained in section 2 (i.e. MinMaxRIPB) is used. For each n ranging from 10 to 100 in increments of 5, we run the heuristics 100 times with different seeds for random number generator. The average of the total powers is reported in Figure 1. The average of the maximum interference is plotted in Figure 2. We find that the energy corresponding to SMEMIT-heuristic is about 3 percent less than that of LMST heuristic. But the maximum interference of SMEMIT-heuristic is least among all other heuristics.

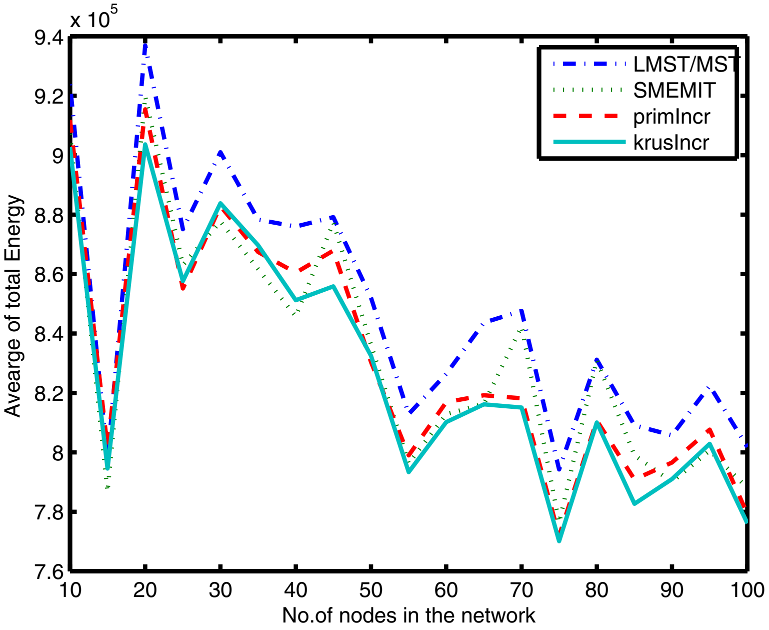


Fig. 1. Average Total Energy

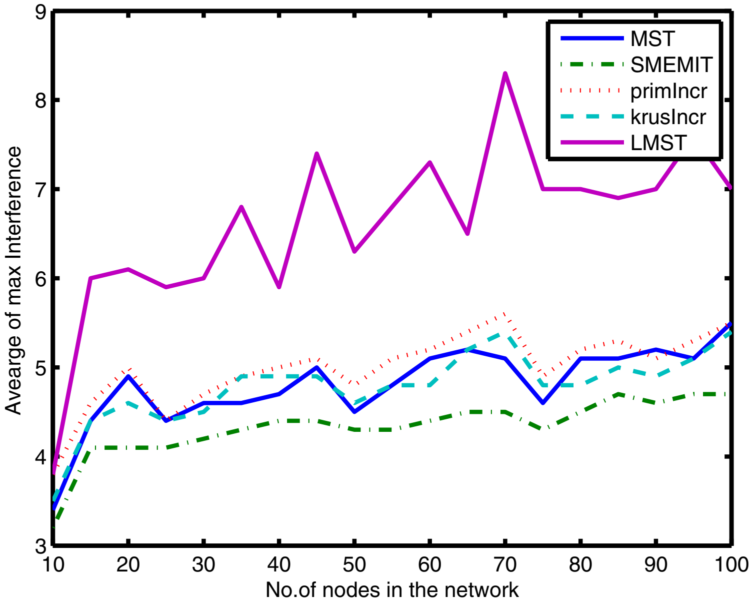


Fig. 2. Average of Maximum Interference number, (MinMaxRIPB)

Table 1. Average of maximum interference number, (MinMaxRIPB)

No.of Nodes	MST	SMEMIT	Prim-incr	Kruskal-incr	LMST
10	4.2	3.2	3.8	3.5	3.8
15	4.4	4.1	4.6	4.4	6
20	4.9	4.1	5	4.6	6.1
25	4.4	4.1	4.4	4.4	5.9
30	4.6	4.2	4.7	4.5	6
35	4.6	4.3	4.9	4.9	6.8
40	4.7	4.4	5	4.9	5.9
45	5	4.4	5.1	4.9	7.4
50	4.5	4.3	4.8	4.6	6.3
55	4.8	4.3	5.1	4.8	6.8
60	5.1	4.4	5.2	4.8	7.3
65	5.2	4.5	5.4	5.2	6.5
70	5.1	4.5	5.6	5.4	8.3
75	4.6	4.3	4.9	4.8	7
80	5.1	4.5	5.2	4.8	7
85	5.1	4.7	5.3	5	6.9
90	5.2	4.6	5.1	4.9	7
95	5.1	4.7	5.3	5.1	7.6
100	5.5	4.7	5.5	5.4	7

The performance comparison of maximum interference for different heuristics is shown in Table 1. The interference resulting from SMEMIT-heuristic is least among all other heuristics.

The performance comparison of total energy consumption for different heuristics is shown in Table 2. The average of total energy consumption in case of SMEMIT-heuristic is less than that of MST based heuristic.

Table 2. Average of Total Energy consumption

No.of Nodes	MST	SMEMIT	Prim-incr	Kruskal-incr	LMST
10	923871	907204	911876	902700	923871
15	797494	786694	801538	794555	797494
20	936993	919837	915467	903680	936993
25	875107	863064	855175	857606	875107
30	901046	877673	882865	883853	901046
35	878332	861659	867450	869707	878332
40	876021	845776	860483	851219	876021
45	879175	877397	867929	855843	879175
50	852016	836207	830408	832518	852016
55	812845	796242	798990	793320	812845
60	826373	812828	816868	810191	826373
65	843635	816311	819233	816169	843635
70	847585	842378	818169	815052	847585
75	794262	776255	771876	770101	794262
80	831217	831060	810695	810025	831217
85	809088	798936	790905	782681	809088
90	805723	790068	796570	791019	805723
95	822360	800579	807666	802883	822360
100	801796	788013	779230	776066	801796

We also compared the maximum node interference obtained for the various heuristic with the coverage based interference model using LMST based heuristic explained in [13]. The maximum link interference of the graph using this model is much higher than the maximum node interference in all heuristics.

It is also observed that the Energy plot of SMEMIT topology is always below that of MST. This is the justification of the fact that the power consumption of SMEMIT is within twice the the optimal. These studies shows that SMEMIT-heuristic performs better than the existing heuristics and achieves a better balance between energy consumption and interference of the network.

6 Conclusion

In this paper we studied strong minimum energy minimum interference topology control (SMEMIT) problem, and proposed a local search based heuristic, namely SMEMIT-heuristic for this problem. We compared the results of energy and interference of SMEMIT-heuristic with existing heuristics. Our simulation results suggest that SMEMIT-heuristic reduces the maximum interference significantly compromising the energy consumption to a small extent. We also proved that the power assignment using SMEMIT-heuristic is at most twice the optimal.

References

1. Aneja, Y.P., Bari, A., Jaekel, A., Chandrasekaran, R., Nair, K.P.K.: Minimum Energy Strong Bidirectional Topology for Ad Hoc Wireless Sensor Networks. In: IEEE ICC 2009 Proceedings (2009)
2. Bil, D., Proietti, G.: On the complexity of minimizing interference in ad-hoc and sensor networks. Theoretical Computer Science 402, 43–55 (2008)
3. Buchin, K.: Minimizing the maximum interference is hard arXiv: 0802 2134v1[cs.NI] (February 2008)
4. Burkhart, M., Rickenbach, P.V., Wattenhofer, R., Zollinger, A.: Does Topology Control Reduce Interference? In: Proceedings of ACM MobiHoc 2004, pp. 9–19 (2004)
5. Cheng, X., Narahari, B., Simha, R., Cheng, M., Liu, D.: Strong minimum energy topology in wireless sensor networks:NP-Completeness and Heuristics. IEEE Transactions on Mobile Computing 2(3), 248–256 (2003)
6. Gonzales, T. (ed.): Handbook of Approximation Algorithms and Metaheuristics, ch. 67. Chapman and Hall CRC (2007)
7. Halldorsson, M.M., Tokuyama, T.: Minimizing interference of a wireless ad-hoc network in a plane. Theoretical Computer Science 402, 29–42 (2008)
8. Lu, H.-I., Ravi, R.: The Power of Local Optimization: Approximation Algorithms for Maximum-leaf Spanning Tree. In: Proceedings Thirtieth Annual Allerton Conference on Communication, Control and Computing, pp. 533–542 (1996)
9. Kirousis, L.M., Kranakis, E., Krizane, D., Pele, A.: Power consumption in packet radio networks. Theoretical Computer Science 243, 289–305 (2000)
10. Labrador, M.A., Wightman, P.M.: Topology Control in Wireless Sensor Networks. Springer, Heidelberg (2009)
11. Lloyd, E.L., Liu, R., Marathe, M.V., Ramanathan, R., Ravi, S.S.: Algorithmic aspects of topology control problems for Ad Hoc Networks. Mobile Networks and Applications 10, 19–34 (2005)
12. Locher, T., Von Rickenbach, P., Wattenhofer, R.: Sensor Networks Continue to Puzzle: Selected Open Problems. In: Rao, S., Chatterjee, M., Jayanti, P., Siva Ram Murthy, C., Saha, S.K. (eds.) ICDCN 2008. LNCS, vol. 4904, pp. 25–38. Springer, Heidelberg (2008)
13. Luqiao, Z., Qinxin, Z.: Interference and Energy Aware Topology Control, pp. 1357–1359. IEEE (2011)
14. Moaveni-Nejad, K., Li, X.: Low-interference topology control for wireless ad hoc networks. Ad Hoc Sensor Networks International Journal 1, 41–64 (2005)
15. Moscibroda, T., Wattenhofer, R.: Minimizing interference in ad hoc and sensor networks. In: Proc. DIAL-M, pp. 24–33 (2005)
16. Panda, B.S., Pushparaj Shetty, D.: An Incremental Power Greedy Heuristic for Strong Minimum Energy Topology in Wireless Sensor Networks. In: Natarajan, R., Ojo, A. (eds.) ICDCIT 2011. LNCS, vol. 6536, pp. 187–196. Springer, Heidelberg (2011)

17. Rappaport, T.S.: Wireless communications: Principle and Practice. Prentice Hall (1996)
18. Rickenbach, P.V., Schmid, S., Wattenhofer, R., Zollinger, A.: A robust interference model for wireless ad hoc networks. In: 5th Int. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks, WMAN (2005)
19. Rickenbach, P.V., Wattenhofer, R., Zollinger, A.: Algorithmic Models of Interference in Wireless Ad Hoc and Sensor Networks. IEEE/ACM Transactions on Networking 17(1), 172–185 (2009)
20. Santi, P.: Topology Control in wireless ad hoc and sensor Networks. ACM Computing Surveys 37(2), 164–194 (2005)
21. Santi, P.: Topology Control in wireless ad hoc and sensor Networks. Wiley Inter Science (2005)
22. Sharma, A.K., Thakral, N., Udgata, S.K., Pujari, A.K.: Heuristics for Minimizing Interference in Sensor Networks. In: Garg, V., Wattenhofer, R., Kothapalli, K. (eds.) ICDCN 2009. LNCS, vol. 5408, pp. 49–54. Springer, Heidelberg (2008)

Distributed Processing and Internet Technology to Solve Challenges of Primary Healthcare in India

Arun Pande[1], Sanjay Kimbahune[1], Nandini Bondale[2],
Ratnendra Shinde[3], and Sunita Shanbhag [3]

[1] Innovation Lab, Tata Consultancy Services, Mumbai, India
{arun.pande,sanjay.kimbahune}@tcs.com
[2] School of Tech. and Comp. Sci., Tata Institute of Fundamental Research, Mumbai, India
nandini@tifr.res.in
[3] Dept of Preventive and Social Medicine, G.S. Medical College, Mumbai, India
{ratnendra.shinde,drsunitashanbhag}@gmail.com

Abstract. National Reproductive and Child Health program, focused on Mother and Child Health, part of Millennium Developmental Goals, is being implemented through Primary Health Centers of the Public Healthcare System. The health record generated through community health interventions are manual, leading to undue delay in diagnosis and emergency care. In this paper, we show how Distributed Processing and Internet Technology can be applied through innovative platform called *mHEALTH-PHC*, to provide timely, quality healthcare to remote population using existing infrastructure. *mHEALTH-PHC* combines client server, cellular, mobile phone technologies and medical test equipment to establish two way connection between patient in a village and Public Healthcare System. Our field study shows that *mHEALTH-PHC* can be effective in health surveillance, thereby leading to prompt, efficient, quality healthcare. We take the consortium approach involving IT and Public Health experts, Directorate of Health Services, pharmaceutical and health insurance industries, to make quality healthcare affordable and sustainable.

Keywords: Primary Healthcare, mHEALTH-PHC, mobile phone, Internet Technology.

1 Introduction

India has adopted the strategy of Primary Healthcare approach as recommended by Bhore committee since the inception of the five year plans [1]. Primary healthcare is defined as healthcare which is preventive, promotive, curative and rehabilitative in nature. The concept of Primary Healthcare also implies the first contact with the healthcare system, which is accessible, acceptable, affordable and appropriate to all the people in the country. The concept recognizes that health is an effective medium of achieving socio-economic development. Hence a Primary Healthcare System has been established in all states of India.

According to United Nations Foundation report, globally, every minute at least one woman dies from complications related to pregnancy or childbirth [2]. Maternal and

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 188–199, 2012.

child healthcare are the two focal points in Millennium Developmental Goals (MDGs) [3]. India being a signatory to the MDGs-2015, the Govt. of India launched its flagship program called National Rural Health Mission (NRHM) to improve the availability of and access to quality healthcare by people, especially for those residing in rural areas, the poor, women and children [4]. NRHM has emphasized on the Reproductive and Child Health (RCH) program as the top priority program implemented through Primary Health Centre (PHC) and Sub Primary Health Center (Sub-center) in all the states, leading to achievement of the MDGs. Looking at the statistics mentioned below, we are far from achieving the goal. Following section describes the existing infrastructure of Primary Healthcare System and the challenges which hamper the implementation at PHCs and Sub-centers. Section 3 overviews mobile technology for primary healthcare and states technical problems involved. Sections 4, 5 and 6 describe the technology solution using distributed processing and mobile Internet technology to overcome the challenges faced by RCH program managers, based on our field experience and the details of the technology tool called *mHEALTH-PHC*. Section 7 describes the expected outcome based on the field testing of *mHEALTH-PHC*. Section 8 states the conclusions.

2 Primary Healthcare System, RCH Program and Its Challenges

Primary healthcare is a systems approach based on the principles of equitable distribution, intersectoral coordination, community participation and appropriate technology. These principles are inculcated while implementing all national health programs. Primary healthcare as a system has a structure, a function and protocols for documentation.

2.1 The Structure

The structure of primary healthcare follows the revenue collection and distribution patterns existing in the country. Accordingly, one district is considered as a technical and administrative unit of primary healthcare services. Each district is headed by two health professionals; District Health Officer, who is in charge of preventive healthcare services in the rural areas, which include PHCs, Sub-centers and Civil Surgeon, who supervises District Hospital and Rural Hospital (RH) at Taluka level. At national level, the system is monitored through the Ministry of Health and Family Welfare. At state level, there is Directorate of Health Services which plans, coordinates and implements national health programs in all districts of the state.

For every 30,000 population in plain areas and 20,000 population in tribal and hilly areas, one PHC has been recommended. Further, for every 3,000 population in tribal areas and 5,000 population in plain areas, one Sub-center is recommended. The objective is to reach the unreached with the preventive and promotive healthcare services designed in the primary healthcare system.

Profile of PHC. One PHC covers 25-30 villages and has about 6-8 Sub-centers providing basic health services to the villagers. The Sub-center has two workers;

male health worker, called Multi-Purpose Worker (MPW) and female health worker, called Auxiliary Nurse Midwife (ANM). Every village has three health workers viz. Aanganwadi worker, ASHA (Accredited Social Health Activist) and a trained Dai /village health guide. The average distance between two PHCs is about 15-20 kms. and that between Sub-center and PHC is about 5-10 kms.

2.2 The Functions

Following are the major functions of PHC. 1) Implementation of National Health Programs. 2) Health survey of the community to establish base line health status. 3) Ensuring safe water and environmental sanitation. 4) Promote health awareness. 5) Monitor disease surveillance and health related data. 6) Epidemic investigations, containment and research.

All the activities at the PHC are documented on prescribed formats of reporting. The relevant health data is generated on daily basis and is compiled and reported on monthly basis. The PHC has both; the centre based and outreach activities.

2.3 RCH Program

RCH program focuses on preventive, promotive and curative mother and child health services. Mother and child together constitute 62% of the total population [1]. RCH program, therefore, gains more importance as the status of mother and child health is a matter of concern in India in view of the present health indicators. Following statistics are given by Balwar[5] in Indian context. The sex ratio is dismally distorted and in some states it is as low as 850 females to 1000 males. Early marriages and universality of marriages is a known factor contributing to adverse maternal and child health. 46% of women marry before the age of 18 yrs. of age. 13% of married women have unmet needs for family planning. Less than 15% of women receive adequate antenatal care. 3 out of 5 women have home deliveries. 16% deliveries occur in the absence of trained Dai and only 37% women have access to postnatal checkups.

The maternal mortality rate is still high in the country, which is 3 per 1000 live births, as against 1 per 1000 and the infant mortality rate ranges 50-70 per 1000 as against 30 per 1000, both prescribed by MDGs. The perinatal mortality rate is 49 per 1000 pregnancies. The child deaths in the first year of life account for 18.7% of total deaths in the country. In the presence of these challenging circumstances, the RCH program becomes the key program of Primary Healthcare System.

2.4 Challenges

We list some of the challenges, under which the PHC have to implement the activities of the national health programs, in general and RCH in specific.

1. The database on maternal and child health is inadequate. The health information system at Sub-center is still being operated manually. The ANM and ASHA conduct home visits to deliver antenatal and postnatal services to the mothers. They also record the health data of children below five years. This data is recorded

manually. Therefore there is inadvertent delay in transmitting this information to the PHC, located 5-10 kms away from the Sub-center. This delays the feedback from PHC and the mother is therefore deprived of prompt advice and intervention. The manual recording of data is subject to human errors, thus influencing health interventions unfavorably.

2. Accuracy, diligence, regularity and appropriateness of health information is often inadequate and requires strengthening.
3. Capacity building of health workers in terms of data management is lacking and the health records are not interpreted epidemiologically. Thus the interventions are differed till feedback is received from the PHC.
4. Accessibility of health services, lack of transport and approach roads, secondary status of women in the community, preferences for male child, short birth intervals, inadequate immunization coverage, socio-cultural practices and poverty are other compounding factors adversely affecting maternal and child health.
5. For Indian rural conditions, there are other challenges, such as, good approach roads, safe drinking water, uninterrupted power supply and availability of educational facilities for staff families to stay in villages.
6. In the current healthcare infrastructure, the large percentages of available funds are spent on salaries, vehicles, travel expenses and meager balance is available for medicines and equipment.
7. For healthcare management and monitoring system, there is a need for information flow from a patient at village level to a doctor at a PHC level and that it is monitored simultaneously at different levels of management.

The following section describes the use of mobile phones in primary healthcare and lists the technical challenges.

3 Mobile Phones for Primary Healthcare

Wireless technology, particularly cellular wireless, has created communication infrastructure in remotest areas of the countries. There are 806 million telephony subscribers including wireless and wireline connections of which 267.74 million subscribers are from rural area accounting to tele-density of 33.21% in rural areas. Phone, being a user friendly device, it is no surprise that mobile phones have exceeded the number of personal computers in the world in a span of a decade. Mobile is widely accepted as a personal device whereas computer is perceived to be complex device and hence rural masses have great hesitation in using it.

With each generation of cellular networks, data capability of the network has been improving over the years. 2.5G and 3G networks have impressive band widths. Also, mobile phones are evolving into powerful handheld devices with multimedia capabilities, at same or lower prices[6].This, we believe, is very significant technology development, especially for primary healthcare. Another major aspect is technology anxiety.

Village health worker in remote areas often need guidance from the experts. Few countries have tested call centers where health experts answer the voice queries of the village health workers**.** It is difficult for call centers to provide specific advice in absence of patient history and other observations. With appropriate user interface and tested usability of the software, mobile phone can be a robust device in the hands of village health worker to provide quality healthcare to remote village population with the help of expert advice [7,8].

3.1 Technical and Computational Challenges

While using mobile phone and Internet technology for quality healthcare, we face with the following challenges. Some have been resolved and others to be addressed in the field. Intended improvement in the existing solutions is also mentioned below.

Database Design. The data in the field comprises of patient details with medical history, health worker details, pathological test parameters, x-ray images, voice records of ANM and doctor's prescription. Along with these parameters, time as well as location of the patient needs to be tracked. This information is currently stored using relational database management system. We would like to improve the database design by combining relational and object oriented design concepts, considering events in time and space and integrating intelligence into the data relationship. Medical knowledge base for specific cases, such as pregnant women and new born babies, could also be represented in Ontology form. Ontology would get "smarter" over the years because of rich experience of treating large number of patients.

Privacy of the Data. Privacy of medical data is of great concern all over the world. Currently we have a simple scheme of User Name, Password and masking the names of the patients. Eventually we will align with national and international rules for ensuring data privacy.

Human Computer Interface (HCI). We are faced mainly with three HCI challenges; (1) Mobile phone has small real estate as compared to a PC and its keyboard is not friendly for Indian languages, (2) Most of the village health workers involved in delivering primary health care, have no exposure to using computer applications and (3) PHC doctors, though computer literate, continue to be comfortable with 'paper and pen' to write down the observations of a patient and prescribe the medicine. We have tackled these and initial solutions are in place. They can be seen from section 4 and 5.

Scalability and Computational Problems. Considering the number of ASHAs, health workers, ANMs and patients, the database would be huge and we believe we need to use Cloud Computing services as well as high capacity server for carrying out speech and handwritten character recognition.

Following section describes our technology tool called *mHEALTH-PHC* in detail.

4 Field Study and *mHEALTH-PHC*

We visited Sub-centers in rural and semi urban areas in Thane district of Maharashtra to get the first hand information on current healthcare services. Discussions with doctors at PHC, ANMs at Sub-centers and ASHAs in the villages, provided valuable feedback on our intended use of technology for timely and quality healthcare for the remote rural patients. ASHA is ‘eyes and ears’ of primary healthcare set up since she goes house to house and reports any illness or pregnancy to the Sub-center.

To deliver the primary healthcare services to rural patients by cutting down the distance barrier, we developed a platform called *mHEALTH-PHC*, which connects the rural patient to the doctor through ANM or ASHA. *mHEALTH-PHC* platform architecture reflects the processing of healthcare services at different levels in the existing primary healthcare infrastructure. This platform is based on ‘mKRISHI’ platform which was developed to provide personalized agro services to farmers using latest invent in mobile and computer technology [9].

The *mHEALTH-PHC* uses various technologies like, mobile internet, Interactive Voice Response, Indian language font rendering and usability frameworks to connect the remote Patient to the doctor. The tool has four components; 1) Software in local language on a mobile phone used at patient’s end called Client Software, 2) Servers in secured data center, 3) Doctor’s console, which is viewed by doctor/expert to suggest treatment/test and prescribe medicines to the patients, and 4) An Interactive Voice Response application to record the information/observations.

ASHA’s interface with the platform is with a simple mobile phone using Interactive Voice Response in a local language. This interface captures the ASHA’s visits to families in villages. She can also be notified any emergency or task using the same interface.

ANM at Sub-center provides preliminary healthcare to pregnant women and performs normal deliveries of babies. *mHEALTH-PHC* tool can aid ANM, as it can be integrated with the portable, battery operated medical test devices for blood and urine analysis. This facilitates inclusion of pathology test reports as part of ‘patient medical history’. Wireless Internet makes it possible to upload patient’s personal information and medical history to the server. A web console gives an integrated view of patient’s medical history including ANM’s voice comments about the patient’s illness and symptoms to PHC doctor.

Through Doctor’s Console, the doctor at PHC can view the patient’s details with medical history and listen to ANM’s recorded voice query. Doctor would record his prescription in voice/text or may handwrite using electronic pen. The prescription is sent over wireless internet to ANM’s mobile phone and ANM can take action accordingly. If required, a specialist available at RH or in city hospital could be contacted over Internet and the entire case, including patient’s medical history, could be referred through *mHEALTH-PHC* for expert advice.

Thus, all *distributed healthcare processing* units at different locations such as ASHA in village, ANM at sub-center, doctors at PHC and doctors at RH or city hospital can be connected through wireless, wireline internet and web technology. The data and information required for ‘processing healthcare’ is readily made available at each location through the network with the help of *mHEALTH-PHC*. In this paper we focus on using *mHEALTH*-PHC for RCH program run as part of primary healthcare under NRHM. Figure 1 shows the architecture of the *mHEALTH-PHC* platform.

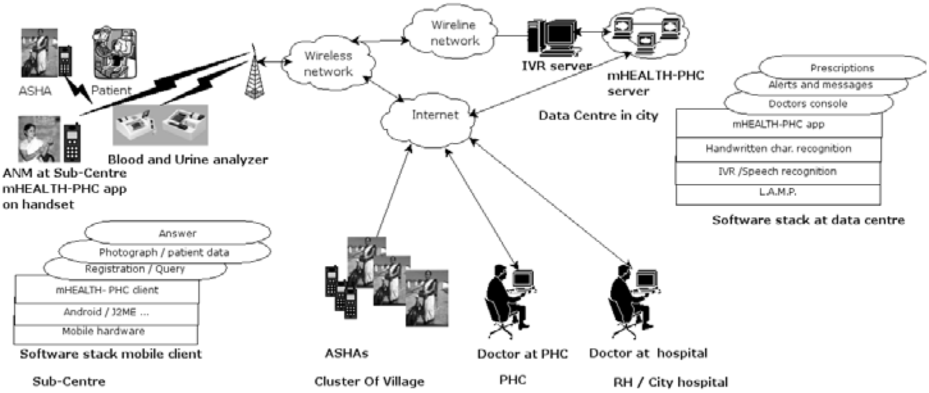


Fig. 1. *mHEALTH-PHC* Platform showing Distributed Processing

We tested the operative aspect *mHEALTH-PHC* at PHC in Thane district, Maharashtra. The results are encouraging. It was found that ANM could use the relatively complex application with minimum training of 1 hour. Along with the processes defined as part of RCH program, *mHEALTH-PHC* client software also supports other issues like tuberculosis, malaria, leprosy, water borne diseases and snake bites.

mHEALTH-PHC was selected by World's premier Health Organization for display and demonstration in their e-Health pavilion at 'World Telecom Conference 2009' hosted by International Telecommunication Union (ITU), at Geneva, Switzerland [10].

4.1 System Design and Implementation

mHEALTH–PHC design is governed by the following considerations. Though field demo of the system is successful, the system will be implemented in the field with participation from city hospital, pharmaceutical industry, micro health insurance company and the government.

Deliver high quality primary health care is the first goal Hence mechanism to audit and take corrective actions if required is built in. This is achieved by a feature where PHC doctor can seek advice of specialist at RH or city hospitals. Views are created for specialists to monitor the patient history, symptoms and the prescription given by PHC doctor. System authorizes specialist to intervene if the treatment prescribed is not optimum.

Primary Healthcare at affordable price is another goal. We have adopted Software as a Service (SaaS) architecture. This would ensure that all hardware cost and maintenance issues are handled centrally at data center. Application support issues in the field for client software will be eventually handled by MPW who would then be trained. The training would be useful to serve local community as well. Using mobile application, we have avoided power related problem which was one of the causes of failure of computer based systems in villages.

Considering village population, we have ensured that village health workers with high-school education are able to operate the system with ease and with little training. We have provided the user interface in local language using Multi Language Text Rendering (MLTR) software. Entering lot of text is avoided by using voice input.

Small display area of a mobile phone is yet another concern. We converted business processes into mobile software which ensured quick familiarity of the software to village health workers.

Our customized views enable pharmaceutical companies to know the supply and demand situation of their medicines at specific sub-center. Also, companies can post latest information on medicines on doctor's console. Similarly, customized views for micro health insurance officers would ensure efficient and low cost administration of micro health insurance policies.

5 *mHEALTH-PHC*; Specific Modules for RCH

mHEALTH-PHC platform primarily digitizes 'Mother and Child Health Card' as used in RCH program. ANM can register a pregnant woman with the help of client software on mobile phone. After registration, she gets unique identity number (Id) as is given in the registers called 'R-15' in the RCH program and the new born baby would get unique Id as is given in the conventional 'R-16' register. The Ids given by the system are different than conventional ones and the Id numbers of mother and child are linked in the system for better future reference. The platform connects two major users of the health system beneficiary, i.e. pregnant woman (via ANM) and the doctor. Hence there are two major components of this tool; ANM Interface and Doctor's Interface. The following sections cover details of these components.

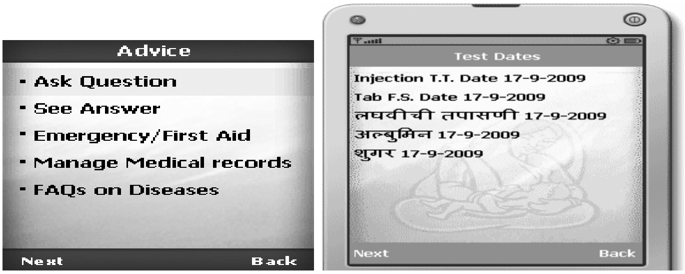
5.1 ANM Interface

ANM interface is in local language. This application captures the entire pregnancy lifecycle and post-delivery details. Based on the data, it automatically builds the history of mother beneficiary in the system. This software has the following functionalities; Authentication, New Patient Registration, Pre Delivery Checkup Module, Pre Delivery Trimester Checkup Module, Post Delivery Update, Update the history of Registered Patient and Making a Health Query.

Through 'Authentication' module, ANM can log-in in the application, using login name (her Id) and unique password. 'New Patient (Pregnant Lady) Registration' module registers the pregnant lady and captures her profile. It includes her name, age, height, husband's name, poverty line details, and the center chosen for her delivery.

'Pre Delivery Checkup Module' captures the details such as mother beneficiary's weight, pregnancy week count, the uterus height, fetal heart-count, blood pressure, swelling on face or hand. ANM can capture any prescribed medicines and/or medical tests too. 'Pre Delivery Trimester Checkup Module' Captures trimester checkup details such as blood test, hemoglobin count and percentage, urine test, tetanus injections and ferrous sulphate tablet dose. If medical equipment such as palm ultrasound, blood pressure and blood, urine test machines are integrated with the mobile, then this application can capture the test report for a given patient.

‘Post Delivery Update’ module updates the delivery details. Using this, ANM can record details for newborn child, such as birth date and time, weight at birth, gender, location, etc. It can also be used to keep the record of vaccination details for the child. Fig. 2 shows the mobile screen shots in local language.



Test Dates

Injection T.T. Date 17-9-2009
Tab F.S. Date 17-9-2009
लघवीची तपासणी 17-9-2009
अल्बुमिन 17-9-2009
शुगर 17-9-2009

Fig. 2. ANM interface Mobile Screen in English and Local language (Marathi)

‘Making a Health Query’ menu, ANM can record a voice query for the doctor. Each query gets a ‘query Id’ to track the question. The advice given by the doctor for a query can be viewed by the ANM in the application Inbox. If the query is marked as urgent, an automatic ‘SMS’ is sent to doctor to act immediately.

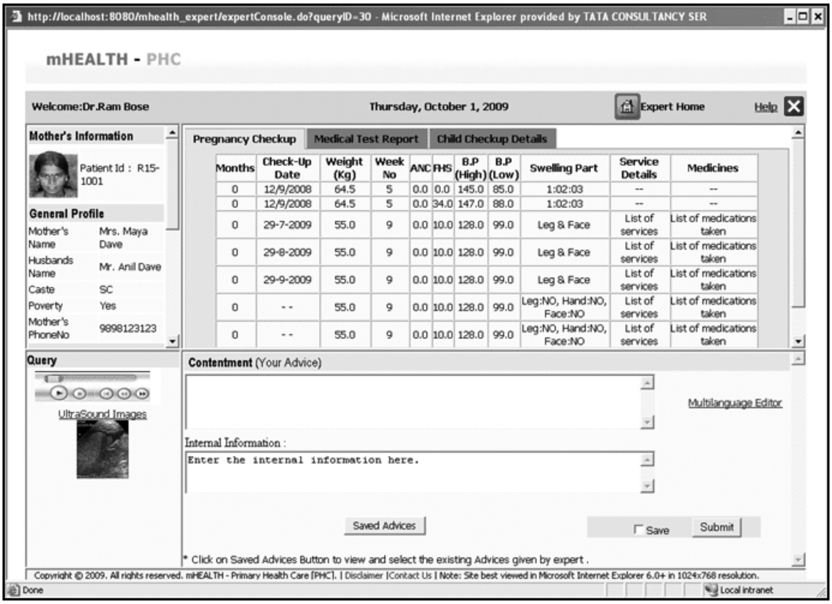
5.2 Doctor’s Interface

Infrastructure at PHC includes the facility of personal computer and internet connection. A doctor’s interface called as ‘Doctor’s Console’ can be accessed on computer. Fig. 3 shows the screen shot of Doctor’s Console. Beneficiary’s personal profile as well as medical history is available to the doctor in th eas it is currently available in RCH program format. Doctor’s console has the following functionalities; 1) Authentication, 2) Check Queries 3) Give Advice and 4) Send Alerts.

Each doctor gets a unique authentication details like user name and password. Through ‘Authentication’ module doctor can log-in to the system. Authentication prevents misuse as well as brings in accountability.

‘Check Queries’ module is a box is like an e-mail inbox. The doctor can filter on the selected queries and see the open queries. He can click on a particular query for a beneficiary and get the related details. They include the voice query or picture sent by an ANM, beneficiary’s personal profile and medical history, data related to previous treatment etc. In case the doctor at PHC is not able to resolve the query, he/she can forward the query to RH or city hospital. Doctors can also listen to various posts done by ASHAs to get rich information about health related instances in a particular area and plan accordingly.

Using ‘Give Advice’ module, the doctor can advise by analyzing the beneficiary’s profile, medical history and test reports (such as ultrasound images, blood test). The doctor can use text or voice for advising. Prescription can also be written using electronic pen. Any epidemic alerts or good practices, which need to be communicated to the larger community, the doctors can broadcast the message or send the alerts through ‘Alerts’ module. Doctor can select specific area or a group to send the alerts. The alerts can be in local language.



Patient Id : R15-1001

General Profile

Mother's Name	Mrs. Maya Dave
Husbands Name	Mr. Anil Dave
Caste	SC
Poverty	Yes
Mother's PhoneNo	9898123123

Months	Check-Up Date	Weight (Kg)	Week No	ANC	FHS	B.P (High)	B.P (Low)	Swelling Part	Service Details	Medicines
0	12/9/2008	64.5	5	0.0	0.0	145.0	85.0	1:02:03	--	--
0	12/9/2008	64.5	5	0.0	34.0	147.0	88.0	1:02:03	--	--
0	29-7-2009	55.0	9	0.0	10.0	128.0	99.0	Leg & Face	List of services	List of medications taken
0	29-8-2009	55.0	9	0.0	10.0	128.0	99.0	Leg & Face	List of services	List of medications taken
0	29-9-2009	55.0	9	0.0	10.0	128.0	99.0	Leg & Face	List of services	List of medications taken
0	- -	55.0	9	0.0	10.0	128.0	99.0	Leg:NO, Hand:NO, Face:NO	List of services	List of medications taken
0	- -	55.0	9	0.0	10.0	128.0	99.0	Leg:NO, Hand:NO, Face:NO	List of services	List of medications taken

Contentment (Your Advice)

Internal Information :

```
Enter the internal information here.
```

□ Save

* Click on Saved Advices Button to view and select the existing Advices given by expert .

Copyright © 2009. All rights reserved. mHEALTH - Primary Health Care [PHC]. | Disclaimer |Contact Us | Note: Site best viewed in Microsoft Internet Explorer 6.0+ in 1024x768 resolution.

Fig. 3. Screen shot of Doctor’s Console

6 Expected Outcome

Apart from providing the timely primary healthcare to rural remote beneficiaries, *mHEALTH-PHC* will be effective in providing holistic healthcare management system. Formal studies and preliminary project assessment in both, the developed and developing world demonstrate that mobile technology improves the efficiency of healthcare delivery and makes healthcare more effective [2]. In the current application of *mHEALTH-PHC*, all the following benefits will be available, not only for RCH program but for overall better performance at Sub-center and PHC.

1) Facilitation and ease of data collection, compilation, analysis and interpretation. 2) Prompt feedback between PHC and Sub-center. 3) The preventive interventions in maternal care such as early registration, prevention of anemia, tetanus toxoid immunization and high risk approach. 4) More institutional deliveries 5) Ensured follow up of beneficiaries. 6) Mechanism to track and monitor the mortality and the malnutrition cases. 7) Baseline data for future comparisons will be generated and quality interventions will be feasible. 8) The ‘distance and transport’ barriers in health care delivery will be overcome. 9) Effectiveness of the technology in reaching to the rural population for promoting health, its ease of application and cost-effectiveness

can be utilized for framing policy decision in functioning of primary healthcare system.

7 Consortium Approach for Implementation

Task of providing quality healthcare at affordable price to villagers is so daunting that community and private participation is required along with the government. Careful analysis of the task shows the requirement of the following participants: Qualified medical practitioner and paramedical staff, services of specialists as and when necessary, availability of medicines, pathology labs for preliminary tests, appropriate medical coverage through health insurance and update for doctors and other medical staff. This appears to be a tall order to make quality and affordable primary healthcare available to rural population. With our preliminary analysis and field experience, we believe that many of the above challenges can be addressed with consortium approach. Distributed processing and Internet technologies embedded into *mHEALTH-PHC* platform, is a natural fit to connect consortium partners. Instead of only government or only private healthcare system, the authors of this paper recommend Public Private Partnership as essential part of national healthcare system.

In consortium approach, medical experts located in cities, private hospitals can participate in delivering rural healthcare. Doctors do not have to travel except for emergency or serious cases, thus saving precious time. Medical interns can participate in rural healthcare delivery and enrich their studies of public health and social medicine. Pharmaceutical industry finds this platform ideal to build awareness of new medicines and recent advances in medical science among doctors in villages. Pathological equipment manufacturers found this platform suitable for integration to transport the data to application servers. As the consortium partners are interdependent, it will facilitate to make this approach sustainable and cost effective in long run.

8 Conclusions and Future Work

To reach the unreachable, we have proposed the innovative tool in the form of *mHEALTH-PHC* platform using technologies like, mobile internet, Interactive Voice Response, Indian language font rendering and usability frameworks, which can connect the remote patient to the doctor. Internationally, *mHEALTH-PHC* was appreciated during its demonstration at ‘ITU World Telecom Conference 2009’. During our field trials, grass root level healthcare workers appreciated its interface in local language, its user friendliness and usability. *mHEALTH-PHC* translates technology into a need based quality driven activity of mother and child healthcare. It ensures timely interventions and effective service for the beneficiaries. The entire health information system in the RCH program gets converted into a strategic management system. In future, scaled up field trials are being planned covering many Sub-centers and PHCs. More medical devices will be integrated with the system. The data collected would be analysed for trends and corrective action in terms of implementation.

Acknowledgements. Authors would like to acknowledge the management of Tata Consultancy Services, Tata Institute of Fundamental Research, Mumbai and G.S. Medical College & KEM hospital, Mumbai for their support. They also thank participation of grass root healthcare workers, doctors and eco partners for their continued support.

References

1. Park, J.E.: Textbook of Preventive and Social Medicine. Bhanodas Jalot Publications, Jabalpur (2009)
2. Vital Wave Consulting: mHealth for Development: The Opportunity of Mobile Tech. for Healthcare in Developing World. UN Foundation-Vodafone Foundation Partnership, Washington, D.C. and Berkshire, UK (2009)
3. United Nations: Millennium Developmental Goals (2015), `http://www.un.org/millenniumgoals`
4. `http://mohfw.nic.in/NRHM/Documents/Mission_Document.pdf`
5. Balwar, R. (ed.): Textbook of Public Health and Community Medicine, Armed Forces Medical College in collaboration with WHO India Office (2009)
6. Ramana, M.V.: Mobile based Primary Health Care system for Rural India, `http://www.w3.org/2008/02/MS4D_WS/papers/cdac-mobile-healthcare-paper.pdf`
7. Mechael, P., Batavia, H., Kaonga, N., Searle, S., Kwan, A., Goldberger, A., Fu, L., Ossman, J.: Barriers and Gaps Affecting mHealth in Low and Middle Income Countries: Policy White Paper (May 2010), `http://www.globalproblems-globalsolutions-files.org/pdfs/mHealth_Barriers_White_Paper.pdf`
8. McIver, W.: e-Health in the Age of Paradox: A Position Paper, `http://www.citeulike.org/user/rima/article/6699227`
9. `http://www.tcs.com/offerings/technology-products/mKRISHI/Pages/default.aspx`
10. `http://ehealth.posterous.com/mhealth-phc`

Packet Forwarding Strategies for Cooperation Enforcement in Mobile Ad Hoc Wireless Networks

Nidhi Patel and Sanjay Srivastava

DAIICT, Gandhinagar, Gujarat, India

Abstract. In self-organized ad hoc networks, all networking functions rely on the contribution of the relay nodes. Since nodes are energy constrained, nodes may not wish to relay packets for other nodes, hence leading to a drop in network throughput. In this paper, we address the issue of enforcement of cooperative behavior in network nodes. A number of approaches based either on credit or on reputation mechanisms have been proposed in the literature but these have been found to be largely unsatisfactory in the context of mobile ad hoc networks. A game theoretic approach to the solution to the cooperation problem that relies on rational and selfish behavior of network nodes may be more effective. This paper aims at determining conditions under which, such cooperation without external incentives can exist. We have focused on the packet forwarding function of the nodes and proposed a game theoretic model for achieving cooperation. Our simulation results show that cooperation is enforced in network based on a suitable strategy function under a wide range of parameters like strategies, initial condition of nodes, energy cost for sending or forwarding packets, traffic rate, noise effect, mobility, etc.

Keywords: Mobile ad hoc wireless network, cooperation, game theory, repeated game model, strategy function, payoff function, simulation.

1 Introduction

In multi-hop wireless ad hoc networks, networking services are provided by the nodes themselves, i.e. the nodes must make a mutual contribution to packet forwarding. If the network is under the control of a single authority, the nodes cooperate, i.e. military networks. If each node is its own authority, cooperation between the nodes cannot be taken for granted, but it is reasonable to assume that each node has the goal to maximize its own benefits by enjoying network services and at the same time minimizing its contribution which can significantly damage network performance.

In recent years, researchers have identified the problem of stimulating cooperation in ad hoc networks and proposed several solutions based on a reputation or on a virtual currency. This paper aims at determining under which conditions such cooperation without incentives can exist. We focus on the most basic networking mechanism, namely, packet forwarding. We define a model in a game theoretic framework for static as well as mobile networks and identify the conditions under which an equilibrium based on cooperation exists.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 200–211, 2012.

2 Modeling Packet Forwarding as a Game

2.1 System Model

Connectivity Graph. Let us consider an ad hoc network of n nodes. Let us denote the set of all nodes by N. Transmission range is fixed. Two nodes are said to be neighbors if they reside within the transmission range of each other and it is decided by calculating the distance between them. The neighbor relationship between the nodes is presented with an undirected graph known as the connectivity graph. Each vertex of the connectivity graph corresponds to a node in the network and two vertices are connected with an edge if the corresponding nodes are neighbors.

Routes. Communication is multi-hop. This means that packets from the source to the destination are forwarded by intermediate nodes. For a given source and destination, the intermediate nodes are those that form the shortest path between the source and the destination in the connectivity graph. Such a chain of nodes is called a route.

Time. We use a discrete model of time where time is divided into slots. The duration of one timeslot is 1 to 100. We assume that both the connectivity graph and the set of existing routes remain unchanged during a time slot, whereas changes may happen at the end of each time slot. We assume that the duration of the time slot is much longer than the time needed to relay a packet from the source to the destination. This means that a node is able to send several packets within one time slot.

Session. We have considered session duration different than timeslot. The session duration is calculated based on the difference between maximum session time and minimum session time, which are generated randomly in between 1 and 120 during which the session takes place. Session duration may exceed one timeslot, in that case, a session is broken into two sessions, one will be completed in current timeslot and another will be completed in next timeslot.

Packet Loss. Reasons for packet losses are non-cooperative behavior of nodes and noise (limited capacity of link or limited capacity of node).

Traffic Rate. It is non-constant bit rate.

Payoff. Payoffs for source and relay nodes are calculated based on the payoff functions. In our model, the payoff of the destination is 0 as we assume that only the source benefits if the traffic reaches the destination.

Energy Cost. We assume that the nodes have the same, fixed transmission power. So cost is the same for every node in the network. Energy cost of a node for sending or forwarding the packets is taken from the datasheet prepared after a survey of real data of nodes in sensor network.

Mobility. We assume that the nodes do not change their position during a time slot, whereas they move based on Random Waypoint model at the end of each time slot. In section 3.2, we have defined how we have generated mobile network scenario.

2.2 Strategies

Random. A node plays defect or cooperate with 0.5 probability.

Grim Trigger (GT). A node playing this strategy starts with cooperation and continues doing so unless the opponent has played defect in the past, in which case player plays defect forever.

Tit-For-Tat (TFT). A node playing this strategy starts with cooperation and then mimics the behavior of its opponent in the previous time slot.

Always Defect (AllD). A node playing this strategy defects in the first time slot.

2.3 Payoff Functions

Payoff functions for source and relay nodes are considered based on network parameters. Payoff function for source should have parameters in terms of benefit, loss and application specific. Benefit can be in terms of throughput (number of packets successfully reached to destination). Loss can be in terms of energy cost required for nodes to send the packets. Application specific parameter denotes the parameter for priority. For example, if there are two applications like chatting and ecommerce transaction then in ecommerce transaction, the payoff to a node should be more than the payoff to the node in chatting, so it is considered as a parameter in payoff function which is related to priority of an application.

The payoff ξ_s(r, t) of source node s on route r in time slot t is:

$$\xi s(r,t) = \tau(r,t).PFs - c.Ts(r)\,. \tag{1}$$

- $\tau(r,t) =$ NAR (Normalized Acceptance Rate) = the ratio of number of packets successfully reached to destination and number of packets sent.
- PFs is priority factor for source node s
- c is the cost of sending one unit of traffic
- Ts(r) is traffic rate

The payoff $\eta_f(r,t)$ of relay f on route r in timeslot t is negative and represents the cost for node f to forward packets on r in t. It is defined as follows:

$$\eta_f(r,t) = -\,c\,.Ts(r)\,. \tag{2}$$

- c is the cost of forwarding one unit of traffic
- Ts(r) is traffic rate

The total payoff Π_i(t) of node i in time slot t is computed as:

$$\Pi i(t) \quad = \sum_{q \in Si(t)} \xi_i(q,t) + \sum_{r \in Fi(t)} \eta_i(r,t)\,. \tag{3}$$

- Si(t) is the set of routes in t where i is the source
- Fi(t) is the set of routes in t where i is relay

2.4 Strategy Function

Update Cooperation Level of Nodes Based on Payoff. This function includes two factors: 1. for history, 2. for traffic

Function to calculate expected payoff in next timeslot is:

$$P_{(n+1)} = (1 - w)Pay_{(n)} + wP_n \quad . \tag{4}$$

- $P_{(n+1)}$ = expected payoff in timeslot (n+1)
- $Pay_{(n)}$ = payoff at the end of timeslot n
- P_n = Expected payoff in timeslot n

w is defined as:

$$w = w0(G(n)/<G>) \,. \tag{5}$$

- G(n) = total packets sent in nth timeslot by a node
- <G> = average traffic over past timeslots = total number of packets sent in past timeslots by node/total number of timeslots
- w0 = weight factor used for history

We have considered payoff as a parameter for updating the cooperation level based on the fact of game theory that the players play to maximize their payoff. So after completion of each timeslot, expected payoff of the node in the next timeslot is calculated and at the end of next time slot, expected payoff is compared to the actual payoff of the node and based on the difference (increment/decrement), the value of cooperation level is mapped in between 0 and 1. For example: actual payoff = a1, expected payoff = e1, now based on the comparison between these two payoff, the cooperation is calculated like (a1 +/- e1)/max(a1,e1).

2.5 Packet Forwarding Game

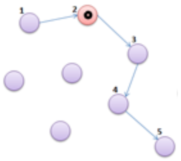


Fig. 1. Effect of transmission range on fraction of nodes converting to bad

a) From files (network scenario) generated by setdest, get x and y coordinates for all the nodes. Calculate the distance between all the nodes.

b) If distance is less than transmission range (predefined for simulation), the nodes are neighbors and the link is available.

c) Based on the value of cooperation level assigned to the relay nodes, they decide to forward or to drop the packets. If it is greater than or equal to 0.5, the node will forward the packet and if it is less than 0.5, the node will drop the packet. We have considered initial condition of the nodes which is decided based on initial cooperation levels assigned to the nodes. For example, cooperation level 0.3 to bad nodes (they will drop the packets) and 0.7 to good nodes (they will forward the packets). We can perform simulation with different number of nodes having good/bad behavior.

d) Nodes with different strategies are defined and assigned cooperation levels accordingly. Node 2 in figure 1 is defector node which drops all the packets.

e) For a single timeslot, multiple sessions are generated randomly by selecting minimum and maximum session times, source and destination pairs are randomly selected and shortest path is found. So the routes are ready with source-relays-destination. If session time exceeds the duration of one timeslot, that particular session is broken into two sessions, one will be completed in current timeslot and rest will be completed in next timeslot.

f) For all the sessions, the value of priority factor is randomly generated in between 0.5 and 1, and then assigned to the source node (node 1).

g) Traffic rate is randomly generated in between 1 and 30 for the source node (node 1).

h) Source (node 1) decides number of packets to be sent by multiplying duration of session and traffic rate.

i) The relay nodes (nodes 2, 3 and 4) decide to forward or to drop the packets according to the value of cooperation levels assigned to them. If noise effect is considered, packets are dropped based on the noise probability.

j) Payoff of source and relay nodes (nodes 1, 2, 3 and 4) is calculated when a session gets completed based on the payoff functions as defined in the section of payoff functions.

k) Total payoff of all the nodes is calculated at the end of each time slot as a node can be a source as well as relay in a single timeslot.

l) At the end of current timeslot, expected payoff of a node in next timeslot is calculated and at the end of next timeslot, this expected payoff is compared with the actual payoff of a node for updating the cooperation level of the nodes.

m) The values of payoff functions are mapped to cooperation levels in between 0 and 1 by comparing the actual payoff at the end of the current timeslot and expected payoff which was calculated at the end of previous timeslot and considering the increase/decrease in payoffs.

n) In next time slot, in presence of mobility, the location of the nodes gets changed based on Random Waypoint model as described in section 3.2. The cooperation levels of the nodes are updated based on the strategy function and the same game keeps repeating till number of timeslots defined.

3 Simulation

3.1 Mobility Model

Random waypoint mobility model is used in mobility management schemes for mobile communication systems. It is designed to describe the movement pattern of mobile users and how their location, velocity, etc change over time. The mobile nodes move randomly and freely without restrictions. The destination, speed and direction are all chosen randomly and independently of other nodes. It is a benchmark mobility model to evaluate the mobile ad hoc network routing protocols because of its simplicity and wide availability.

3.2 Wireless Network Scenarios

The node-movement generator is available under directory: ~ns/indep-utils/cmu-scen-gen/setdest. Suppose we want to create a node-movement scenario consisting of 20 nodes moving with maximum speed of 10.0m/s with an average pause between movements being 2s, the simulation time 200s and the topology boundary 500X500. The command line is: ./setdest [-n num of nodes] [-p pausetime] [-s maxspeed] [-t simtime] [-x maxx] [-y maxy] > [outdir/movement-file], which will look like: ./setdest -n 20 -p 2.0 -s 10.0 -t 200 -x 500 -y 500 > scen-20-test. We redirect the output to file scen-20-test. The file begins with the initial position of the nodes and goes on to define the node movements. The setdest program generates node-movement files using the random waypoint model.

3.3 Simulation Parameters

Table 1. Simulation parameters

Parameter	Value
Number of nodes	n=25, n=50
Timeslot	100, 150, 200
Area	500X500
Transmission range	150m, 200m, 250m, 300m
Mobility model	Random waypoint model
Strategies	TFT, AllD, Random, GT
Energy cost	0.015A, 0.15A
Priority factor	randomly between [0.5,1]
Weight factor (history)	0.3, 0.6, 0.9
Total simulation runs	100
Number of sessions per timeslot	randomly between [n/2,n]
Initial cooperation level	good - 0.7, bad - 0.3
Network	static, mobile
Traffic rate	[1-15], [1-30] pkts per sec
Noise probability	1/10, 1/30, 1/50

4 Result and Analysis

Figure 2 shows that when transmission range is less, more fraction of nodes are converted to bad as less number of nodes are directly connected and when the transmission range is more, less fraction of nodes are converted to bad as more number of nodes are directly connected. So, as transmission range increases, fraction of nodes converting to bad decreases.

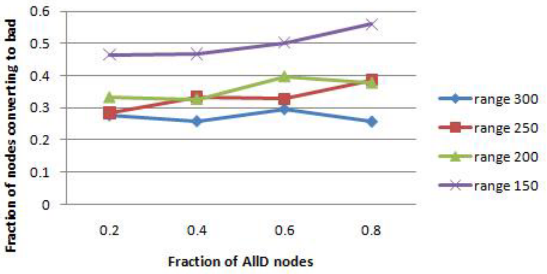


Fig. 2. Effect of transmission range on fraction of nodes converting to bad

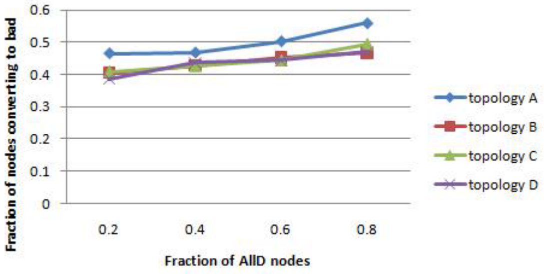


Fig. 3. Effect of topology on fraction of nodes converting to bad

Figure 3 shows the comparison of fraction of nodes converting to bad for different network scenarios generated based on Random WayPoint model. Topology A is the worst case scenario in which less number of nodes are directly connected compared to others (more relays are present in a route so more chances of packet dropping), still there exists cooperation in the network. So, the performance of strategy function is independent of topology.

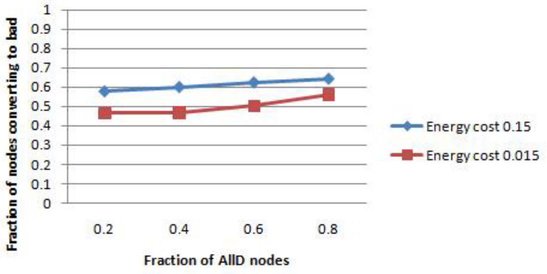


Fig. 4. Effect of energy cost on fraction of nodes converting to bad

Figure 4 shows that if energy cost for sending or forwarding packets is more, then the fraction of nodes converting to bad are more as the payoff which they get is reduced. So, as energy cost increases, fraction of nodes converting to bad increases.

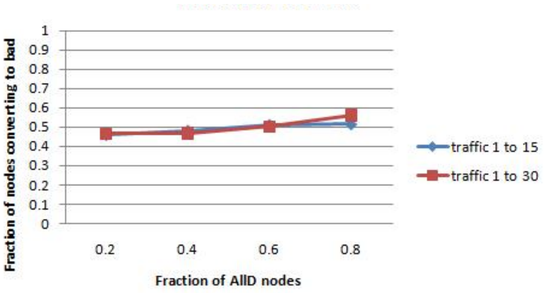


Fig. 5. Effect of traffic on fraction of nodes converting to bad

Figure 5 shows the comparison between high traffic density and less traffic density. So we can say that traffic density has no significant effect on our game model. So, the performance of strategy function is independent of traffic rate.

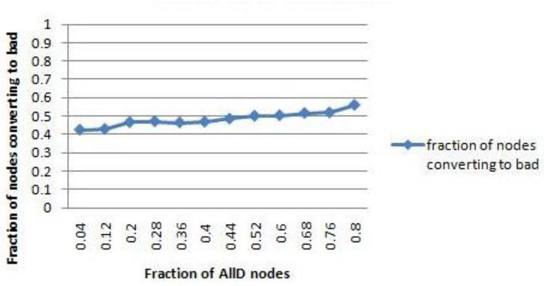


Fig. 6. Effect of AllD strategy on fraction of nodes converting to bad

Figure 6 shows that the number of nodes with AllD strategy does not affect all the nodes in the network, so we can say that cooperation still exists in presence of large number of defector nodes. As we are using payoff as a parameter for updating the cooperation level of the nodes, even if more number of defector nodes are present in the network, less fraction of good nodes are converted to bad. So, the performance of the strategy function is independent of the AllD strategy.

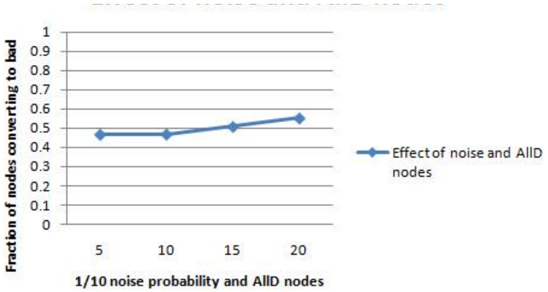


Fig. 7. Effect of noise on fraction of nodes converting to bad

Figure 7 shows the effect of noise as well as AllD nodes. Our game model is robust against noise. Adding to that, we have simulated the scenario where AllD nodes are also present with some noise. So, the performance of strategy function is independent of the noise effect.

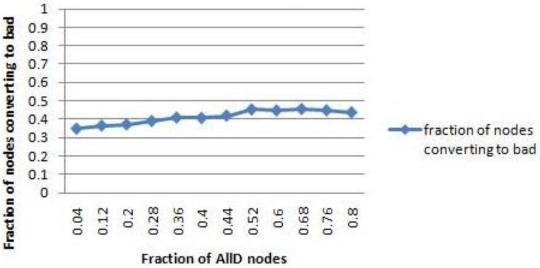


Fig. 8. Effect of AllD strategy with mobility on fraction of nodes converting to bad

Figure 8 shows that mobility increases mutual dependencies between the nodes; thus, mobility gives a natural incentive for cooperation. So less fraction of nodes are converted to bad.

We have also simulated with mixed strategies including random strategy, grim trigger strategy and AllD strategy in the same scenario for which the cooperation is achieved as very less fraction of nodes are converted to bad.

5 Related Work

5.1 Cooperation without Incentive Mechanisms

[15] Addresses the problem of whether cooperation can exist without incentive mechanisms and proposes a model based on game theory to investigate equilibrium conditions of packet forwarding strategies. We have extended the work. We have relaxed some of the assumptions and added some new. We have introduced new payoff functions, strategy function to update the cooperation level of the nodes, strategies. We are considering noise, mobility and initial condition of nodes.

An approach that addresses cooperation in the absence of any incentive mechanism is provided by Srinivasan et al. [5]. Their work focuses on the energy-efficient aspects of cooperation. In their solution, the nodes are classified in different energy classes. The nodes differentiate among the connections based on the energy classes of the participants and apply different behaviors according to the type of the connection. This framework relies on an ideal mechanism that distributes class information. It requires a secure mechanism to prevent malicious nodes from cheating with the class information provided by the relays to the source. We introduce a game theoretical model that does not rely on any additional mechanism, thus we believe our investigations to be more generic. Srinivasan et al. also make use of time slots, but they generate only one communication session for the whole network in each time slot. They randomly choose the participating nodes for this session.

5.2 Incentive Mechanisms in Ad Hoc Network

Marti et. al. [4] considers an ad hoc network where some misbehaving nodes agree to forward packets but then fail to do so. They propose a mechanism, called watchdog, in charge of identifying the misbehaving nodes, and a mechanism, called pathrater, that deflects the traffic around them. However, misbehaving nodes are not punished, and thus there is no motivation for the nodes to cooperate. To overcome this problem, Buchegger and Le Boudec [7] as well as Michiardi and Molva [8] define protocols that are based on a reputation system. In both approaches, the nodes observe the behavior of each other and store this knowledge locally. Additionally, they distribute this information in reputation reports. According to their observations, the nodes are able to behave selectively (e.g., nodes may deny forwarding packets for misbehaving nodes).

Zhong et al. [9] presents a solution, where an offline central authority collects receipts from the nodes that relay packets and remunerates them based on these receipts. Another solution, presented by Buttyan and Hubaux [19], [10], is based on a virtual currency, called nuglets: If a node wants to send its own packets, it has to pay for it, whereas if the node forwards a packet for the benefit of another node, it is rewarded.

An incentive mechanism is proposed for multi-hop cellular networks by Jakobsson et al. [17]. They use the concept of lottery tickets to remunerate the forwarding nodes in a probabilistic way. They consider an asymmetric scheme where the uplink (from the initiator to the base station) is multi-hop and the downlink (from the base station to the initiator) is single-hop. Ben Salem et al. [18] investigate the symmetric scheme where both uplink and downlink are multi-hop. They use the concept of sessions to authenticate the nodes involved in a given communication and to correctly perform the charging and rewarding mechanism. Lamparter et al. [20] consider a charging scheme for ad hoc stub networks that relies on the presence of an Internet Service Provider.

6 Conclusion

It has been shown that a game theoretic model based on the rational and selfish behaviour of the nodes can prove effective in enforcing cooperative behaviour among the network nodes. Our simulation results show that under a range of game strategies, the cooperation behaviour of the nodes is quite robust. Further, strategies can be designed that are able to limit the uncooperative behaviour from spreading across the networks. This ability of the approach to limit the number of uncooperative nodes is clearly an important result.

We show that there may not be a need to keep track of individual behaviour of other nodes thus minimizing the energy overhead for cooperation enforcement. As the nodes interact with the network without identifying the players it interacts with, no authentication of the nodes is required. Our simulation results show that even in a more realistic model where noise is present, the behaviour of defectors affects only a fraction of the nodes in the network. We have also shown the existence of a cooperative equilibrium of packet forwarding strategies by considering various network parameters for simulation.

References

1. Axerod, R.: The evolution of Cooperation, revised edn. (2006)
2. C. From. Accepted From Open Call Why Does It Pay To Be Selfish in A MANET? IEEE Wireless Communications, 87–97 (2006)
3. Bandyopadhyay, S., W.L.: A Game-Theoretic Analysis on the Conditions of Cooperation in a Wireless Ad Hoc Network Electrical and Computer Engineering. Information Sciences
4. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In: Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2000), pp. 255–265 (2000)
5. Srinivasan, V., Nuggehalli, P., Chiasserini, C.F., Rao, R.R.: Cooperation in Wireless Ad Hoc Networks. In: Proceedings of IEEE INFOCOM 2003, San Francisco, March 30-April 3 (2003)
6. Srinivasan, V., Nuggehalli, P., Chiasserini, C., Rao, R.R.: An Analytical Approach to the Study of Cooperation in Wireless Ad Hoc Networks, vol. 4, pp. 722–733 (2005)
7. Buchegger, S., Le Boudec, J.-Y.: Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes-Fairness In Dynamic Ad-hoc NeTworks). In: Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002), Lausanne, Switzerland, June 9-11, pp. 80–91 (2002)
8. Michiardi, P., Molva, R.: Core: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks. In: Communication and Multimedia Security 2002, Portoroz, Slovenia, September 26-27 (2002)
9. Zhong, S., Yang, Y.R., Chen, J.: Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad Hoc Networks. In: Proceedings of IEEE INFOCOM 2003, San Francisco, March 30-April 3 (2003)
10. Buttyán, L., Hubaux, J.-P.: Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on Mobile Ad Hoc Networks 8(5) (October 2003)
11. Seredynski, M., Bouvry, P.: Modelling the Evolution of Cooperative Behavior in Ad Hoc Networks using a Game Based Model. Evaluation
12. Seredynski, M., Bouvry, P.: Evolutionary Game Theoretical Analysis of Reputation-based Packet Forwarding in Civilian Mobile Ad Hoc Networks. Technology
13. Crosby, G.V., Pissinou, N.: Evolution of Cooperation in Multi-Class Wireless Sensor Networks (2007)
14. Félegyházi, M., Buttyán, L., Hubaux, J.-P.: Equilibrium Analysis of Packet Forwarding Strategies in Wireless Ad Hoc Networks – The Static Case. In: Conti, M., Giordano, S., Gregori, E., Olariu, S. (eds.) PWC 2003. LNCS, vol. 2775, pp. 776–789. Springer, Heidelberg (2003)

15. Félegyházi, M., Buttyán, L., Hubaux, J.-P.: Equilibrium Analysis of Packet Forwarding Strategies in Wireless Ad Hoc Networks - the Static Case. In: Conti, M., Giordano, S., Gregori, E., Olariu, S. (eds.) PWC 2003. LNCS, vol. 2775, pp. 776–789. Springer, Heidelberg (2003), `http://lcawww.epfl.ch/felegyhazi/`
16. Member, S., Hubaux, J., Member, S.: Nash Equilibria of Packet Forwarding Strategies in Wireless Ad Hoc Networks. IEEE Transactions On Mobile Computing (2004)
17. Jakobsson, M., Hubaux, J.-P., Buttyán, L.: A Micro-Payment Scheme Encouraging Collaboration in Multi-hop Cellular Networks. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 15–33. Springer, Heidelberg (2003)
18. Ben Salem, N., Buttyán, L., Hubaux, J.P., Jakobsson, M.: A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular Networks. In: Proc. 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), Annapolis, USA, June 1-3 (2003)
19. Buttyán, L., Hubaux, J.-P.: Enforcing Service Availability in Mobile Ad Hoc WANs. In: Proc. 1st ACM/IEEE International Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc 2000), Boston, MA, USA (August 2000)
20. Lamparter, B., Paul, K., Westhoff, D.: Charging Support for Ad Hoc Stub Networks. Journal of Computer Communication, Special Issue on Internet Pricing and Charging: Algorithms, Technology and Applications (2003)

A Study on Scalability of Services and Privacy Issues in Cloud Computing

R.S.M. Lakshmi Patibandla, Santhi Sri Kurra, and Nirupama Bhat Mundukur

Department of MCA, School of Computing, Vignan University,
Vadlamudi, Guntur, AndhraPradesh, India
{patibandla.lakshmi,srisanthi,nirupamakonda}@gmail.com

Abstract. Cloud Computing is rapidly emerging and the new development in Information Technology. There are many patterns, or categories, in the world of cloud computing that are needed for the enterprise architecture. Some of the categories of services are storage, database, information, process, application, platform, integration, security, privacy, management/governance, testing, and infrastructure. Scalability is one of the important features applied on any of the services. The existing analysis specially focuses on Architectural and policy Implications without exploring the data privacy issues. In this paper, the application scalability and data privacy initiatives on various services in cloud environments are presented, with an overview of the trends they follow.

Keywords: Services, Scalability, Privacy, Cloud Environment.

1 Introduction

Cloud computing is commonly associated to offering of new mechanisms for infrastructure provisioning [2,1]. The illusion of a virtually infinite computing infrastructure, the employment of advanced billing mechanisms allowing for a pay-per-use model on shared multitenant resources, the simplified programming mechanisms (platform), etc. are some of the most relevant features. Among these features/challenges, those introduced by adding scalability and automated on-demand self-service are responsible for making any particular service something more than "just an outsourced service with a prettier marketing face" [4]. As a result of its relevance, the wealth of systems dealing with "cloud application scalability" is slowly gaining weight in the available literature [3,5, 6, 7, 8, 9, 10, 11, 12, 14]. As can be observed in the previous references, automation is typically achieved by using a set of service provider-defined rules that govern how the service scales up or down to adapt to a variable load. These rules are themselves composed of a condition, which, when met, triggers some actions on the infrastructure or platform. The degree of automation, abstraction for the user (service provider) and customization of the rules governing the service vary. Some systems offer users the chance of building rather simple conditions based on fixed infrastructure/platform metrics (e.g. CPU, memory,

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 212–230, 2012.

etc.), while others employ server-level `1q a combinations of simple rules) to be included in the rules. Regarding the subsequent actions launched when the conditions are met, available efforts focus on service horizontal scaling (i.e. adding new server replicas and load balancers to distribute load among all available replicas) or vertical scaling (on-the-fly changing of the assigned resources to an already running instance, for instance, letting more physical CPU to a running virtual machine (VM)). Unfortunately, the most common operating systems do not support on-the-fly (without rebooting) changes on the available CPU or memory to support this "vertical scaling". It is, thus, necessary to extend infrastructure clouds to other kinds of underlying resources beyond servers, LBs and storage. Cloud applications should be able to request not only virtual servers at multiple points in the network, but also bandwidth-provisioned network pipes and other network resources to interconnect them (Network as a Service, NaaS) [18]. Clouds that offer simple virtual hardware infrastructure such as VMs and networks are usually denoted Infrastructure as a service Clouds (IaaS) [1, 17].

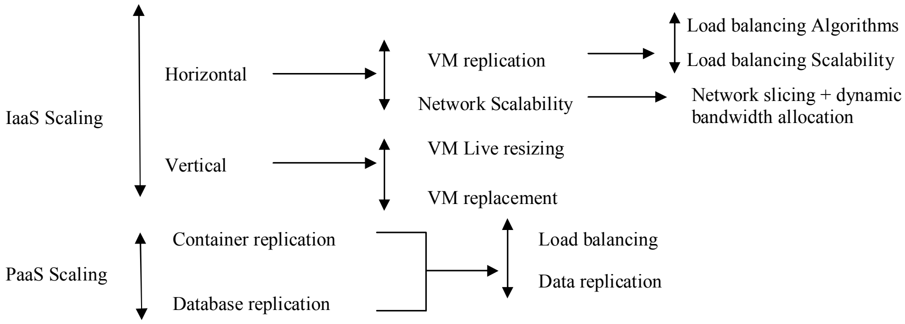


Fig. 1. Summary of the Available Mechanisms for Holistic Application Scalability

A different abstraction level is given by Platform as a Service (PaaS) clouds. PaaS clouds supply a container-like environment where users deploy their applications as software components [20]. PaaS clouds provide sets of "online libraries" and services for supporting application design, implementation and maintenance. Despite being somewhat less flexible than IaaS clouds, PaaS clouds are becoming important elements for building applications in a faster manner [1,2] and many important IT players such as Google and Microsoft have developed new PaaS clouds systems such as Google App Engine1 (GAE) and Microsoft Azure2. Due to their importance this document also discusses scalability in PaaS clouds at two different levels: container level and database level. Figure 1 provides an overview of the mechanisms handy to accomplish the goal of whole application scalability. Computing user of their private data being stolen or misused, and also assists the cloud computing provider to confirm to privacy law. Cloud computing, in which services are carried out on behalf of customers on hardware that the customers do not own or manage, is an increasingly fashionable business model. The user to the cloud, which means that they typically result in users' data, uploads the input data for cloud services being present in unencrypted form on a machine that the user does not own or control. This poses some inherent privacy challenges. There is a risk of data theft from machines in the cloud, by rogue employees of cloud service providers or by data thieves breaking into service providers' machines, or even by other customers of the same service if there is inadequate separation of different customers' data in a machine that they share in the cloud. Governments in the countries where the data is processed or stored may have legal rights to view the data under some circumstances [2,1]. There is also a risk that the data may be put to unauthorized uses. It is part of the standard business model of cloud computing that the service provider may gain revenue from authorized secondary uses of user's data, most commonly the targeting of advertisements. However, some secondary data uses would be very unwelcome to the data owner (such as, for example, the resale of detailed sales data to their competitors). At present there are no technological barriers to such secondary uses. There are, however, some legal constraints on the treatment of users' private data by cloud computing providers. Privacy laws vary according to jurisdiction. The structure of the paper is organized as follows. In section 2 we present server scalability. In section 3 we discuss scaling the network platform for cloud computing. In section 4 we describe privacy management and its Architectures, giving an overview of how the privacy manager may be used. In section 5 the paper concludes with a general analysis and discussion of next steps.

2 Server Scalability

Most of the available IaaS clouds deal with single VM management primitives (e.g. elements for adding/removing VMs) [3, 5, 6, 7, 8], lacking mechanisms for treating applications as a whole single entity and dealing with the relations among different application components; for instance, relationships between VMs are often not considered, ordered deployment of VMs containing software for different tiers of an application is not automated (e.g. the database's IP is only known at deployment time; thus, the database needs to be deployed first in order to get its IP and configure the Web server connecting to it), etc. Application providers typically want to deal with their application only [12, 19], being released from the burden of dealing with (virtual) infrastructure terms.

2.1 Towards Increased Abstraction and Automation: The Elasticity Controller

Such a fine-grained management (VM-based) may come in handy for few services or domestic users, but it may become intractable with a big number of deployed applications composed of several VMs each. The problem gets worse if application providers aim at having their application automatically scaled according to load. They would need to monitor every VM for every application in the cloud and make decisions on whether or not every VM should be scaled, its LB re-configured, its network resources resized, etc. Two different approaches are possible: 1) increasing the abstraction level of the provided APIs and/or 2) advancing towards a higher automation degree. Automated scaling features are being included by some vendors [3, 10], but the rules and policies they allow to express still deal with individual VMs only; one cannot easily relate the scaling of the VMs at tier-1 with its load balancers or the scaling of VMs at tier-3, for instance. As an example of this behavior, Marshall et al. proposed a resource manager built on top of the Nimbus toolkit that dynamically adapted to a variety of job submission patterns increasing the processing power up to 10 times by federating on top of Amazon's (deploying new VMs adhered to the cluster on separate infrastructures) [14].On the other hand, [19, 11, 12] propose more abstract frameworks (they allow users to deal with applications as a whole, rather than per individual VM) that also convey automation. Unavoidably, any "scalability management system"(or elasticity controller in Figure 2) is bound to the underlying cloud API (the problem of "discrete actuators" as named by Lim et al. [11]). One essential task for any application-level elasticity controller is, thus, mapping user scaling policies from the appropriate level of abstraction for the user to the actual mechanisms provided by IaaS clouds (depending on the specific underlying API). The implementation of the elasticity controller can be done in several different ways with regard to the provided abstraction level: • a per-tier controller, so that there is a need for coordination and synchronization among multiple controller actuator pairs [11]. Treating each tier as an independent actuator with its own control policy can cause shifting of the performance bottleneck between tiers. Lim et al propose that a tier can only release resources when the other tiers are not holding an interlock. A single controller for the whole application (for all tiers), which let users specify how an application should scale in a global manner. For instance, the application provider could specify (based on her accurate knowledge of her application) to scale the application logic tier whenever the number of incoming requests at the web tier is beyond a given threshold [12].

2.2 Expressing How and When to Scale: Feeding Controller with Rules and Policies

All the works above rely on traditional control theory in which several sensors feed a decision making module (elasticity controller) with data to operate on an actuator

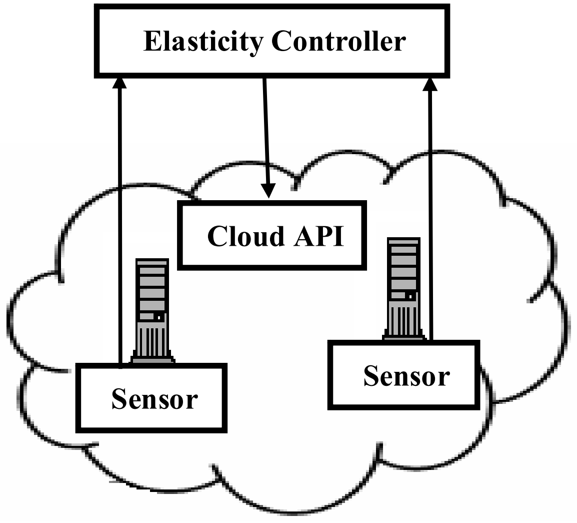


Fig. 2. Conventional Control Theory Applied to the Application Level Scalability in the Cloud

(cloud API), as shown in Figure 2. Similarly, all the systems above answer the question on how to automate scalability (i.e. how to implement the elasticity controller) in a similar manner either for VM-level [3, 10] or for application-level “scalability management systems” [11, 12].

User-defined rules are the chosen mechanism (as opposed to reconfigured equations sets) to express the policies controlling scalability as shown below. A rule is composed of a series of conditions that when met trigger some actions over the underlying cloud. Every condition itself is composed of a series of metrics or events that may be present in the system, such as “money available” or “authorization received” (either provided by the infrastructure or obtained by the application provider herself), which are used together with a modifier (either a comparison against a threshold, the presence of such events) to trigger a set of cloud-provided actions.

RULE:

if CONDITION(s) then ACTION(s)

CONDITION:

(1.. *) (metric. value MODIFIER)

ACTION:

(*) IaaS cloud-enabled actions (e.g. deploy new VM)

If we focus on application-level scalability (rather than dealing with per VM scaling), Lim et al. present a mathematical formulation in which the user just configures the threshold4 for replicating storage servers so as to increase the cloud storage capability [22, 11]. Rodero-Merino et al. leave full control for users to express their rules and use a rule engine as a controller. The Open Virtual Format (OVF) is extended to define the application, its components, its contextualization needs, and the rules for scaling [9]. This way, service providers can generate a description of their application components, the way their application behaves with regards to scalability and the relevant metrics that will trigger the actions expressed in such rules. As shown in Figure 1, in addition to dynamically adding more VM replicas, application behavior could also include many other aspects determining application scalability: adding load balancers to distribute load among VM replicas is also which considers the fact that going from 1 to 2 machines can increase capacity by 100% but going from 100 to 101 machines increases capacity by no more than 1%. An important point. In an IaaS cloud the number of VMs balanced by a single LB can hugely increase, thus overloading the balancer5.

LB scalability requires the total time taken to forward each request to the corresponding server to be negligible for small loads and should grow no faster than O (p) (p being the number of balanced VMs) when the load is big and the number of balanced VMs is large [21]. However, although mechanisms to scale the load-balancing tier would benefit any cloud system, they are missing in most current cloud implementations. Amazon already providers its Elastic Load Balancer service aimed at delivering users with a single LB to distribute load among VMs. However, Amazon does not provide mechanisms to scale LBs themselves. Virtualization of the load balancing machines offers the chance to define scalability rules by using previously presented systems [19, 11, 12]. Recently, Liu and Wee [24, 23] proposed a "rule of thumb" procedure to configure presentation-tier (Web server) scalability: for CPU-intensive web applications, it is better to use a LB to split computation among many instances. For network intensive applications, it may be better to use a CPU-powerful standalone instance to maximize the network throughput. Yet, for even more network intensive applications, it may be necessary to use DNS load balancing to get around a single instance bandwidth limitation [24]. The question emerges whether DNS-based load balancing can be scalable enough or not. Practical experiences with large well-known services such as Google's search engine and recent experimental works on cloud settings [24, 23] seem to point in that direction. Also, for many enterprise applications, hardware LB is the most common approach.

3 Scaling the Network and Platform

Properly replicated/sized VMs led us to think about LBs as a possible bottleneck. Assuming this problem is also resolved takes us to think about the link that keeps application elements stuck together, even across different cloud infrastructure services: the network, which is often overlooked in cloud computing. Networking over virtualized resources is typically done in two different manners: "Ethernet virtualization" and overlay networks and TCP/IP virtualization. These techniques are respectively focused in the usage of virtual local area network (VLAN) tags (L2) to separate traffic or public key infrastructures to build L2/L3 overlays [26, 25, 18, 28]. Separating users' traffic is not enough for reaching complete application scalability: the need to scale the very network arises in consolidated data centers hosting several VMs per physical machine. This scalability is often achieved by over-provisioning the resources to suit this increased demand. This approach is expensive and induces network instability while the infrastructure is being updated. Also, it is static and does not take into account that not all the applications consume all the required bandwidth during all the time. Improved mechanisms taking into account actual network usage are required. On the one hand, one could periodically measure actual network usage per application and let applications momentarily use other applications' allocated bandwidth. On the other hand, applications could request more bandwidth on demand over the same links [18]. Baldine et al. proposed to "instantiate" bandwidth-provisioned network resources together with the VMs composing the service across several cloud providers [18]. Similar to the OVF extensions mentioned above, these authors employ Network Description Language (NDL)-based ontologies for expressing the required network characteristics. These abstract requirements are mapped to the concrete underlying network peculiarities (e.g. dynamically provisioned circuits vs. IP overlays). A complete architecture to perform this mapping has also been proposed [27]. Unfortunately, there is no known production-ready system that fully accomplishes the need for dynamically managing the network in synchrony with VMs provisioning. These techniques to increase the utilization of the network by virtually “slicing” it have been dubbed as “network as a service” [18]. This *`a la cloud* network provision paradigm can be supported by flow control [30], distributed rate limiting [30], and network slicing techniques [32]. By applying this mechanism the actual bandwidth can be dynamically allocated to applications on demand, which would benefit from a dynamic resource allocation scheme in which all the users pay for the actual bandwidth consumption. To optimize network usage statistical multiplexing is used to compute the final bandwidth allocated to each application. Statistical multiplexing helps to allocate more bandwidth to some applications while some others are not using it (most system administrators usually provision on a worst-case scenario and never use all the requested resources). This way, the cloud provider can make a more rational use of its network resources, while still meeting applications’ needs. IaaS clouds are handy for application providers to control the resources used by their systems. However, IaaS clouds demand application developers or system administrators to install and configure all the software stack the application components need. In contrast, PaaS clouds offer a ready to use execution environment, along with convenient services, for applications. Hence, when using PaaS clouds developers can focus on programming their components rather than on setting up the environment those components require. But as PaaS clouds can be subject to an extensive usage (many concurrent users calling to the hosted applications), PaaS providers must be able to scale the execution environment accordingly. In this section, we will explore how scalability impacts on the two core layers of PaaS platforms: the *con- tainer* and the *database management system* (DBMS), as they are the backbone of any PaaS platform: the combination of container + database is the chosen stack to implement many networked (e.g. Internet) applications, which are the ones PaaS platforms are oriented to. The container is the software platform where users’ components will be deployed and run. Different PaaS clouds can be based on different platforms, for example GAE and its open source counterpart AppEngine [31] provide containers for servlets (part of the J2EE specification) and Python scripts, while Azure and Aneka [34] offer an environment for .NET applications. Each platform type can define different lifecycles, services and APIs for the components it hosts. Databases, on the other hand, provide data persistence support.

The database storage service must address the de-mand for data transactions support combined with big availability and scalability requirements. As it is explained later in this section, this can be addressed in different manners. Figure 3 shows an overview of the possible architecture of a PaaS system, where both the container and the database layer achieve scalability through replication of the container and the DBMS (horizontal scaling). This is the scenario this work focuses on, as it is the only one the authors deem feasible in clouds with certain scalability requirements. Applying vertical scaling by using “more powerful” hardware would soon fail, as many clouds will typically face loads that one single machine cannot handle whatever its capacity. Other services can be offered by a PaaS platform apart from the container and the database, which also will need to be scaled to adapt to demand. For example,

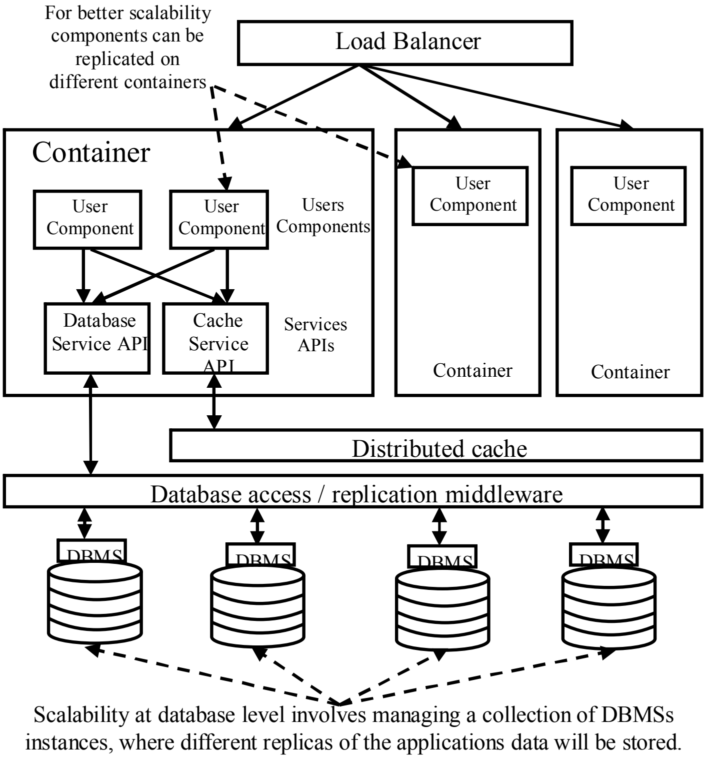

For better scalability components can be replicated on different containers
Load Balancer
Container
User Component
User Component
Users Components
Database Service API
Cache Service API
Services APIs
User Component
Container
User Component
Container
Distributed cache
Database access / replication middleware
DBMS
DBMS
DBMS
DBMS
Scalability at database level involves managing a collection of DBMSs instances, where different replicas of the applications data will be stored.

Fig. 3. Schematic View of a PaaS System

Azure offers services for inter-component communication (bus) or access control. Unfortunately, in this work it would only be possible to study scalability issues of all those services in a too superficial manner. Instead, a most thorough analysis of the most important services, the container and the database, has been preferred.

3.1 Container-Level Scalability

At container level, a better scalability can be achieved by enabling multitenant containers (having the ability to run components belonging to different users). This imposes strong isolation requirements, which maybe not all platforms can achieve by default. For example, the standard Java platform is known to carry important security limitations that hinder the implementation of safe multitenant environments [33]. A more straightforward option is to run each user's components on non-shared containers, which is the approach taken by GAE.In both cases scaling is implemented by instantiating/releasing containers where several replicas of the developer components can be run (i.e. it is a form of horizontal scaling). This should automatically be done by the platform, so developers are not forced to constantly monitor their services' state. Either automatic scaling IaaS systems (such as those mentioned in Section 2) can be used, or the platform itself can scale up and down the resources. The latter approach is the one used by both AppEngine and Aneka. AppEngine can run in any Xen based cloud, private (built for example with Eucalyptus [36]) or public (such as EC2). However, it is not clear from [32] if AppEngine supports transparent cloud federation so that a private PaaS cloud could deploy its containers in VMs running in third party-owned IaaS clouds to face sudden load peaks. In any case AppScale could apply Eucalyptus ability to be integrated with other clouds. Aneka, on the other hand, supports cloud federation natively, so federated Aneka clouds can borrow resources among them, or a given Aneka cloud can use containers hosted in VMs running in different IaaS clouds. Automatic scaling of containers has several implications for developers regarding component design. If the PaaS platform can stop container replicas at any moment (for example due to low load), components could be designed to be as stateless as possible (as GAE recommends for the applications it hosts). In order to ease application development, support for stateful components should be offered by the platform. In this case with stateful components, LB and container replica management modules must be aware of state. This way, LBs know which container replicas hold session data to forward requests accordingly; and container instances are not removed as long as they hold some session. If more than one container instance holds data from the same session (for better scalability and failure resilience), then some mechanism is necessary that allows to keep the different data replicas updated. Transparent session data (e.g. "shopping cart") replication usually denoted *soft state replication* can be offered through systems such as Tempest [35] tool or SSM [38]. It is also possible to use distributed cache systems such as memcached [37] for explicit data sharing among component replicas. Each solution will have a different impact on component development. Roughly speaking, distributed caches work at application level, i.e. they are explicitly accessed by the hosted components code to store/retrieve the information shared among component replicas, while soft state/session replication systems work in a transparent manner for the application developer.

3.2 Database Scalability

The abundance of literature on database scalability is huge, but only the most important points for PaaS databases are highlighted here. PaaS systems must expect very high requests rates as they can host many applications that demand intense data traffic. Three mechanisms can be used (and combined) to handle this: distributed caching, NoSQL databases, and database clustering. Caching systems, such as memcached [37], are used to store intermediate data to speed up frequent data queries. A request for some data item will first check the cache to see if the data is present, and will query the database only if the data is not in the cache (or it has expired).

Distributed caching systems provide the same cache across several nodes, which is useful in clustered applications. For example, GAE offers memcache6 as a service for application developers. The term "*NoSQL*" refers to a wide family of storage solutions for structured data that are different from the traditional relational, fully SQL-compliant, databases [40]. NoSQL systems offer high scalability and availability, which seems a good fit in cloud environments with potentially many applications hosted under high demand. On the other hand, the replica management mechanisms they use provide less guarantees than traditional systems. Usually, updates on data copies are not immediately done after each write operation, they will be "eventually" done at some point in the future. This causes these systems to be unable to implement support for transparent and fully ACID compliant transactions, hence imposing some limitations on how transactions can be used by developers. Besides, the fact that they only support (the equivalent to) a subset of SQL can be a hurdle for some applications. An example is BigTable [39], which is used by GAE to provide its object-oriented data storage service. HBase7, an open source implementation of BigTable, is used by AppEngine.Finally, if fully relational and SQL-compliant databases are to be provided (as in the case of Microsoft's Azure), clusters can be built to provide better scalability, availability and fault tolerance to typical DBMS systems. Unfortunately, these clusters must be built so several or all nodes contain a replica of each data item, which is known to compromise performance even for moderate loads when transactions are supported [42]. Present database replication systems from every major relational DBMS have several limitations, and further research is needed to achieve the desired performance [41]. The major problem comes from the fact that transactions require protecting the data involved while the transaction lasts, often making that data unavailable to other transactions. The more transactions running at the same time, the more conflicts will be raised with the corresponding impact on the application performance.Yet, some database replication solutions exist that offer some degree of scalability. These can be part of the DBMS itself (in-core) or be implemented as a middleware layer between the database and the application. Most of the middleware based solutions use a proxy driver used by client applications to access the database. This proxy redirects requests to the replication middleware, which forwards them to the database. The middleware layer handles requests and transforms then in database operations to ensure that all data copies are updated. Also, it takes care of load balancing tasks. Examples of such solution are C-JDBC, Middle-R and DBFarm It can be concluded from this section that replication of databases/components is the most important issue to consider when scaling a PaaS platform. Accordingly, Figure 4 sketches the main replication ideas presented in this section. At container level, the same component can be run on different container instances to achieve better scalability and failure tolerance. But then the platform should make available some mechanism for consistent data replication among components. At database level, copies of the application data can be hosted in different DBMS instances again for better scalability and failure tolerance. Unfortunately, keeping consistency can lead to transaction conflicts.

Table 1 sums up the most relevant works related to holistic application scaling in cloud environments at the three different levels discussed in this short review: server, network and platform level. The most relevant features are highlighted and appropriate references are given for readers' convenience.

Table 1. Works Related with Scalability at Different Levels

RVER LEVEL	
Automatic VM Scaling [3,10]	Services that scale single VMs horizontally depending on a set of predefined, fixed and VM-related performance metrics.
Dynamic Workload-Pattern Matching [14]	Nimbus scaled out by federating on top of Amazon's (deploying new VMs adhered to the cluster on separate infrastructures as in the row above). However, Nimbus included a new technique: it dynamically adapted to a variety of job submission patterns, which resulted in further scalability.
Whole Application Scaling [19, 11, 12]	Mechanisms to express the scaling features of the whole service are provided by these systems. Complex rules are available based on service performance metrics that relate measurements of different VMs or different tiers to control scalability.
Non-scalable Load Balancing	Amazon offers Load Balancers to distribute load among your created VM replicas. However, this system does not offer any mechanism to scale load balancer themselves
DNS-based Load Balancing	DNS load balancing seems to be a reasonable approach in a public cloud where every VM receives a public IP. What is the away to go in private or hybrid clouds in which application components can be placed in public and private clouds
NETWORK LEVEL	
On-demand Creation of Virtual Network Resources [27, 18] Network slicing, [30, 18, 32,29]	An architecture and proof of concept system are available that "instantiate" bandwidth-provisioned network resources together with the VMs composing the service across several cloud providers Keep separate per application flows by adapting to on demand network utilization needs by every application, dynamic network bandwidth allocation
PLATFORM LEVEL	
App Scale [31]	Platforms will require container replicas to be deployed or released dynamically to handle load variations. AppScale can scale the VMs used to host containers depending on actual application demand, automatically configuring the load balancers.
Aneka [34]	For high loads, and to avoid over provisioning of resources, it would be useful to be able to federate clouds so components can be run in external/public clouds if needed. Aneka is able to deploy containers and run users applications in several IaaS providers.
Tempest/SSM [35]	"Soft state" in its title refers to data that does not need to be permanently stored, such as user session data. Replication of soft state data makes such data available to all application replicas so that everyone can attend user requests.
Automatic Session Replication	Some container implementations can use soft state replication solutions or their own replication system for automatic replication of users sessions. These solutions work at container level and are transparent to the application developer.
Memcached [37]	Distributed cache systems, such as memcached, offer a key/value distributed storage system that can be used to reduce database access requests. The values stored are available to all application replicas; so distributed caches can be used to share state information among those replicas in an explicit manner.
Big Table/H Base [39]	Traditional fully SQL and ACID compliant DBMSs have limited scalability. Recent DBMSs are rather oriented to high availability and scalability, although they can relax some ACID conditions and do not fully implement SQL. This approach can be more suitable in cloud platforms.
C-JDBC/Middle-R/DBFarm [51, 52, 53]	If fully SQL and ACID compliance is a requirement, then an option to increase the scalability is to combine several DBMS to manage replicas of all or part of each database. C-JDBC and Middle-R provide a middleware layer that allows to combine DBMS in a flexible manner.

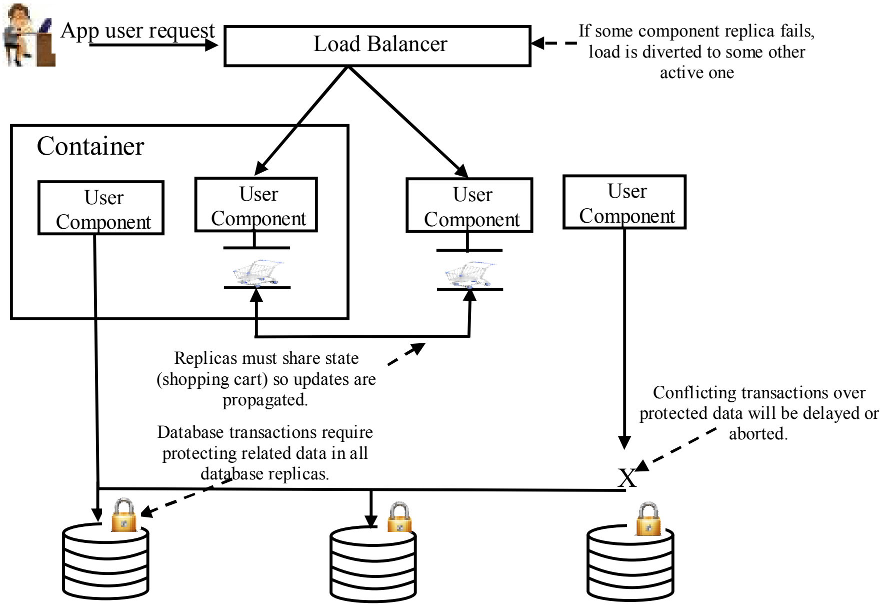

App user request
Load Balancer
If some component replica fails, load is diverted to some other active one
Container
User Component
User Component
User Component
User Component
Replicas must share state (shopping cart) so updates are propagated.
Database transactions require protecting related data in all database replicas.
Conflicting transactions over protected data will be delayed or aborted.
X

Fig. 4. Data Replication in PaaS

4 Privacy Architectural Options

In this section we describe different possible architectures for privacy management within cloud computing, and demonstrate how trusted computing can be used to strengthen this approach. The most appropriate architecture to be used depends upon the cloud infrastructure deployed for a particular environment, and the trust relationships between the parties involved.

4.1 Privacy Manager in the Client

Privacy Manager software on the client helps users to protect their privacy when accessing cloud services. A central feature of the Privacy Manager is that it can provide an obfuscation and de-obfuscation service, to reduce the amount of sensitive information held within the cloud. In addition, the Privacy Manager allows the user to express privacy preferences about the treatment of their personal information, including the degree and type of obfuscation used. Personae – in the form of icons that correspond to sets of privacy preferences – can be used to simplify this process and make it more intuitive to the user. So for example, there could be an icon with a mask over a face that corresponds to maximal privacy settings, and other icons that relate to a lower level of protection of certain types of personal data in a given context. The user's personae will be defined by the cloud service interaction context. The user may define personae, although a range of default options would be available.

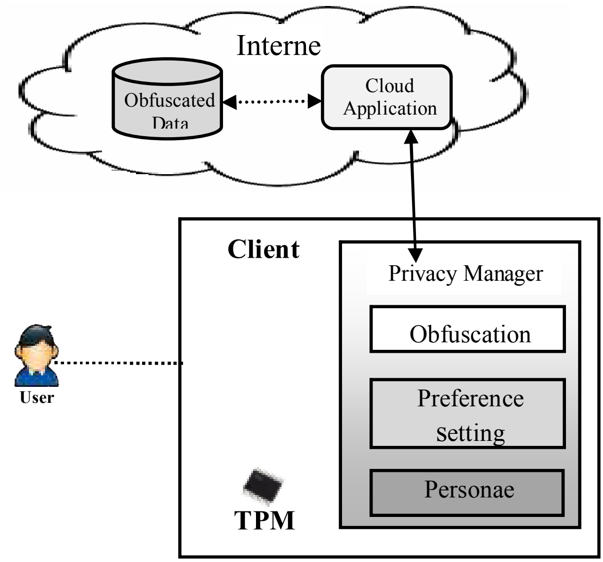


Fig. 5. Client-Based Privacy Manager

Trusted computing solutions, like those being developed by the Trusted Computing Group (TCG) [45], can address the lower-level protection of data, and this can be exploited in our solution. The TCG is an organization set up to design and develop specifications for computing platforms that create a foundation of trust for software processes, based on a small amount of extra hardware called a Trusted Platform Module (TPM). This tamper-resistant hardware component within a machine acts as a root of trust. In the longer term, as specified by TCG, trusted computing will provide cryptographic functionality, hardware-based protected storage of secrets, platform attestation and mechanisms for secure boot and integrity checking [44]. Allied protected computing environments under development by certain manufacturers and open source operating systems such as Linux can support TCG facilities further. For details about how trusted computing might be used to enhance privacy, see [48].

As an enhancement to our solution, a TPM on the client machine can be used to protect the obfuscation keys and provide further benefits. The privacy manager software and the methods linking sensitive data to pseudonyms can be protected by a TPM (see Figure 5). The TPM can provide encryption services and also allow integrity checking of the Privacy Manager software. In general, the main benefits that trusted computing could provide for client-based privacy management are hardware based cryptographic functionality, confidentiality and integrity. In terms of confidentiality, it decreases the risk of unsecured access to secret material, by means of tamper-resistant hardware-based protection of keys. Moreover, protected data on the platform is not usable by other platforms. Trusted computing could yield greater trust in integrity of the privacy management software, integrity of the involved platforms, and platform identities.

4.2 Privacy Manager in a Hybrid Cloud

As an alternative, as illustrated in Figure 6, the Privacy Manager may be deployed in a local network, or a private cloud, to protect information relating to multiple parties. This would be suitable in environments, such as enterprise environments, where local protection of information is controlled in an adequate manner and its principal use would be to control personal information passing to a public cloud. The Privacy Manager can itself be virtualized within the internal cloud. Note that the TPM could also be virtualized, within the private cloud.

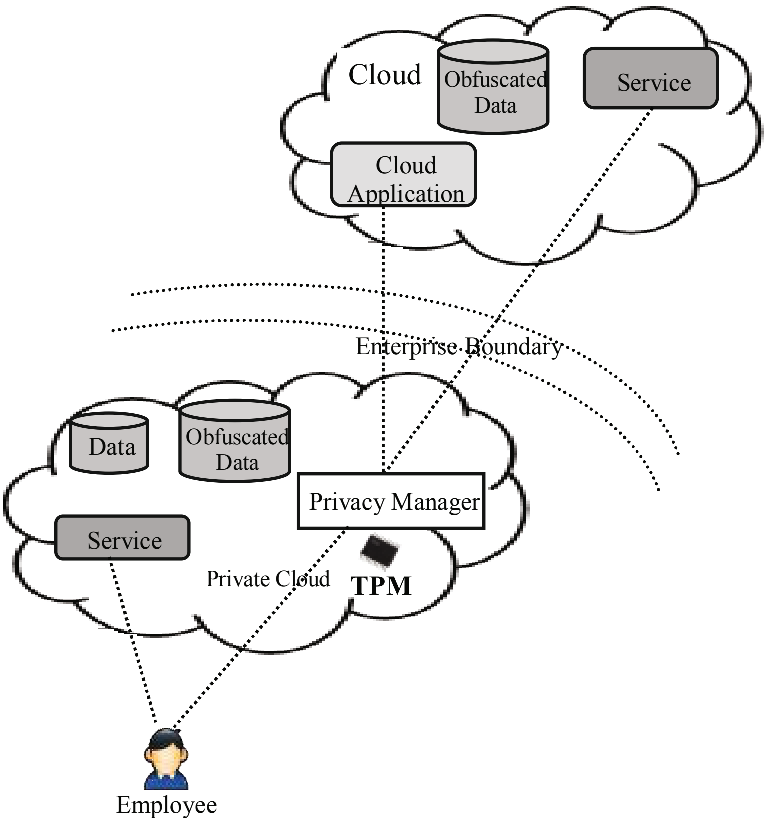


Fig. 6. Enterprise-focused Privacy Manager

Advantages to this approach include that the benefits of the cloud can be reaped within the private cloud, including the most efficient provision of the Privacy Manager functionality. It can provide enterprise control over dissemination of sensitive information, and local compliance. A significant issue however is scalability, in the sense that the Privacy Manager might slow down traffic, provide a bottleneck and may not be able to adequately manage information exposed between composed services. There are various different options with respect to this type of architecture. For example, the proxy capability could be combined, even in a distributed way, with other functionalities, including identity management. Another example is that trusted virtual machines [45] could be used within the privacy cloud to support strong enforcement of integrity and security policy controls over a virtual entity (a guest operating system or virtual appliance running on a virtualized platform). It would be possible to define within the Privacy Manager different personae corresponding to different groups of cloud services, using different virtualized environments on each end user device. In this way, virtualization is used to push control from the cloud back to the client platform. As with the previous architecture, there could be mutual attestation of the platforms, including integrity checking.

4.3 Privacy Infomediary within the Cloud

Figure 7 shows how the Privacy Manager may be deployed as (part of) a privacy infomediary [50], mediating data transfer between different trust domains. The Privacy Manager would act on behalf of the user and decide the degree of data transfer allowed, based upon transferred user policies and the service context, and preferably also an assessment of the trustworthiness of the service provision environment. Notification and feedback by the Privacy Manager to the user would also be preferable here, in order to increase transparency and accountability. The infomediary could be a consumer organization or other entity that is trusted by the users. It might alternatively be an entity that already exists within the cloud in order to provide an alternative

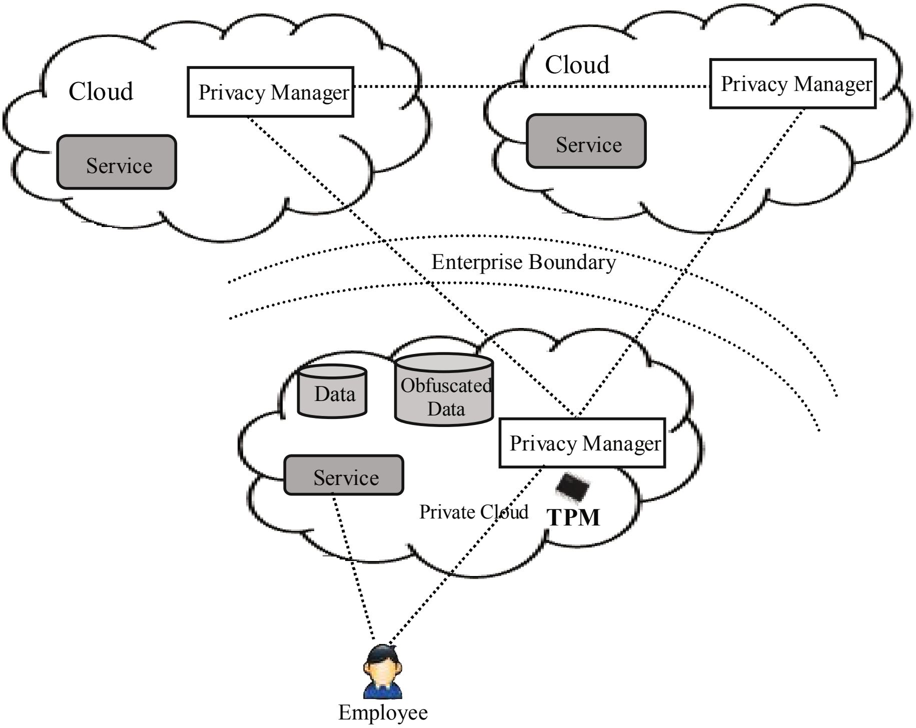


Fig. 7. Privacy Manager within the Cloud

function, such as an identity provider or auditor, and the functionality could be an extension of that. For example, the open source project Otemba [49] implements key management and user management, separating cryptographic keys from the cloud infrastructure. A key management role might be extended to a general infomediary role. The infomediary may also play a role in checking that the user preferences are satisfied before providing a decryption key for decrypting any data that needs to be decrypted in order for the cloud service to be provided (for example, it could be a Trust Authority in order to provide IBE decryption keys [43, 46].

Again, trusted infrastructure [44] could be useful in ensuring that the infrastructural building blocks of the cloud are secure, trustworthy and compliant with security best practice.

5 Conclusion

There have been great advances towards automatically managing collections of interrelated and context-dependent VMs (i.e. a service) in a holistic manner by using policies and rules. The degree of resource management, the bonding to the underlying API and coordinating resources spread across several clouds in a seamless manner while maintaining the performance objectives are major concerns that deserve further study. Also, dynamically scaling LBs and its effects on whole application scalability are yet to be reported. The reported works on network scalability are also scarce.

Some experimental reports have shed some light on possible ways to go, but there is no known production-ready system that fully accomplishes the need for dynamically managing the network in synchrony with VM provisioning. Also, such approaches will have to convince carriers, which are very reluctant to introduce innovations in production networks since they can damage the provided service when taken to such a demanding production environment. On the other hand, to achieve proper PaaS scalability cloud providers must address issues both at container and database level. Multitenant containers could be used to save resources, but this implies unsolved security concerns, while non-shared containers will demand more resources (and so imply more operation costs for the cloud provider). Replication of components and databases can be also applied for better scalability. However, replication often brings strong performance penalties that must be taken into account. How to achieve replication without incurring in high performance degradation is an open research topic. A PaaS platform should ideally give access to traditional relational databases with support for ACID transactions. But as more replicas are created to attend an increasing demand, the overload necessary to keep consistency will induce delays on requests. The ideal PaaS cloud must balance the need for powerful programming abstractions that ease developers tasks with the support for transparent scalability, and we have described Privacy Architectural options for cloud.

References

[1] Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M.: A break in the clouds: towards a cloud definition. SIGCOMM Comput. Commun. Rev. 39(1), 50–55 (2009)

[2] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst. 25(6), 599–616 (2009)

[3] Righscale web site (May 2010), http://www.rightscale.com
[4] Owens, D.: Securing elasticity in the cloud. Queue 8(5), 10–16 (2010)
[5] Sun cloud web site (May 2010), http://kenai.com/projects/suncloudapis
[6] vcloud api programming guide. (May 2010), http://communities.vmware.com/static/vcloudapi/vCloudAPIProgrammingGuidev0.8.pdf
[7] Varia, J.: Amazon white paper on cloud architectures (September 2008), http://aws.typepad.com/aws/2008/07/white-paper-on.html
[8] Gogrid web site (May 2010), http://www.gogrid.com
[9] Galán, F., Sampaio, A., Rodero-Merino, L., Loy, I., Gil, V., Vaquero, L.M.: Service specification in cloud environments based on extensions to open standards. In: COMSWARE 2009: Proceedings of the Fourth International ICST Conference on Communication System Software and Middleware, pp. 1–12. ACM, New York (2009)
[10] Amazon auto scaling service (August 2010), http://aws.amazon.com/autoscaling/
[11] Lim, H.C., Babu, S., Chase, J.S.: Automated control for elastic storage. In: ICAC 2010, pp. 19–24. ACM, New York (2010)
[12] Rodero-Merino, L., Vaquero, L., Gil, V., Galán, F., Fontán, J., Montero, R., Llorente, I.: From infrastructure delivery to service management in clouds. Future Generation Computer Systems 26, 1226–1240 (2010)
[13] Berger, E., Browne, J.C.: Scalable load distribution and load balancing for dynamic parallel programs. In: IWCBC 1999: Proceedings of the International Workshop on Cluster-Based Computing 1999, Rhodes/Greece (1999)
[14] Marshall, P., Keahey, K., Freeman, T.: Elastic site: Using clouds to elastically extend site resources. In: IEEE International Symposium on Cluster Computing and the Grid, pp. 43–52 (2010)
[15] Wu, H., Kemme, B.: A Unified Framework for Load Distribution and Fault-Tolerance of Application Servers. In: Sips, H., Epema, D., Lin, H.-X. (eds.) Euro-Par 2009. LNCS, vol. 5704, pp. 178–190. Springer, Heidelberg (2009)
[16] Olivier, S., Prins, J.: Scalable dynamic load balancing using up. In: ICPP 2008: Proceedings of the 2008 37th International Conference on Parallel Processing, pp. 123–131. IEEE Computer Society, Washington, DC, USA (2008)
[17] Youseff, L., Butrico, M., da Silva, D.: Toward a unified ontology of cloud computing. In: Proceedings of the Grid Computing Environments Workshop, Austin, Texas, USA, pp. 1–10 (November 2008)
[18] Baldine, I., Xin, Y., Evans, D., Heerman, C., Chase, J., Marupadi, V., Yumerefendi, A.: The missing link: Putting the network in networked cloud computing. In: ICVCI 2009: International Conference on the Virtual Computing Initiative (2009)
[19] Buyya, R., Ranjan, R., Calheiros, R.N.: InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services. In: Hsu, C.-H., Yang, L.T., Park, J.H., Yeo, S.-S. (eds.) ICA3PP 2010. LNCS, vol. 6081, pp. 13–31. Springer, Heidelberg (2010)
[20] Lenk, A., Klems, M., Nimis, J., Tai, S., Sandholm, T.: What's inside the cloud? An architectural map of the cloud landscape. In: ICSE 2009: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, Vancouver, Canada, pp. 23–31 (May 2009)
[21] Berger, B., Berger, E., Browne, J.C.: Scalable load distribution and load balancing for dynamic parallel programs. In: Proceedings of the International Workshop on Cluster-Based Computing 1999, Rhodes/Greece (1999)

[22] Lim, H.C., Babu, S., Chase, J.S., Parekh, S.S.: Automated control in cloud computing: challenges and opportunities. In: Proceedings of the 1st Workshop on Automated Control for Data Centers and Clouds, pp. 13–18. ACM, New York (2009)

[23] Wee, S., Liu, H.: Client-side load balancer using cloud. In: SAC 2010: Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 399–405. ACM, New York (2010)

[24] Liu, H., Wee, S.: Web Server Farm in the Cloud: Performance Evaluation and Dynamic Architecture. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) CloudCom 2009. LNCS, vol. 5931, pp. 369–380. Springer, Heidelberg (2009)

[25] Bavier, A., Feamster, N., Huang, M., Peterson, L., Rexford, J.: In vini verities: realistic and controlled network experimentation. In: SIGCOMM 2006:Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 3–14. ACM, New York (2006)

[26] Jiang, X., Xu, D.: VIOLIN: Virtual Internetworking on Overlay Infrastructure. In: Cao, J., Yang, L.T., Guo, M., Lau, F. (eds.) ISPA 2004. LNCS, vol. 3358, pp. 937–946. Springer, Heidelberg (2004)

[27] Keshariya, M., Hunt, R.: A new architecture for performance-based policy management in heterogeneous wireless networks. In: Mobility 2008 Proceedings of the International Conference on Mobile Technology, Applications, and Systems, pp. 1–6. ACM, New York (2008)

[28] Alex, T.W., Shenoy, P., Merwe, J.V.: The case for enterprise-ready virtual private clouds. In: HotCloud 2009: Proceedings of the Workshop on Hot Topics in Cloud Computing, pp. 1–5 (2009)

[29] Raghavan, B., Vishwanath, K., Ramabhadran, S., Yocum, K., Snoeren, A.C.: Cloud control with distributed rate limiting. In: SIGCOMM 2007: Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 337–348. ACM, New York (2007)

[30] Sherwood, R., Chan, M., Covington, A., Gibb, G., Flajslik, M., Handigol, N., Huang, T.-Y., Kazemian, P., Kobayashi, M., Naous, J., Seetharaman, S., Underhill, D., Yabe, T., Yap, K.-K., Yiakoumis, Y., Zeng, H., Appenzeller, G., Johari, R., McKeown, N., Parulkar, G.: Carving research slices out of your production networks with open flow. SIGCOMM Comput. Commun. Rev. 40(1), 129–130 (2010)

[31] Chohan, N., Bunch, C., Pang, S., Krintz, C., Mostafa, N., Soman, S., Wolski, R.: AppScale design and implementation. UCBS, Tech. Rep. 2009-02 (2009)

[32] Motiwala, M., Elmore, M., Feamster, N., Vempala, S.: Path splicing. In: SIGCOMM 2008: Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication, pp. 27–38. ACM, New York (2008)

[33] Herzog, A., Shahmehri, N.: Problems Running Untrusted Services as Java Threads. IFIP International Federation for Information Processing, vol. 177, pp. 19–32. Springer, Heidelberg (2005)

[34] Vecchiola, C., Chu, X., Buyya, R.: Aneka: A Software Platform for. NET-based Cloud Computing

[35] Marian, T., Bal Krishnan, M., Birman, K., van Renesse, R.: Tempest: Soft state replication in the service tier. In: DSN 2008: Proceedings of the 38th International Conference on Dependable Systems and Networks, Anchorage, Alaska, pp. 227–236 (June 2008)

[36] Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The eucalyptus open-source cloud-computing system. In: CCGRID 2009: Proceedings of 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, Shanghai, China, pp. 124–131 (May 2009)

[37] Petrovic, J.: Using memcached for data distribution in industrial environment. In: ICONS 2008: Proceedings of the 3rd International Conference on Systems, Cancún, México, pp. 368–372 (April 2008)

[38] Ling, B.C., Kiciman, E., Fox, A.: Session state: beyond soft state. In: NSDI 2004: Proceedings of the 1st Symposium on Networked Systems Design and Implementation, San Francisco, California, USA, pp. 295–308 (March 2004)

[39] Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E.: Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems 22 (June 2008)

[40] Leavitt, N.: Will NoSQL databases live up to their promise? Computer 43, 12–14 (2010)

[41] Cecchet, E., Candea, G., Ailamaki, A.: Middleware-based database replication: the gaps between theory and practice. In: SIGMOD 2008: Proceedings of the 28th ACM International Conference on Management of Data, Vancouver, Canada, pp. 739–752 (June 2008)

[42] Gray, J., Helland, P., O'Neil, P., Shasha, D.: The dangers of replication and a solution. In: SIGMOD 1996: Proceedings of the 16th ACM International Conference on Management of Data, Montreal, Quebec, Canada, pp. 173–182 (June 1996)

[43] Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)

[44] Pearson, S. (ed.): Trusted Computing Platforms. Prentice Hall (2002)

[45] Trusted Computing Group: Trusted Platform Module (TPM) Specifications. Web site, https://www.trustedcomputinggroup.org/specs/TPM/ (2009)

[46] Casassa Mont, M., Pearson, S., Bramhall, P.: Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services. In: IEEE Workshop on Data and Expert Systems Applications, pp. 377–382. IEEE Computer Society Press, Washington DC (2003)

[47] Dalton, C., Plaquin, D., Weidner, W., Kuhlmann, D., Balacheff, B., Brown, R.: Trusted virtual platforms: a key enabler for converged client devices. ACM SIGOPS Operating Systems Review 43(1), 36–43 (2009)

[48] Pearson, S.: Trusted Computing: Strengths, Weaknesses and Further Opportunities for Enhancing Privacy. In: Herrmann, P., Issarny, V., Shiu, S.C.K. (eds.) iTrust 2005. LNCS, vol. 3477, pp. 305–320. Springer, Heidelberg (2005)

[49] Otemba project: The Reasons for Otemba's Existence, http://sourceforge.net/apps/trac/otemba/wiki/Reasons%20for%20existence

[50] Gritzalis, D., Moulinos, K., Kostis, K.: A Privacy-Enhancing e-Business Model Based on Infomediaries. In: Gorodetski, V.I., Skormin, V.A., Popyack, L.J. (eds.) MMM-ACNS 2001. LNCS, vol. 2052, pp. 72–83. Springer, Heidelberg (2001)

[51] Julie, E.C., Marguerite, J., Zwaenepoel, W.: C-jdbc: Flexible database clustering middleware. In: USENIX 2004: Proceedings of the USENIX 2004 Annual Technical Conference, pp. 9–18 (June 2004)

[52] Milan-Franco, J.M., Jiménez-Peris, R., Patiño-Martínez, M., Kemme, B.: Adaptive Middleware for Data Replication. In: Jacobsen, H.-A. (ed.) Middleware 2004. LNCS, vol. 3231, pp. 175–194. Springer, Heidelberg (2004)

[53] Plattner, C., Alonso, G., Özsu, M.T.: DBFarm: A Scalable Cluster for Multiple Databases. In: van Steen, M., Henning, M. (eds.) Middleware 2006. LNCS, vol. 4290, pp. 180–200. Springer, Heidelberg (2006)

A Recommendation Model for Handling Dynamics in User Profile

Chhavi Rana[1,*] and Sanjay Kumar Jain[2]

[1] University Institute of Engineering and Technology, MDUniversity, Rohtak, 124001, India
chhavi1jan@yahoo.com
[2] Department of Computer Science Engineering, National Institute of Technology, Kurukshetra, 136119, India
skj_nith@yahoo.com

Abstract. Recommender System has become the necessary agent for a naive user in the information bombardment arena of World Wide Web. In the last decade, World Wide Web emerged as an all encompassing technology that is revolutionizing the way people live. With the passage of time, user behaviors evolve and as such should be the recommendations provided. There exists a wide gap in literature to cater effectively with the issue of temporal evolution of data on the internet. This paper will specifically analyze various ways through which temporal issue can be handled in generating user profile that evolve with time. It will develop a recommendation model to handle the dynamics in user profile. The prime focal point is to examine if recommendation accuracy can be improved by adding temporal dimension. An empirical study is carried out to compare the analysis of the traditional data mining tasks and the proposed method.

Keywords: Recommender Systems, User profile, Temporal, Collaborative filtering, Web Mining, personalization, navigation patterns, web logs.

1 Introduction

Recommender systems are tools that give personalized information to a particular user based on his profile that is generated by his or her navigational pattern and related data. This information is presented in the form of recommendation list that user can choose to browse on the web. The basic methodology behind this approach is multi-disciplinary, involving data mining and machine learning techniques. Using the content information of the website and the web usage data, mining techniques can be used to build user profile that will predict user preferences.

Currently recommendations are provided using previous browsing patterns [1] assuming that user's behavior does not change rapidly and earlier recorded annotations can help in predicting potential behavior. This assumption is valid only to some extent. In fact, the user's general interests can be relatively stable, but they can also be influenced by many additional and varying factors prominent of which is time.

[*] Corresponding author.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 231–241, 2012.

For example, imagine a user who is journalist and has been promptly given a new task in hand which requires him to learn a new skill set. This has been happening everywhere in the corporate sector. People in work sphere have been changing there profiles trying to do something new. As such, the requirements of the user will change overtime while browsing internet also. In this paper, we will investigate how this additional factor that is time could be exploited to improve the accuracy of a recommender system. The main motivation behind such a system is to support the re-design process of such user profiles with time and develop a recommendation model that will more accurately predict user behavior that evolves with time.

Taking temporal evolution of data into account has been a widely researched topic where traditional temporal databases are concerned. In the field of recommender system, handling temporal dynamics and their effect on collaborative filtering accuracy has been recently studied by Koren [7]. He empirically shows that using the concept drift schemas on temporal dimension, the accuracy of recommender systems can be increased after a time. In this paper, we explore the existing work in the area of frequent pattern mining and incremental clustering and take the benefits of using it for improving the quality of the generated recommendations. Using time as a factor to update the user model looks very relevant in the fast changing world and it can be used to extend a system in many ways. This paper focuses on two main objectives. First, with time how web contents are getting modified that are relevant to user; second, how user interests are also changing with time and how this new dimension can be added to his or her profile.

In the rest of this paper, Section 2 will introduce our proposed recommendation model for handling dynamics in user's profiles. Section 3 describes related work in this area. Section 4 presents the empirical evaluation of the proposed method. The future work is discussed in Section 5 and in Section 6, conclusions are presented.

2 Recommendation Model

This paper presents a recommendation model for an effective web personalization system by integrating web usage and web content mining and making it more effective, efficient and dynamic. It will take use of temporal features of the items like year, month, date, time, popularity and freshness to discard the outdated items while discovering frequently patterns as well as change prediction with evolving user profiles that are separately updated through incremental clustering. The recommendation model intend to use a hybrid method involving mining of web usage data and web content data to predict more accurate recommendations. As noted by Mobasher in [10], "usage-based recommendations can be challenging either when there is not enough usage data in order to mine patterns related to certain categories, or when the site content changes and new pages are added but are not yet included in the web log. The integration of data related to the content of the website items provides a way of overcoming such problems, thus improving the whole recommendation process. Thus, the need to associate web usage and content knowledge, by enhancing the information in the Web usage logs with semantics derived from the content of the Web site's pages".

Web usage mining together with Web content mining is a very promising solution that can help in producing personalized recommendations, making access to on-line information more efficient. This issue is becoming crucial as the size of the Web increases at breathtaking rates. Mobasher [10] has researched widely on integrated web usage and content mining technique for more effective personalization, but have not incorporated the temporal dimension in evolving user profile.

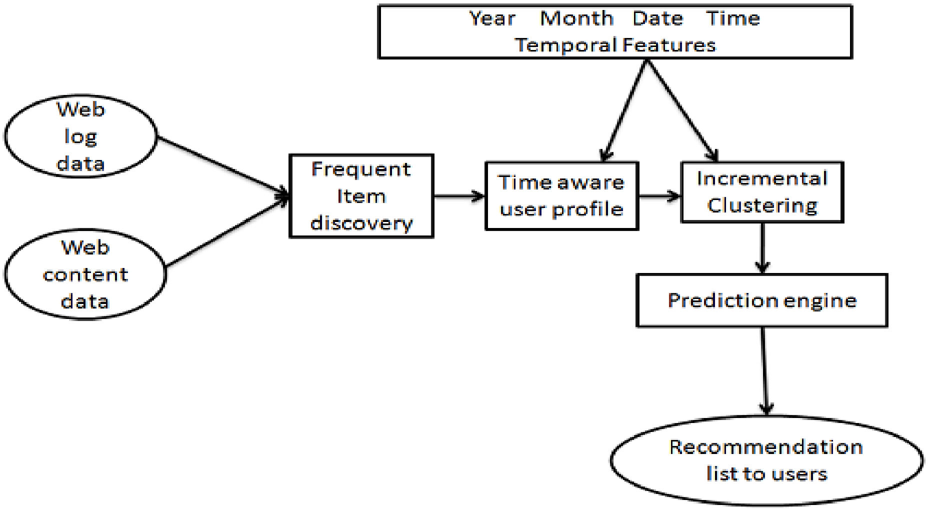


Fig. 1. Recommendation Model

In figure 1 the proposed recommendation model is depicted. Here, the rectangular boxes represent system components and each arrow indicates information flow. For example, the temporal Feature is a module that tracks changes of variables with time. It contacts appropriate web content features as well features in user profile and stores all temporal states in a local database. When the usage data and content data is analyzed by the frequent item discovery module, temporal feature module provides information about known temporal variables for a user or an item of a specific time point. The temporal feature module performs all the reasoning related to the variables that changes with time. It determines if a temporal variable is important for a prediction, removes noisy context data and makes predictions for missing temporal variables. The Time aware User profile module and Incremental clustering components represent generating a user profile taking into account the temporal features and incrementing as well as updating these profiles through incremental clustering respectively in the system.

The user is modeled with his preferences and in time aware user profile it is represented as a vector of item ratings with another added dimension that is time. The user profiling model captures user behavior at various intervals of time as determined by the application domain. Like Movies are released every Friday, so we can have a weekly updating module in case of movie recommender system, for online shopping forum winter and summer sales are specific time periods to watch out the behavior round the year. As such relevant knowledge according to application domain needs to be fed in the system through a temporal module. The time aware user profile module is responsible for integrating temporal data into the prediction algorithm. It takes information provided by the temporal feature module and enhances the representation of the user/item model with intervals of time. The time aware user profiling is enhanced by providing the timely updates provided by incremental clustering reasoned to find out needed information to motivate a recommendation because of the particular temporal conditions. Incremental clustering module will update the user profile, provides the updates for each of them and takes the data to the prediction engine for future recommendations. It closely collaborates with the prediction engine to identify the best recommendations. The prediction engine finally takes the enhanced data model and generates a list of rating predictions. Recommender systems can have several model adapters and prediction engines, which can generate different lists of predictions. A recommender module can then takes all the produced recommendation lists and combines them into a final recommendation list. It can use information obtained from the temporal feature manager to filter out, or change the ratings for the final list. A user as output gets a list of recommendations with possible timely updates.

The purpose of this recommendation model is to make contributions in improving the overall quality of Web based Information Systems, to support designers during the design process and to ensure ease of use to end users. In the development of this model, a study of the various web-mining techniques for personalization and recommender system was done. After this study and analyzing a wide gap in taking up temporal feature into account to improve the accuracy of recommender system except recent work done by [7] , this gap will be dealt with by suggesting a different approach. The paper will attempt to suggest an improvement by introducing a new model consisting of improved techniques and algorithms that will deal with the current drawbacks and forthcoming challenges to deal with temporal evolution data on the web. A novel recommendation model has been chalked out that could optimize the concurrent techniques and implement new techniques designed to have more effective web personalization experience for the user.

The proposed recommendation model focuses on techniques that incorporate temporal information into the Collaborative Filtering recommendation process. The paper investigates various frequent pattern mining and incremental clustering techniques suitable for Collaborative Filtering. Referring to the general recommendation model presented in Figure 1, both time aware user profiling and Incremental clustering are examples of implementation of the temporal feature adapter module and the resulting data is fed into the prediction engine component. The present recommendation model have analyzed two possibilities to incorporate temporal features in the CF. One selects frequent pattern through mining, i.e., items whose ratings are used in the user model and are used to compute recommendations. The second refers to the module that uses incremental clustering to update the time aware user model and make prediction of items suggested in the final recommendation.

2.1 Time Aware Frequent Pattern Mining

Frequent pattern mining has become one of the most activated researched topics in data mining and knowledge discovery in databases in the last decade. Among them, Apriori, Eclat, and FP-growth are most commonly used. The preliminary problem was mining transactional data, which depict the shopping behavior of customers in the supermarkets, mail-order companies and online shops, for products that are frequently bought together. Market based analysis was the initial task that reflected such kind of problem. For this problem, which became generally known as frequent item set mining, a large number of competent algorithms were developed, which are based on complex data structures and clever processing schemes. Various extensions of frequent pattern mining ranges from item sets to item sequences and they are fairly uncomplicated. The research in this direction has opened up inspiring new application areas, like genome mining and temporal pattern extraction from data describing, for instance, alarms occurring in weather monitoring systems.

The latest research in this field has taken up the more complex tasks of mining frequent trees and general frequent sub graphs, which have applications in biochemistry, web mining, citation network and program flow analysis [2]. Since the structure of the data introduces new problems, these tasks are more difficult and the solution of such tasks is trivial for item sets. This paper intends to enhance the basic algorithms for frequent item set mining by adding extensions to frequent sequences on the basis of temporal features. Then the core problems of mining unstructured data that evolves with time could be dealt with easily.

2.2 Incremental Clustering

Clustering is used for data analysis and prelude to classification tasks in a wide variety of applications. In the area of information retrieval it has been judged as an important tool for building taxonomy of a large amount of documents by forming groups of closely related documents. An incremental clustering should simultaneously optimize two potentially conflicting criteria: first, the clustering at any point in time should remain faithful to the current data as much as possible; and second, the clustering should not shift dramatically from one time step to the next instead the transaction ought to be smooth and precise.

The present information overload, fueled by the accessibility of data from hypermedia and the World Wide Web, has led to the creation of a large amount of data. To efficiently store and retrieve this information is one of the prominent challenges for information retrieval systems. The high rate of updates in the document databases on the World Wide Web is becoming a major problem for such systems. Several researchers have investigated that existing clustering algorithms are not appropriate for maintaining clusters in such a dynamic environment, and they have been struggling with the task of updating clusters without repeatedly performing complete reclustering [3]. As such recommender system also faces the problem of predicting user behavior when his profile as well the contents on the website are changing with time. Unless and until an approach is developed that takes into account these dynamic changes, the prediction will remain useless for the users. Thus incremental clustering can help in the updating of user profiles as this is what it is precisely meant for, dynamically updating user clusters and correspondingly user profiles which is one of the prime components of the presented recommendation model.

2.3 Time Variant User Profile

A user profile is a collection of personal data associated to a specific user. A profile refers therefore to the explicit digital representation of a person's identity. A user profile can also be considered as the computer representation of a user model. Dynamic user profile issues and design are considered in this paper. Though dynamic user profile in information retrieval has been studied long ago, their incorporation in recommendation model has not been widely researched upon. A model that incorporates user profiles for a priori query improvement has been proposed by [11] two decades earlier. The developed a model that personalizing the user query and resulting model presented two important advantages. Firstly, the model made it possible to retrieve broader range of documents, some parts of which would not be brought to the user's attention if only queries were used. Secondly, the model could successfully tailor the retrieved information to a particular user's interests and preference. As time goes, user's interest will probably be changed and thus it seems more suitable that document space should not only change but should have different values which reflect the weights of the information in user profile. Since then though a number of technique have been applied to dynamically enhance the user modeling process yet no comprehensive work is available regarding its usage in recommender system.

In this paper, a recommendation model is presented which is composed of a dedicated module in which the user profiles are updated in such a manner so that timely updations in user behavior can be added automatically with passage of time as seen through his past behavior or recent browsing patterns discovered through frequent pattern mining. Such profiles are termed as time aware user profile. Another important issue is updating of such profiles with time which is dealt with the use Incremental clustering.

3 Current State of the Research

A couple of researchers have been working on various factors including time that could improve the accuracy of recommender system. Koren [7] highlights the effect of temporal dynamics in RS in his research. He emphasizes on the need of including temporal changes in RS to improve the accuracy of recommendations. He also proposes a model that traces the time changing behavior throughout the life span of data and thus exploiting all relevant components which is in contrast with the earlier concept drift explorations where only single concept is traced. On the other hand, Lathia, Hailes, & Capra [9] provide a different perspective, which is a system oriented approach different from [7] user preference model. He studies the effect of retraining CF algorithm every week as a time dependent prediction problem and proposes an adaptive temporal CF technique. This technique temporally adapts the size of KNN neighborhood based on the performance measured up to current time. Another approach that incorporated temporal information to achieve better recommendation accuracy is proposed by Queue et al. [15]. They combined the two dimensions involving temporal dynamics, one proposed by [16] involving product launch time and other by [8] that is based on rating time, together with implicit feedback data to construct a pseudo rating data. This rating data gives recommendations that are more accurate.

A feature-based machine learning approach was proposed by Chu & park [5] to personalized recommendation that is capable of handling the cold-start issue effectively. They maintain profiles of content of interest, in which temporal characteristics of the content, e.g. popularity and freshness, are updated in real-time manner. Another recommender system named “Eigentaste 5.0: Constant-Time Adaptability Recommender System” was developed by Nathanson, Bitton, &

Goldberg [13] that dynamically adapts the order that items are recommended by integrating user clustering with item clustering and monitoring item portfolio effects. Another group of researcher, Tang et al., [16] focused on dealing with the issue of scalability by applying a different kind of technique that scale down candidate sets by considering the temporal feature of items and simultaneously increasing the accuracy of the performance of recommender systems. In addition, Pessemier et al. [14] presents an empirical evidence that older consumption data has a negative influence on the recommendation accuracy in case of consumer centric RS.

Chen et al. [4] also showed the time decaying effect of sequential pattern on the user preference within content based RS. Golbandi, Koren, & Lempel [6] proposed to use an adaptive bootstrapping process that elicits users to provide their opinions on certain carefully chosen items or categories while changing the questions with time adapting to user responses. Such a process could help in determining evolving user profile. In addition, Nasraoui et al., [12] studied the behavior of collaborative filtering based recommendations under evolving user profile scenarios and proposed a systematic validation methodology that allows for simulating various controlled user profile evolution scenarios and validating the studied recommendation strategies. Thus, by far though temporal features have been taken into account in different forms, yet a comprehensive work involving frequent pattern mining and incremental clustering to include temporal dimension in improving the accuracy of recommender system has not been done so far.

4 Experiment

4.1 Evaluation

The effectiveness of our proposed model is tested by using the existing algorithm of frequent pattern mining and incremental clustering over real life data and the performance is compared with other clustering algorithms. The testing of algorithm is done using Weka machine learning software. The experiments were run on an Intel 1.8 GHz Pentium 4 dual processor CPU with 2 GB RAM and Windows XP operating system. The implementation runs on Netbeans and Java. The evaluation are done using web data mining software and Log likelihood is taken as the performance measure.

4.2 Dataset

The experiments were carried out with real data set taken from UCI research lab site. A total of four dataset were used namely credit dataset, weather dataset, breast cancer dataset and primary tumor dataset.

All the dataset are collected from real life implementation and belong to a different application area so that we can get an overall working of our recommendation model and the related algorithms. Table 1 shows a brief description of datasets used in the experiment.

Table 1. Brief description of the data sets used in the experiments

Dataset	#instances	#attributes	Discrete	source
Credit	690	16	6	UCI (Newman et al., 1998)
Weather	14	5	2	UCI (Newman et al., 1998)
Primary tumor	339	18	no	UCI (Newman et al., 1998)
Breast cancer	286	10	no	UCI (Newman et al., 1998)

4.3 Results

The new approach with a combination of frequent pattern mining and incremental clustering is evaluated in terms of Log Likelihood measure, which is used as metrics for judging the performance of clustering algorithm. The newly developed model that uses a combination of these algorithms is compared with the two other standard Clustering approaches, which do not consider such a combination. The comparison baseline is the traditional approach which includes Densitybasedcluster and Heirarchialcluster and their comparison is carried out with our newly developed model called Frequentcluster algorithm to the whole set of items. The performance results of these experiments are presented in Table 2 showing Log Likelihood parameter for each algorithm respectively for range of datasets together with our proposed approach. The behavior of the combination of frequent pattern mining and Incremental clustering algorithm (called frequentcluster) with different learning techniques was analyzed in order to determine if multicluster could be used for recommendation model. In WEKA for this case study, the presented combination by different algorithms showed high accuracy (Table 2). Building and evaluation times of the individual algorithms are more in relation to frequentcluster (Table 2). Two algorithms (EM and filteredassociater) were used as base algorithm to build frequentcluster and this meta learning scheme is applied to evaluate the performance with individual clustering algorithm. Table 2 shows the model building and evaluation times. Its application in recommender systems must be considered if the employed time in the model building is not prolonged, since for this type of system the immediacy is one of the main factors to consider as indispensable requirement.

The dataset comprising of all the instances was divided into 80%-20% training and test set, and then the training set has been used to learn the model but the test set is used to assess the quality of the final model, i.e., to compute the log-likelihood. To judge the accuracy of the traditional Densitybasedcluster, Heirarchialcluster, and our newly developed Frequentcluster algorithm clustering algorithms, we note, within each cluster Number of cluster formed, Modeling time and Log likelihood. Thus, in the last column of each dataset of Table 2, we report the Log likelihood for each dataset obtained for each competitor algorithm, respectively. The first thing Table 2 reveals is the comparison of processing time among Frequentcluster and other traditional algorithms. At a glance, we can explicate that our proposed Frequentcluster algorithm (combination of frequent pattern mining and incremental cluster) takes less processing time than other algorithm. Original hierarchical algorithm has the highest value than other improved algorithms. However, nowadays high performance computer, super computer, etc. are available for users which lessen processing timing tremendously.

Table 2. Results obtained

Dataset	Algorithm	Modeling Time(sec)	Number of cluster found	Log likelihood
Credit	Densitybasedcluster	1.10	2	-36.6786
	Heirarchialcluster	2.12	7	-32.53145
	Frequentcluster	1.01	2	-37.2451
Weather	Densitybasedcluster	.01	2	-9.25907
	Heirarchialcluster	.02	1	-9.4063
	Frequentcluster	.01	2	-9.5154
Primary tumor	Densitybasedcluster	.04	2	-10.73842
	Heirarchialcluster	.87	6	-9.71095
	Frequentcluster	.02	2	-11.82632
Breast cancer	Densitybasedcluster	.05	2	-9.79929
	Heirarchialcluster	.17	3	-9.36546
	Frequentcluster	.01	2	-9.68314

It can be easily inferred from Table 2. that except hierarchical approach, the other two approaches depicts better Log likelihood. The difference is especially significant when average dataset is large, with relative improvement in Log likelihood between our approach and the densitybasedcluster. On the other hand, these two clustering approaches give relatively close results to each other. To be more specific, clustering based on frequent pattern mining of user profile is a little better, but the superiority is not clear. In Table 2, we find that it is only in one case (for the breast cancer data) that the classical Densitybased algorithm yields a better Log likelihood measure, as compared to the Frequentcluster. However, from evaluation table, we may note that this difference is not statistically significant. However, Frequentcluster based clustering on smaller dataset is superior in term of feasibility, since it only takes as input ratings of available data, and does not use metadata of any genre. As widely discussed, metadata are not always available, and even if available, often suffer from the problem of subjectivity of the metadata creator. However, in real applications, we can only consider opinions of a few items in dataset due to the scalability problem, and since our Approach yield significantly more accurate prediction than traditional approach when number of items is significantly large, our approach can efficiently contribute to improving the accuracy of rating prediction in real applications. In addition to these dataset, we intend to test the proposed approach with other real life datasets, such as Netflix (dataset available at netflixprize.com) and Yahoo! Movies (ratings publicly displayed at movies.yahoo.com). Currently, though the proposed technique is efficient in terms of precision, it shows some restrain in terms of scalability. We hope to present more results of our experiment in the nearest future.

5 Future Work

The temporal dimension is widely taken into account during the analysis of web data since with passage of time both the contents of the websites as well as the requirements of users gets changed. Consequently, the models associated with these parameters must be constantly updated to reflect the current situation. The problem arises when a heavily used web site with thousands of users per day needs to be investigated, and this paper precisely deals with the question that how this can be achieved with a optimal resources.

It is thought that the answer lies in the development of a flexible architecture and the development of new techniques for unsupervised and undirected knowledge discovery from web usage data, and the integration of content information and meta-data with the discovered usage patterns[10]. It is believed that in the future, web mining methods will increasingly treat content, structure, and usage in an integrated manner in iterated cycles of extracting and utilizing semantics, to be able to understand and reshape the Web. Future work will include:

How can we incorporate evolving user interest in existing user profile in the recommender system?

How can we embed the changing content of the web pages to provide improved recommendation that deal with the temporal evolution of information on the web?

To improve the developed a recommendation model using newly devised technique involving temporal component and empirical test it on real world data.

Additionally, it can also be investigated how CF hybridization deals with the recommendation component, where temporal features are used to filter out irrelevant recommendations

Thus, the future work in this direction involves dealing with all of the above listed issues. The results of such work needs to be demonstrated through extensive experiments both on synthetic and real life data and their practical significance should be discussed in detail. The core approaches to solve these problems are experimentally yet to be tested on a large dataset. An empirical study will be carried out in future to show that the proposed recommendation model for personalization involving new techniques is efficient.

6 Conclusions

In this paper, we proposed a recommendation model for time-aware recommender system. This paper presents a recommendation model to incorporate temporal dimension of data in the recommendation process. This will help in improving the accuracy of the overall recommendation process. It should be flexible enough and application oriented that could be used by the research community and directly applied to practice. In addition, the underlying algorithm behind this model is analyzed and their efficiency is evaluated on small real life datasets. Moreover, this paper summarized the work done so far and presented some issues for future work. The main goal is to understand the potential benefits of using time as a dimension and to provide an in-depth analysis of various approaches as well.

References

1. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans. Inf. Syst. 23(1), 103–145 (2005)
2. Borgelt, C.: Simple Algorithms for Frequent Item Set Mining. In: Koronacki, J., Raś, Z.W., Wierzchoń, S.T., Kacprzyk, J. (eds.) Advances in Machine Learning II. SCI, vol. 263, pp. 351–369. Springer, Heidelberg (2010)
3. Charikar, M., Chekuri, C., Feder, T., Motwani, R.: Incremental Clustering and Dynamic Information Retrieval. SIAM J. Comput. 33(6), 1417–1440 (2004)
4. Chen, T., Han, W.L., Wang, H.D., Zhou, Y.X., Xu, B., Zang, B.Y.: Content recommendation system based on private dynamic user profile. In: International Conference on Machine Learning and Cybernetics, pp. 2112–2118 (2007)
5. Chu, W., Park, S.T.: Personalized Recommendation on Dynamic Content Using Predictive Bilinear Models. In: 18th International WWW Conference, Madrid, Spain, pp. 691–706 (2009)
6. Golbandi, N., Koren, Y., Lempel, R.: Adaptive Bootstrapping of Recommender Systems Using. In: WSDM, Current, pp. 595–604 (2011)
7. Koren, Y.: Collaborative filtering with temporal dynamics. Communication of the ACM 53(4), 89–97 (2010)
8. Ding, Y., Li, X., Orlowska, M.E.: Recencybased collaborative filtering. In: Proceedings of the 17th Australasian Database Conference, vol. 49, pp. 99–107. Australian Computer Society, Inc. (2006)
9. Lathia, N., Hailes, S., Capra, L.: Temporal collaborative filtering with adaptive neighbourhoods. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR 2009, pp. 796–797 (2009)
10. Mobasher, B., Dai, H., Luo, T., Sun, Y., Zhu, J.: Integrating Web Usage and Content Mining for More Effective Personalization. In: Bauknecht, K., Madria, S.K., Pernul, G. (eds.) EC-Web 2000. LNCS, vol. 1875, pp. 165–176. Springer, Heidelberg (2000)
11. Myaeng, S., Korfhage, R.: Dynamic user profile in information retrieval. In: ACM Annual Computer Science Conference, Louisiana, United States, pp. 417–432 (1985)
12. Nasraoui, O., Cerwinske, J., Rojas, C., Gonzalez, F.: Performance of Recommendation Systems in Dynamic Streaming Environments. In: Proc. of SDM – SIAM International Conference on Data Mining, Minneapolis MI (2007)
13. Nathanson, T., Bitton, E., Goldberg, K.: Eigentaste 5.0: constant-time adaptability in a recommender system using item clustering. In: Proceedings of the First ACM Conference on Recommender Systems, New York, USA, pp. 149–152 (2007)
14. Pessemier, T.D., Dooms, S., Deryckere, T., Martens, L.: Time dependency of data quality for collaborative filtering algorithms. In: Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, pp. 281–284 (2010)
15. Queue, T., Park, Y., Park, Y.-T.: A time-based approach to effective recommender systems using implicit feedback. Expert Systems with Applications 34(4), 3055–3062 (2008)
16. Tang, T.Y., Winoto, P., Chan, K.C.C.: Scaling Down Candidate Sets Based on the Temporal Feature of Items for Improved Hybrid Recommendations. In: Mobasher, B., Anand, S.S. (eds.) ITWP 2003. LNCS (LNAI), vol. 3169, pp. 169–186. Springer, Heidelberg (2005)

Allocation of Slotted Deadline Sensitive Leases in Infrastructure Cloud

Dhairya Vora[1], Sanjay Chaudhary[1], Minal Bhise[1],
Vikas Kumar[1], and Gaurav Somani[2]

[1] DA-IICT, Gandhinagar, Gujarat, India
[2] The LNMIIT, Jaipur, Rajasthan, India
{vora_dhairya,sanjay_chaudhary,
minal_bhise,vikas_kumar}@daiict.ac.in
gaurav@lnmiit.ac.in

Abstract. Resource allocation is an important aspect in cloud computing. In Cloud Computing environment, the user can access required resources in the form of a service. The resource may be a platform, a software or infrastructure. In an IaaS (Infrastructure as a Service) Cloud, users send requests to the cloud-provider in the form of a *lease*; The cloud-provider makes a scheduling plan for leases in order to maximize the number of leases it can accommodate. A lease stores information about the required resources, including the time at which the resources are required. *Haizea* is a popular resource lease manager which handles the scheduling of such leases. An algorithm for deadline sensitive leases is presented which accepts more number of leases by dividing a lease into multiple slots and by backfilling already accommodated leases. Experimental results show that our scheduling algorithm gives better performance than existing algorithms in Haizea.

Keywords: scheduling algorithms, resource allocation, cloud, backfilling.

1 Introduction

In IaaS cloud computing environment resources are provided to the user on pay_per_use basis. The request for the resources (memory, CPU and storage) is made through SLA (Service Level Agreement) which contains the description of requirements for completion of user's task. Some of the popular open source cloud platforms are Open-Nebula[13], Eucalyptus[6] and Nimbus[2]. Amazon EC2 is one of the popular cloud which is available for public use and uses pay as you go model[1]. To adjust variable load, cloud provides scalability by allowing the user to reconfigure the resources. IaaS cloud provides resources (like memory, computation capacity and storage) to the users in the form of virtual machines [12,11]. Virtual machines are configured as per the parameters specified in SLA. Scheduling is required to decide how resources should be allocated to a particular task amongst a set of tasks for efficient scheduling and execution[14]. A scheduling algorithm inside an infrastructure cloud ascertain the placement and

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 242–252, 2012.

resource allocation to various virtual machines. The resource allocation is done in terms of leases, as done in the Haizea lease manager[7][8][9][10][13]. The goal of the scheduler, described in this paper is to maximize acceptance of leases on first come first serve basis. There are four types of leases which are supported by Haizea:

1. Immediate: Cloud user can ask for specific resources. If resources are available, then the lease is accepted else it is rejected.
2. Best effort: Cloud user can ask for specific resources which do not have any time constrain. Cloud provider can provide the resources whenever asked resources are free. These leases are never rejected.
3. Advance reservation: Cloud user can ask for specific resources at a specific time. If resources are available at the asked time, then the lease is accepted else it is rejected.
4. Deadline sensitive: Cloud user can ask for specific resources. This lease has specific startTime, endTime (deadline), duration. If cloud provider can provide resources in this time duration then the lease is accepted else it is rejected.

Once a lease is accepted, it cannot be rejected. While trying to accommodate newly arrived leases, scheduler has to verify that the already accommodated leases are still able to run successfully.

2 Related Work

Haizea provides advanced reservation and deadline sensitive type of leases along with the traditional immediate and best effort policies [13,10,8,7,9]. Haizea uses simple allocation policies for deadline sensitive leases. It tries to find out a single slot of required time between startTime and endTime of the given lease which can allocate the requested resources. If it is unable to find such a time slot, it rejects the lease. Authors in[4], have provided an optimal resource allocation strategy in cloud. Authors in [5] have worked on deadline sensitive leases and provided an improved version of scheduling algorithm which can accept more leases and reject less leases. This has been done by using swapping and backfilling. Authors in [3]have worked on haizea to provide negotiation support in the lease manager by introducing counter offers for negotiation. Our work provides an improved a scheduling algorithm which slots the total available time in order to accommodate new leases.

2.1 Default Algorithm to Handle Deadline Sensitive (DS) Leases

User can have requirement of specific amount of resources which should be given for some specific amount of time between given start time and given end time (deadline). Such requirement should be given to Haizea in the form of deadline sensitive lease. When a deadline sensitive lease comes to Haizea, first it tries to find out a single slot which can provide the required resources for required duration of time. If Haizea can find such a time slot, it accepts the newly arrived deadline sensitive lease. If it cannot find such a single slot then Haizea reschedules already accommodated deadline sensitive and best effort leases. For this, first it finds out which leases should be rescheduled and then how they should be rescheduled. Leases having start time or end time greater than or equal to the start time of newly arrived lease are considered for being rescheduled. Once these leases are fixed, Haizea adds the newly arrived deadline sensitive lease to the list. Leases of the list are sorted in non-ascending order of their slack value.

$$slackValue = \frac{(deadLine - startTime)}{durationTime} \tag{1}$$

On the basis of the slack value, Haizea tries to find out a single slot for each of the leases at a time. At the time of allocating single slot, preemption of best effort leases is also considered. At any point, if any lease misses its deadline, then the new lease is rejected and the older schedule is loaded again [8]. Improvement in allocation policy of deadline sensitive leases is given in [5] . Instead of rejecting a newly arrived deadline sensitive type of lease, scheduler tries to find out multiple slots which together can accommodate this newly arrived lease. If scheduler is still unable to accept the newly arrived lease, it tries to swap two already accommodated consecutive leases. Two consecutive leases having endTime greater than the startTime of newly arrived lease and/or having startTime less than the endTime of the newly arrived lease are swapped. Algorithm checks that resources asked by first lease are less than resources asked by the second lease and they do not violet any constrains after swapping. After swapping, if the scheduler is still unable to accept the lease then the newly arrived lease is rejected.

2.2 Lease Description

User sends the request for the resources in the format given below:

```
  <lease-request arrival="00:10:00.00">
     <lease id="1" preemptible="true">
        <nodes>
           <node-set numnodes="2">
              <res amount="50" type="CPU"/>
              <res amount="1024" type="Memory"/>
           </node-set>
        </nodes>
        <start>
           <exact time="14:00:00.00"/>
        </start>
        <duration time="02:00:00.00"/>
        <deadline time="26:00:00.00"/>
        <software>
           <disk-image id="foobar1.iso" size="1024"/>
</software></lease></lease-request>
```

Main parameters of a lease are numnodes (number of virtual nodes required), amount of physical resource for each node (required CPU and require Memory), start time (exact time), duration and end time (deadline). This lease requires only CPU and memory resource as a resource. Additionally, a lease can have more resources like disk image etc. User can specify the disk image which should be loaded at the beginning of loading the virtual machine. On receiving a lease, Haizea stores lease's information into Lease object. Then it tries to schedule the lease. If it can accept the lease, it reserves the physical resources for this by storing this information in SlotTable. A Slottable is used to store the information about physical nodes, reservations done on these nodes and available resources. As an output, Haizea prepares a schedule which decides where and when the virtual machines should be placed. When a new lease is submitted to Haizea, it accepts this lease if can give guarantee for providing asked resources to this lease at the asked time otherwise the lease is rejected. Haizea supports four types of leases: Immediate, Best effort, Advance reservation and Deadline sensitive. One can categorize an arrived lease on the basis of parameters provided by the user. Following table explains the four types of leases supported by the Haizea.

Table 1. Types of leases based on parameters

Preemptible	Start time	End time	Lease Type
No	Current time	= start time + duration	Immediate
No	Any other than current time	= start time + duration	Advance reservation
Yes	Any	> start time + duration	Deadline sensitive
Yes	Any	None	Best effort

3 Proposed Algorithms

This paper proposes resource allocation algorithm based on Multi Slot Allocation for deadLine sensitive leases. The algorithm and different procedures are explained as:

3.1 Proposed Algorithm to Handle Deadline Sensitive Leases

Scheduler searches the single slot for incoming deadline sensitive lease, if single slot is not available then the scheduler tries to accept the newly arrived lease by multiple slots allocation policy where it finds multiple time slots which together can provide asked amount of resource for asked amount of time. If scheduler can find such multiple slots, it accepts the lease and reserves the resources in all those time slots for the newly arrived deadline sensitive lease. If still scheduler cannot accept this lease, it reschedules already accommodated deadline sensitive lease to make resources free for the newly arrived deadline sensitive lease by backfilling policy. If still it is unable to accept the newly arrived deadline sensitive lease, it reschedules the best effort leases in that region and tries to provide resources to the newly arrived deadline sensitive lease. If it is unable to do so, then it rejects the newly arrived deadline sensitive lease.

Multiple Slot Allocation Policy. ETs (Eligible Timeslots) include all the time slots between startTime and endTime of newly arrived lease which have asked resources free for some duration. ATs(Accepted Timeslots) are the time slots selected from ETs such that the minimum number of slots is used and wastageTime is minimum.

$$wastageTime = requirement - sum(duration\_of\_all\_slots\_of\_ATs) \quad (2)$$

Table 2. Terms used in the proposed algorithm

Input	Data type	Details
ETs	Array of object	Array of eligible time slots which can provide required resources (input to the algorithm)
ATs	Array of object	Array of accepted time slots out of ETs (output of the algorithm)
Requirement	Number	Time requirement (in minutes) requested in the lease
Min_Slots_ Required	Number	Minimum number of slots required for the lease to be accepted (calculated by the algorithm)
ELs	Array of object	Eligible leases for backfilling

Initially ETs are sorted in the descending order of their durations. Next, the minimum number of slots required is decided. If first x slots together are able to satisfy the time requirement, then x is the minSlotsRequired. For creating initial set of ATs, first x slots from ETs are put in ATs. Next, each slot of ATs with one slot at a time, is replaced from ETs untill wastageTime becomes zero or negative. When wastageTime becomes zero, the procedure stops with current selection of ATs. If wastageTime becomes negative, it stops with previous selection of ATs.

Algorithm 1. CHOOSE-SLOTS

DESCRIPTION: Given the ETs, ATs are selected based on:

1. All ATs together satisfies the requirement
2. minimum number of slots are used
3. wastageTime is minimum

INPUT: ETs, requirement OUTPUT: ATs

```
ETS-SORT-DESC(ETs)
FIND-MIN-SLOTS-REQUIRED(ETs, requirement)
ATs = INTIALIZE-ATS (ETs, minSlotsRequired)
OPTIMIZE-ATS (ATs,ETs,requirement)
return
```

Algorithm 2. ETS-SORT-DESC

DESCRIPTION: Sorts the given ETs in the descending order of their time duration. For reducing the total number of slots, time slots with larger duration are used first in algorithm4. INPUT: ETs (unsorted) OUTPUT: ETs (sorted in descending order)

```
for $i = 0$ to ET.size do
    for $j = i$ to ET.size do
        if ETs[i].duration < ETs[j].duration then
            temp = ETs[i]
            ETs[i] = ETs[j]
            ETs[j] = temp
        end if
    end for
end for
return
```

Algorithm 3. FIND-MIN-SLOTS-REQUIRED

DESCRIPTION: Using the sorted ETs, it fixes the minSlotsRequired. Aim of this algorithm is to find a minimum number of ATs which together can satisfy the requirement. INPUT: ETs, requirement OUTPUT: minSlotsRequired

```
minSlotsRequired = 0
requirementAchieved = 0
while requirementAchieved < requirement do
    requirementAchieved = requirementAchieved + ETs[minSlotsRequired]
    minSlotsRequired ++
end while
return minSlotsRequired
```

Algorithm 4. INTIALIZE-ATS

DESCRIPTION: Algorithm selects the initial set of accepted time slots. It uses the slots with maximum time duration. INPUT: ATs, ETs, minSlotsRequired OUTPUT: ATs

```
for $i = 0$ to minSlotsRequired do
    ATs.add(ETs[0])
    remove ETs[0]
end for
return
```

Algorithm 5. OPTIMIZE-ATS

DESCRIPTION: Algorithm optimizes the selection of ATs. Aim of this optimization is to minimize the unused time from ATs. INPUT: ATs, ETs, requirement OUTPUT: ATs

```
for $i = 0$ to ET.size do
    for $j = 0$ to AT.size do
        changeInDuration = ATs[j]-ETs[i]
        if ATs.totalduration-changeInDuration >= requirement then
            ATs[j] = ETs[i]
            remove ETs[i]
        end if
    end for
end for
return
```

3.2 Backfilling Policy

This considers all the leases to be rescheduled whose time duration intersects with the time duration of newly arrived lease. For this purpose, FIND-LEASES-TO-BE-RESCHEDULED function is used. A lease is considered for rescheduling if it satisfies any of the following conditions:
(1) Having start time less than the start time of newly arrived lease but end time greater than the start time of newly arrived lease.
(2) Having start time greater than the start time of newly arrived lease and less than the end time of newly arrived lease. For rescheduling already accommodated leases, scheduler finds all the leases whose requested resources are free. All these leases are put into an array named ELs (Eligible Leases). ALGO-BACKFILLING reschedules such leases. Each lease is selected at a time, scheduler applies multiple slot allocation policy on that lease to make resources free in that region. The same procedure is repeated unless the requirement is not achieved.

Algorithm 6. ALGO-BACKFILLING

```
DESCRIPTION: Algorithm backfills the leases coming between given startTime and given endTime.
INPUT: startTime, endTime, requirement
ELs = FIND-LEASES-TO-BE-RESCHEDULED
reqAchived = 0 3
ATs = null
for all el lease in ELs do
    find ETs for el
    remove current slot from ETs
    duration = el.duration
    CHOOSE-SLOTS(ETs, duration)
    if el gets rejected then
        reschedule el to new time slot
        add time slots of el to ATs
        reqAchived = reqAchived+duration
    end if
    if reqAchived >= requirement then
        schedule newly arrived lease in slots by ATs
        accept the new lease
        return
    end if
end for
Reject the newly arrived lease
return
```

4 Experimental Setup

OpenNebula is a cloud toolkit which can be used to setup a cloud on a local infrastructure[13]. Haizea is a resource lease manager which can be used as a virtual machine scheduler for OpenNebula[10]. OpenNebula with Haizea is used as a cloud toolkit to develop a cloud that can support various resource allocation policies. Haizea can be used as a stand alone lease manager or simulator. To verify the proposed algorithm, code has been incorporated in the Haizea‘s present codebase and simulated. Comparative analysis for different parameters proves the importance of the algorithm.

5 Results

In order to evaluate the performance of proposed algorithm to schedule deadline sensitive leases, it is compared with the existing algorithm of Haizea to schedule deadline sensitive leases. To evaluate both the algorithms, following metrics were considered.

1. Number of accepted leases
2. Resource Utilization-CPU
3. Resource Utilization-Memory
4. Lease acceptance ratio

Results for number of backfilled leases and number of slots are also included. For analysis, 500 lease files were generated. For generating each file, these parameters were generated randomly:

1. Number of leases(1 to 100)
2. Lease type of each lease
3. Number of nodes required for each lease
4. Amount of CPU required for each node
5. Amount of memory required for each node
6. Start time of each lease
7. Running time required (duration) for each lease
8. End time (deadline) of each lease

CPU utilization is defined as follows:

$$CPUutilization = \frac{Total\_CPU\_Used}{Total\_Available\_CPU} \quad (3)$$

5.1 Number of Accepted Leases

Table 3 shows the comparison between the number of accepted leases by the existing algorithm and the number of accepted leases by the implemented algorithm.

Table 3. Comparison of number of accepted leases

Total	Existing	Proposed Algorithm	Difference
9	5	7	2
27	8	17	8
52	11	29	18
73	12	43	31
91	12	55	43

As shown in figure 1, the number of accepted leases by the implemented algorithm is more than that of the existing algorithm.

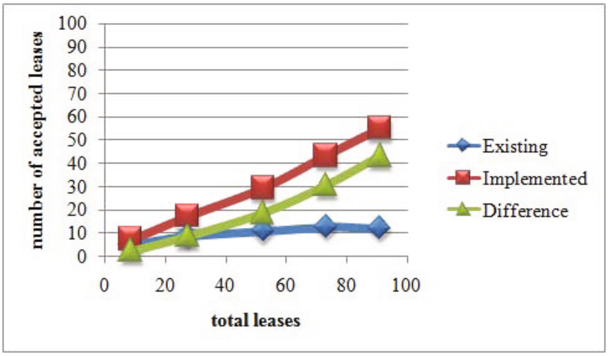


Fig. 1. Comparison of number of accepted leases

5.2 CPU Utilization (%)

Table 4 shows the comparison between the % CPU utilization by the existing algorithm and the % CPU utilization by the implemented algorithm.

Table 4. Comparison of % CPU utilization

Total	Existing	Proposed Algorithm	Difference
9	21.55	45.43	23.88
27	26.55	46.41	19.85
52	25.69	44.78	19.09
73	20.04	44.86	24.82
91	24.35	43.44	19.08

As shown in figure 2, the % CPU utilization by implemented algorithm is more than that of the existing algorithm.

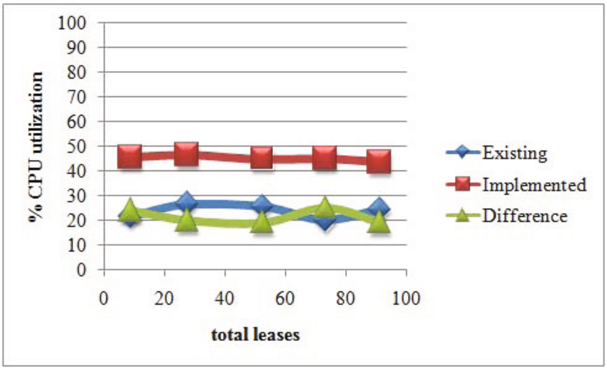


Fig. 2. Comparison of % CPU utilization

5.3 Memory Utilization (%)

Table 5 shows the comparison between the % Memory utilization by the existing algorithm and the % Memory utilization by the implemented algorithm.

Table 5. Comparison of % memory utilization

Total	Existing	Proposed Algorithm	Difference
9	18.11	46.52	28.41
27	21.23	48.85	27.61
52	23.08	47.55	24.46
73	24.52	48.39	23.87
91	24.07	46.29	22.23

As shown in table 5, the % Memory utilization by implemented algorithm is around 50% more than that of the existing algorithm.

5.4 Lease Acceptance Ratio

Table 6 shows the comparison between the lease acceptance ratio by the existing algorithm and the lease acceptance ratio by the implemented algorithm.

Table 6. Comparison of lease acceptance ratio

Total	Existing	Proposed Algorithm	Difference
9	60.30	82.22	21.92
27	32.77	63.20	30.42
52	20.43	55.25	34.82
73	17.04	58.58	41.54
91	13.08	60.22	47.14

6 Conclusions

The proposed algorithm applies two concepts into the existing algorithm: multiple slot allocation and backfilling. When the scheduler is unable to accept newly arrived deadline sensitive lease by single slot allocation policy, it tries to divide this lease in multiple slots and tries to accept. If it is still unable to accept the newly arrived deadline sensitive lease, then it tries to reschedule already accommodated deadline sensitive lease(s). The same multiple slots allocation policy is used for rescheduling already accommodated deadline sensitive lease(s) and hence creating space for the newly arrived lease. By applying the multiple slots allocation and back filling policy, it is shown that the number of accepted leases increase and hence the acceptance ratio of leases is improved.

References

1. Amazon elastic compute cloud (amazon ec2) (May 05, 2011), http://aws.amazon.com/ec2/
2. Nimbus (May 05, 2011), http://www.nimbusproject.org/
3. Akhani, J., Chuadhary, S., Somani, G.: Negotiation for resource allocation in iaas cloud. In: Proceedings of the Fourth Annual ACM Bangalore Conference, COMPUTE 2011, pp.15:1–15:7. ACM (2011)
4. Chang, F., Ren, J., Viswanathan, R.: Optimal resource allocation in clouds. In: Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010, pp. 418–425. IEEE Computer Society, Washington, DC, USA (2010), http://dx.doi.org/10.1109/CLOUD.2010.38
5. Nathani, A., Chaudhary, S., Somani, G.: Policy based resource allocation in iaas cloud. Future Generation Computer Systems. Journal of Special Issue (2011)
6. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The eucalyptus open-source cloud-computing system. In: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID 2009, pp. 124–131. IEEE Computer Society, Washington, DC, USA (2009), http://dx.doi.org/10.1109/CCGRID.2009.93
7. Sotomayor, B., Foster, I., Keahey, K.: Overhead matters: A model for virtual resource management. In: Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing, IEEE Computer Society, Washington, DC, USA (2006)
8. Sotomayor, B., Keahey, K.: Haizea (May 05, 2011), http://haizea.cs.uchicago.edu/
9. Sotomayor, B., Keahey, K., Foster, I.: Combining batch execution and leasing using virtual machines. In: Proceedings of the 17th International Symposium on High Performance Distributed Computing, HPDC 2008, pp. 87–96. ACM (2008)
10. Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: Resource leasing and the art of suspending virtual machines. In: 11th IEEE International Conference on High Performance Computing and Communications, pp. 59–68. IEEE (2009)
11. Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: An open source solution for virtual infrastructure management in private and hybrid clouds. IEEE Internet Computing, 5–8 (2009)
12. Sotomayor, B., Montero, R.S., Foster, I., Llorente, I.M.: Virtual infrastructure management in private and hybrid clouds. IEEE Internet Computing 13, 14–22 (2009)
13. Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: Capacity leasing in cloud systems using the opennebula engine. In: Cloud Computing and Applications, CCA 2008 (2009)
14. Stankovic, J.A., Ramamritham, K., Spuri, M.: Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms. Kluwer Academic Publishers, Norwell (1998)

Grids Security without Public Key Settings

Manik Lal Das

Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar - 382007, India
maniklal_das@daiict.ac.in

Abstract. Grid system [1] involves the collaborative use of computers, networks, devices, software, databases and interfaces maintained by multiple organizations. In recent years, the development of Grid system [2], [3] has gained increasing interests from researchers. In this paper, a security solution is proposed for Grid system without public key settings.

Keywords: Grid system, Grid security, Authentication, Key establishment.

1 Summary of the Proposed Protocol

The communicating entities of a Grid system are user, authentication server, resource server, process, user proxy and resource proxy. Based on applications' requirement, user can create user proxy and resource server can create resource proxy. Security among several entities in Grids is an important concern and numerous security protocols using public key settings have been proposed in literature.

The proposed protocol uses a cryptographically secure keyed hash function for communicating entities authentication and session key establishment between them. The protocol has four phases as follows.

Authentication Server Setup. The Authentication Server (AS) is a trusted entity, who selects a master secret key k for regulating security in Grids.

User Registration. A new user requires to register with the Grid system. User submits her identity, uid, to AS. Then AS selects a random salt s_{uid}, computes user's secret key $K_u = \text{MAC}(k; < uid \| s_{uid} >)$ and sends K_u to the user in a secure manner. Here, MAC() is a keyed hash function.

Resource Server Registration. A new resource server(RS) requires to register with the Grid system. RS submits its identity, rid, to AS. Then AS selects a random salt s_{rid}, computes RS's secret key $K_r = \text{MAC}(k; < rid \| s_{rid} >)$ and sends K_r to RS in a secure manner. AS keeps the records of all registered users and RSs in its database.

Authentication and Session Key Establishment. This phase provides mutual authentication of user and RS (and between their delegated agents) followed by a session key establishment.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 253–254, 2012.

– User chooses a nonce r_u, computes $c_u = \mathrm{MAC}(K_u; < uid >) \oplus r_u$, $h_u = \mathrm{MAC}(K_u; < uid \| r_u >)$, and then sends $< uid, rid, c_u, h_u >$ to AS. AS first validates uid and if uid is valid then AS computes $K'_u = \mathrm{MAC}(k; < uid \| s_{uid} >)$ and obtains $r'_u = \mathrm{MAC}(K'_u; < uid >) \oplus c_u$. Then AS computes $h'_u = \mathrm{MAC}(K'_u; < uid \| r'_u >)$ and checks whether $h'_u = h_u$. If it holds, user is authenticated; otherwise, AS terminates the communication.

– AS chooses a nonce r_a, computes $K'_r = \mathrm{MAC}(k; < rid \| s_{rid} >)$, $c_a = \mathrm{MAC}(K'_r; < rid >) \oplus r_a$, $X = r'_u \oplus r_a$ and $h_a = \mathrm{MAC}(K'_r; < rid \| uid \| r'_u \| r_a \| X >)$. Then, AS sends $< rid, c_a, X, h_a >$ to RS. RS obtains $r_a = \mathrm{MAC}(K_r; < rid >) \oplus c_a$ and gets $r'_u = X \oplus r_a$. RS computes $h'_a = \mathrm{MAC}(K_r; < rid \| uid \| r'_u \| r_a \| X >)$. AS is authenticated if $h'_a = h_a$; otherwise, RS terminates the communication.

– RS chooses a nonce r_s, computes $SK_{RU} = \mathrm{MAC}((r'_u \| r_s); < uid \| rid >)$, $R = r'_u \oplus r_s$ and $h_r = \mathrm{MAC}(SK_{RU}; < rid \| uid >)$. Then, RS sends $< rid, uid, R, h_r >$ to the user. The user obtains $r_s = R \oplus r_u$, computes $SK_{UR} = \mathrm{MAC}((r_u \| r_s); < uid \| rid >)$ and $h'_r = \mathrm{MAC}(SK_{UR}; < rid \| uid >)$. RS is authenticated if $h'_r = h_r$; otherwise, user terminates the communication.

If all the above steps occur successfully, SK $(= SK_{RU} = SK_{UR})$ acts as the session key between user and RS.

– User computes $h_c = \mathrm{MAC}(SK_{UR}; < uid \| rid \| \text{“}DONE\text{”} >)$ and sends $< uid, rid, h_c,$ *request for resource*$>$ to RS. RS checks whether $h_c = \mathrm{MAC}(SK_{RU}; < uid \| rid \| \text{“}DONE\text{”} >)$. If it holds, they start transmitting data encrypted under the key SK; otherwise, terminate the communication.

Conclusion. In this paper, we provided a security solution for Grids without public key settings. The proposed key establishment protocol could be extended to other peers such as proxy user and proxy server. The protocol is efficient in comparisons to other Grid security protocols [4], [5], which do not require public key on server.

Acknowledgement. This work is supported in part by Department of Science and Technology, Ministry of Science & Technology, Government of India through DST/INT/SPAIN/P-6/2009 Indo-Spanish Joint Programme of Cooperation in Science and Technology.

References

1. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A security architecture for computational grids. In: Proc. of the ACM Conference on Computer and Communications Security, pp. 83–92 (1998)
2. The Globus Alliance. The Globus Project, http://www.globus.org/
3. Grid Software security. Components for Grid security, http://www.globus.org/grid_software/security/
4. Chang, Y., Chang, C., Liu, Y.: Password authentication without the server public key. IEICE Transactions on Communications E87-B(10), 3088–3091 (2004)
5. Yoon, E., Yoo, K.: An Efficient Password Authentication Scheme Without Using the Server Public Key for Grid Computing. In: Zhuge, H., Fox, G.C. (eds.) GCC 2005. LNCS, vol. 3795, pp. 149–154. Springer, Heidelberg (2005)

Concurrent HCM for Authorizing Grid Resources

Mustafa Kaiiali, Chillarige Raghavendra Rao, Rajeev Wankar, and Arun Agarwal

Department of Computer and Information Sciences
University of Hyderabad, Hyderabad, India
mustafa_kaiiali@ieee.org
{crrcs,wankarcs,aruncs}@uohyd.ernet.in

Grid computing is concerned with a shared and coordinated use of heterogeneous resources, belongs to distributed virtual organizations to deliver nontrivial quality of services. In grids, security has a major concern. The heterogeneity, massiveness and dynamism of grid environments complicate and delay the authorization process. This brings out the need for a fast and scalable fine-grained access control (FGAC) mechanism to cater well to grid requirements.

Every resource in a grid has its own security policy, which may be identical or quite similar to other security policies of some other resources. This fact motivates the idea of the ability to cluster the resources which have similar security policies in a hierarchical manner based on their shared security rules. This idea was introduced in [1], and it was called the Hierarchical Clustering Mechanism (HCM).

Considering the huge number of users and resources which may exist in the grid, then using HCM may cause a bottleneck to the authorization system as a centralized agent targeted by all authorization requests. The direct solution to this problem could be by replicating the *decision tree* into several authorization servers to reduce the load on the centralized agent. However, one can think of enhancing the HCM itself in order to scale up to numerous authorization requests taking place at a time. If this was not sufficient to avoid the bottleneck, then one can think of replicating HCM.

This paper explains how the HCM *decision tree* can be processed in parallel to serve multiple authorization requests at a time. The authorization server boots with a head process named “Welcoming Process” (See Fig. 1.). This process is responsible for capturing the authorization request and forking a new authorization process to serve the incoming request concurrently. Fig. 2 represents an outline pseudo code of typical authorization server that uses concurrent HCM. When a new request comes, “accept” method returns. Then the server calls “fork” to create a child process named “Authorization Process”. The Authorization Process serves the incoming request while the parent process waits for another request to come.

Creating a new process for every authorization request might be expensive in terms of memory consumption as every process has its own separated segments in memory. Thus the use of “clone” rather than “fork” is strongly recommended as the first one creates a “light weight” process which shares parts of its execution context with the parent process, such as the memory space.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 255–256, 2012.

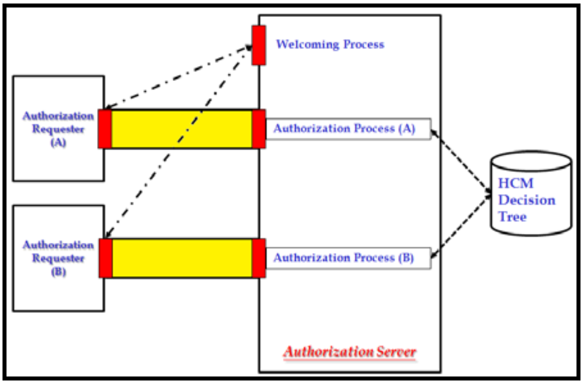

Welcoming Process
Authorization Requester (A)
Authorization Process (A)
HCM Decision Tree
Authorization Requester (B)
Authorization Process (B)
Authorization Server

Fig. 1. Processing Multiple Authorization Requests Simultaneously

```
pid_t pid;
int   welProfd, authProfd;

/* Fill with the Welcoming Process well-known port */
welProfd = socket( ... );
bind(welProfd, ... );
listen(welProfd, 1);

while(true) {         /* Infinite loop */
  authProfd = accept (welProfd, ... );

  if( (pid = fork()) == 0) {
    /* The child process closes the welcoming process' socket */
    close(welProfd);
    /* Process the request & return the user's authorized resource group (UARG) */
    UARG rg = auhtorize(authProfd);
    close(authProfd); /* Done with this authorization request */
    exit(0);          /* Child terminates */
  }
   }
```

Fig. 2. Outline pseudo code for typical HCM authorization server

The "Authorization Process" itself can also be parallelized for maximum utilization of the authorization server' CPUs. The main thread of the process starts at the root node of the *decision tree*. Then for each branch, a new thread is created to parse that particular branch. When a thread encounters an unfulfilled security rule, it gets terminated. If the *decision tree* is huge, it won't be practical to create new thread for each branch. That can cause too much context-switching overhead. Thus, a "MAXNTHREADS" parameter has to be used to control the number of threads. Like that, only the high level nodes will be associated with new threads. One more advantage of parallelizing the authorization process is that it allows what we can call "**Authorization on the fly**" where a thread makes its associated resources available to the scheduler without waiting to complete parsing the whole tree.

Reference

1. Kaiiali, M., Wankar, R., Rao, C.R., Agarwal, A.: Design of a Structured Fine-Grained Access Control Mechanism for Authorizing Grid Resources. In: Paulo, S. (ed.) IEEE 11th International Conference on Computational Science and Engineering, Brazil, July 16-18, pp. 399–404 (2008)

Seamless Provision of Cloud Services Using Peer-to-Peer (P2P) Architecture

Snehal Masne, Rajeev Wankar, C. Raghvendra Rao, and Arun Agarwal

Department of Computer and Information Sciences
University of Hyderabad, Hyderabad, India
snehalmasne@gmail.com, {wankarcs,crrsm,aruncs}@uohyd.ernet.in

Abstract. Cloud computing involves highly variable resource requirement that demands high availability, scalability and performance. At times, single cloud service provider would be saturated or running out of resources and may be unable to provide the services to its client, resulting in poor scalability and reliability. It may tarnish the trust parameter of customer. In addition, there is huge investment in setting up a single scalable cloud, which in turn has many environmental impacts. In this work, we propose an architecture to inter-connect different clouds in P2P fashion to address the problems like efficiency bottleneck and single point of failure that are predominantly associated with traditional approaches. This idea gives access to much larger pools of resources/services. Each provider can maximize their profit by creating new collaborative services. These capabilities can be available and tradable through a service catalogue to support innovations and applications.

Keywords: Cloud scalability, Interoperability, Cloud (P2P) architecture.

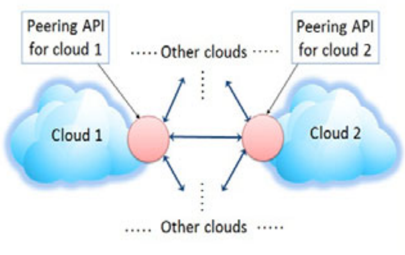
Issues with Traditional Clouds: Usually clouds are owned and operated by individual companies. Each of them has created its own closed network, which is expensive to setup and maintain. In addition, consumers are restricted to offerings from a single provider at a time and thus cannot use the resources (services) of multiple Cloud providers at the same time. Having enormous pool of resource infrastructure also does not seem a good idea, as it involves substantial investment and chances are more that it is going to have adverse effects on the environment as well.

This paper leverages Peer-to-Peer (P2P) cloud networks to speed up the cost-effective provisioning of cloud infrastructure. This proposed theme should not be confused with grid computing. Our P2P concept, as explained, is akin to outsourcing whereas grid computing is more geared toward increasing computing efficiency.

Related Work: Some ideas have already been proposed pertaining to the interoperability. Unified Cloud Interface (UCI) and Cloud Orchestration Platform are the two approaches [1]. The literature suggests having a bigger cloud composed of many other clouds with single point of interaction for all users. These are centralized approaches, as specific cloud provider and customer relation is not visible here.

Proposed Architecture: Figure 1.A shows the Peering API (indicated with circles) acts as the point of interaction among different clouds in P2P. Each of these Peering APIs will contact other clouds and get the work done (request or provision).

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 257–258, 2012.



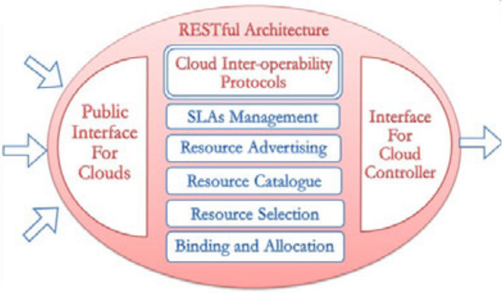


Fig. 1. A. Proposed Peering-API **Fig. 1. B.** Components of our suggested Peering API

Components of Peering API : Refer to Figure 1.B; the dedicated components are :

- Public interface for clouds: This enables a cloud to communicate with other peer clouds. It is reserved only for clouds and not for clients.
- Interface for CC: It serves point of contact to get the details of availability and it performs the resource allocation through this module.
- Cloud Inter-operability Protocols: As various cloud providers have different architectures; this module tries to bridge the gap by having interoperability foundations[2][3] with: Cloud Identity Management, VM Mobility and Security[4].
- SLA management: It talks about agreements on provision of services.
- Resource Advertising and Catalogue: Advertisements for extra resources are sent in the form of simple XML file, at regular intervals, and other clouds keep that in a directory called as Catalogue.
- Resource Selection, Binding and Allocation: Resource selection is based on many parameters like SLAs, trust, pricing, vicinity, past records, etc. The auction model can be used to select cost effective provider(s). This is followed by the binding and allocation through secure channel.

Thus, our proposal tries to exploit the underutilized computing resources in cloud world and achieves the seamless provision of services even in the event of large fluctuations in computing load that cannot be handled by a single cloud system. In fact, this may prove to be the basic design for future of cloud computing.

References

1. Parameswaran, A.V., Chaddha, A.: Cloud Interoperability and Standardization. In: SETLabs Briefings, vol. 7 (2009)
2. Assunção, M., Buyya, R., Venugopal, S.: InterGrid: A Case for Internetworking Islands of Grids. In: Concurrency and Computation: Practice and Experience (CCPE), vol. 20(8), pp. 997–1024. Wiley Press, New York (2008)
3. Bernstein, D.: Keynote 2: The InterCloud: Cloud Interoperability at Internet Scale. In: Sixth IFIP International Conference on Network and Parallel Computing (2009)
4. Vouk, M.A.: Cloud computing-Issues, research and implementations. In: 30th International Conference on Information Technology Interfaces, ITI (2008)

Distributed Fault Tolerant Estimation in Wireless Sensor Network Using Robust Diffusion Adaptation

Meenakshi Panda and Pabitra M. Khilar

Department of CSE, National Institute of Technology, Rourkela, India-769008
meenakshi.nitrkl@gmail.com, pmkhilar@nitrkl.ac.in

Abstract. The problem of robust distributed estimation in wireless sensor network (WSN) when few sensor nodes are faulty is addressed here. In WSN, each sensor node collects scalar measurements of some unknown parameters and then estimates the parameter of interest from the data collected across the network. An iterative distributed linear parameter estimated algorithm is proposed here by using diffusion co-operation. Each node updates its information by using the data collected by it and the information received from the neighbours. The mean square error (MSE) of distributed estimation schemes increases whenever any faulty sensor node in the network fails to transmit correct information, which leads to inaccurate estimation. Hence a robust diffusion linear estimation algorithm using Hubber's cost function is proposed here in order to improve the accuracy of the estimation.

Keywords: Wireless sensor network, robust estimation, faulty node.

1 Introduction

Recent development in low power VLSI technology and efficient wireless communication enhances the ability of WSN to monitor and interact with physical environment [1]. It is a common practice that the sensor node becomes faulty due to various reasons like natural disaster, environmental noise, energy depletion *etc.*. Sometimes few nodes give erroneous data which are called soft faulty sensor node [2]. In this analysis the data received from a faulty node is treated as the data corrupted by impulsive noise or mixed with outliers. In such a scenario, the least mean square based distributed estimation algorithm's performance degrades, so that the fault tolerance is a key challenge in keeping the network sustainable under any unavoidable circumstances. Fault detection and recovery techniques have been reported in literature [3].

A WSN is called to be fault tolerant if it is robust in presence of node and link failure. In both the cases, faulty node's data are recovered from some other nodes which keeps back up of that node's data. For this purpose multi path routing technique [4] is used. But this technique requires more energy, bandwidth, memory, and recovery time consumption and generates traffic over the

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 259–260, 2012.

network. Another approach is each active sensor node identifies its fault status by using either self or co-operative learning techniques and distributes its status over the network so that every node has an idea about the faulty nodes [5]. During event detection each sensor node considers the fault free nodes data by excluding the faulty node status. This technique also consumes more energy and puts overhead on the network during fault status exchange. Active and passive replication techniques are used for fault recovery which requires more memory for storing the data at some intermediate nodes [6]. From the above discussion, we conclude that, each node using its computational power tries to estimate the parameter of interest or keeps record of some sensor nodes data. For this it requires some extra memory and puts overhead on the underline network. This degrades the network service quality. Therefore, there is a great deal of effort to devise an algorithms that is able to improve the service quality of the network with information exchange among nodes.

In the proposed distributed linear parameter estimation techniques, every node in the network communicates with a subset of the nodes, and processing task is distributed among all the nodes in the network [7]. It is assumed that the faulty node in the network is able to communicate with neigbours, but the data is corrupted by impulsive nose. Each node uses distributed diffusion based linear estimation technique to compute the parameters of interest. For this, each sensor node needs fixed and small storage memory. Along with this each node exchanges small amount of information with its neighbors in a very simple manner that means routing algorithms are not required for data communication. With the help of simulation, we can show that the proposed algorithm provides better performance for faulty wireless sensor network.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communication Magazine 40, 102–114 (2002)
2. Guo, S., Zhou, Z.: FIND: Faulty Node Detection for Wireless Sensor networks. In: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys 2009), pp. 253–266 (2009)
3. Asim, M., Mokhtar, H. Merabti, M.: A cellular approach to fault detection and recovery in wireless sensor networks. In: Third International Conference on Sensor Technologies and Applications (SENSORCOMM 2009), pp. 352–357 (2006)
4. Ganesan, D., Govindan, R., Shenker, S., Estrin, D.: Highly Resilient, Energy-Efficient Multi path Routing in Wireless Sensor Networks. Mobile Computing and Communications Review 1(2), 1–13 (1997)
5. Gupta, G., Younis, M.: Fault-Tolerant Clustering of Wireless Sensor Networks. Wireless Communications and Networking 3, 1579–1584 (2003)
6. Staddon, J., Balfanz, D., Durfee, G.: Efficient Tracing of Failed nodes in Sensor Networks. In: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, pp. 122–130 (2002)
7. Cattivelli, F.S., Sayed, A.H.: Diffusion LMS Strategies for Distributed Estimation. IEEE Transactions on Signal Processing 58(3), 1035–1048 (2010)

Robust Distributed Block LMS over WSN in Impulsive Noise

Trilochan Panigrahi[1], Ganapati Panda[2], and B. Mulgrew[3]

[1] Department of ECE, National Institute of Technology Rourkela, India-769008
panigrahit@nitrkl.ac.in
[2] School of Electrical Science, Indian Institute of Technology Bhubaneswar, India
[3] IDCOM, The University of Edinburgh, UK

Abstract. In wireless sensor network each sensor node collects data related to some unknown parameters, corrupted by independent Gaussian noise. Then the objective is to estimate the parameter from the data collected across the network in distributed manner. The distributed estimation algorithm should be energy efficient, provides high estimation accuracy, and is fast in convergence. But the conventional distributed algorithm involves significant communication overhead and is also not robust to the impulsive noise which is common in wireless sensor network environment. Consequently these algorithms defeat the basic purpose of wireless sensor network. This paper studies the problem of robust adaptive estimation in impulsive noise environment using robust cost function like Wilcoxon norm and Huber cost function. Further in order to reduce the amount of communication overhead, block distributed LMS is incorporated.

Keywords: Wireless sensor network, contaminated Gaussian noise, distributed signal processing, incremental LMS, Wilcoxon norm.

1 Introduction

In wireless sensor networks(WSN) the tiny sensor nodes are employed to collect data over a geographical area for the applications like precision agriculture, disaster relief management, and military applications. In these applications, each node with its computational power is able to send data to a subset of the network nodes, and tries to estimate the parameter of interest [1, 2]. Therefore, there is a great deal of effort in devising algorithms that are able to improve the estimate of the parameters of interest in every node with information exchange between nodes [3]. More precisely, in mathematical terms, each node optimizes a cost function that depends on all information in the network. The main challenges in optimizing such functions are that no node has direct access to all information, and the network topology can change over time (due to link failures, position changes, and/or reachability problems). The presence of impulsive noise or outliers *i.e.* when data is contaminated with non-Gaussian noise degrades the performance of the network. The conventional estimation algorithms,

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 261–262, 2012.

which is based on least mean squared error as the cost function, is not robust to impulsive noise. Thus there is a need to develop robust estimation algorithm in a distributed scenario to alleviate the effect of outliers.

Recently the concept of distributed adaptive incremental algorithms has been developed in the literature [3–5] to increase the energy efficiency of sensor network. One of such schemes is incremental cooperative technique which provides a truly global solution in estimating unknown parameters in WSN. But it is a fact that the gradient based incremental algorithm is not robust to impulsive type of noise. To make the algorithm robust for impulsive noise, here a new class of distributed algorithm based on Wilcoxon norm and Hubber's function is introduced.

This paper presents the robust distributed block incremental LMS algorithms in presence of the contaminated Gaussian impulsive noise. With the help of simulation we can show that the robustness of proposed algorithm over the conventional incremental LMS. The proposed algorithm needs same computation and communication resources as required in case of incremental LMS. The remarkable achievement of the proposed algorithm is that a node performs L(block size) times lesser communications compared to conventional sequential distributed LMS algorithms.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communication Magazine 40, 102–114 (2002)
2. Estin, D., Govindan, R., Heidemann, J., Kumar, S.: Next century chalanges:Scalable coordination in sensor network. In: Proc. ACM/IEEE MobiComm 1999, pp. 263–270 (August 1999)
3. Cattivelli, F.S., Sayed, A.H.: Analysis of Spatial and Incremental LMS Processing for Distributed Estimation. IEEE Transactions on Signal Processing 59(4), 1465–1480 (2011)
4. Nedic, A., Bertsekas, D.P.: Incremental Subgradient Methods for Nondifferentiable Optimization. SIAM J. on Optimization 12(1), 1052–1062 (2001)
5. Rabbat, M.G., Nowak, R.D.: Decentralized source localization and tracking [wireless sensor networks. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004), vol.3, pp. iii-921–iii-924 (2004)

Resource Allocation Techniques Based on Availability and Movement Reliability for Mobile Cloud Computing*

JiSu Park[1], HeonChang Yu[1], and EunYoung Lee[2,**]

[1] Dept. of Computer Science Education, Korea University
{bluejisu,yuhc}@korea.ac.kr
[2] Dept. of Computer Science, Dongduk Women's University
elee@dongduk.ac.kr

Abstract. The researches on utilizing mobile devices as resources in mobile cloud environments have been gained attentions recently because of the enhanced computer power of mobile devices with the advent of dual cores chips. In this paper, propose a resource allocation technique which offers reliable resource allocation considering the availability of mobile resources and movement reliability of mobile resources. We also demonstrate the performance of our technique through the experiments.

Keywords: Mobile Cloud, Resource Allocation, Availability, Movement Reliability, Fault Tolerance.

1 Introduction

Recent researches showed the attempts in which mobile resources are used for distributed processing of compute-intensive applications in mobile cloud computing environments [1][2]. However, because the mobile devices move freely, the network connection can be changed or disconnected. The load of the system and the complexity of job management increase while allocating and/or reallocating the necessary resources dynamically depending on the need of requested services. Therefore, the failure to providing requested services will degrade the overall system performance. Therefore, a resource scheduler should assign the task on reliable resources to perform a stable operation.

2 Technique for Reliable Resource Allocation

The availability is classified into three different states. Job Queue State (S_Q) stands for a state, in which a job will start when a resource available message is received, but other tasks except for the designated task cannot be executed. In Job Available State (S_A), a job can be completed or start as soon as resource available message is received. It is the state in which other jobs can be assigned any time. Job Failure State (S_F) is the state in which no job can be processed because of fault or failure. The availability state of a mobile device is determined using the availability information. The formulas are as follows:

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(20110005630).

** Corresponding author.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 263–264, 2012.

S_A: $High\_Availibility\_Theshold \le$ Availibility
S_Q: $Low\_Availibility\_Theshold \le$ Availibility $< High\_Availibility\_Theshold$
S_F: Availibility $< Low\_Availibility\_Theshold$

High_Availability_Theshold and *Low_Availability_Theshold* are the range values which can be configured by users according to the characteristics of mobile environments and applications. To predict the availability state, we use the change of mobile resources over time and Markov chain model. In our prediction model, the state transition probability is calculated by the previous state and the currents state of the mobile resources.

Members of a university campus can be categorized into staff, faculty, undergraduate students, graduate students, and guests. Mobility patterns of these campus members are divided into three classifications: Low Mobility Users (LMU), Middle Mobility Users (MMU), and High Mobility Users (HMU). Mobility probability of a mobile device is defined as follows.

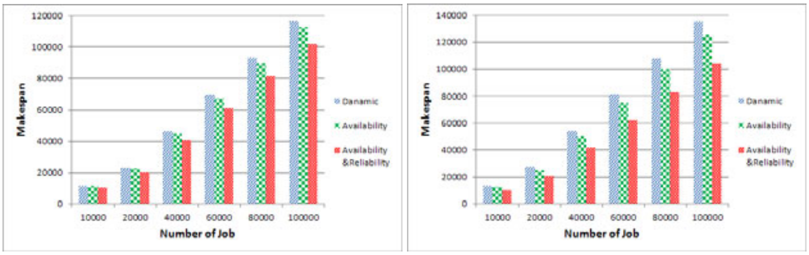
$$Loc(m,t) = (P_m(R_1), \quad P_m(R_2), \dots, P_m(R_k))$$
$$P_m = \sum_{j=1}^{k} P_m(R_{m,j})$$

If a mobile resource locates in one of the regions of *L*, the reliability of the mobile resource increases. In other words, if P_m is high, it means that there is a high probability that the mobile device *m* locates within the area *L*. Therefore, P_m can be used to measure the movement reliability. The movement patterns using the calculated movement reliability can be classified as follows.

LMU: $High\_Reliability\_Theshold \le$ Reliability
MMU: $Low\_Reliability\_Theshold \le$ Reliability $< High\_Reliability\_Theshold$
HMU: Reliability $< Low\_Reliability\_Theshold$

3 Performance Evaluation

Makespan is measured by considering the failure rate based on the performance and the failure rate based on the reliability. Figure shows another performance evaluation. In this experiment, job time was set to 1 minute, and the availability and reliability fault rate were set at 5%, 10% and 10%, 20%, respectively. The greater the amount of work and the higher fault rates, the better performance the proposed technique shows



(a) fault rate 5% and 10%

(b) fault rate 10% and 20%

4 Conclusion

In the proposed technique, the mobility of mobile devices is considered first, and then the performance of mobile devices is considered for reliable resource allocation; so, the resources with high movement reliability will be allocated first, and then the resources with high availability will be allocated.

Transparency Computation for Work Groups

A.B. Sagar

Department of Computers & Information Sciences
Hyderabad Central University, Hyderabad, India
bablusagar@gmail.com
www.uohyd.ernet.in

Transparency is being considered an indispensable ingredient in social accountability and necessary for preserving and guaranteeing ethical and fair processes. Transparency is related to visibility of information, and without it, work groups and stakeholders will be left in blind states. The growing importance to the requirement of transparency in business domains is the motivation to the present work. This paper presents a framework for transparency in the work groups and also specifies outlines for its implementation. This model can be used in organizations whose structure resembles the work group structure defined in this paper. This model does not incur much overhead as it involves only message passing for evaluating transparency of the work groups over tasks. Present day technologies are at such advanced levels that message passing even from very remote areas is not considered too difficult. According to this model, transparency of a work group can be evaluated during and after completion of task execution. During execution, if the task executing work group has reported sufficient number of transparency messages as defined by the task initiator till that time, then it is adjudged to be transparent; and the same applies even after completion of the task. The model tracks the levels of transparency and the shift between the levels/degrees (opaque, translucent, transparent), by requiring the task initiator to define the levels of transparency and the associated number and types of transparency messages, and the transparency of the task executor is evaluated by comparing the reported transparency messages with the required transparency messages. This model is most relevant to the Self Help Groups which are widely prevalent in India (about 3.37 million as of April 2011). However, it is applicable to any work group which resembles a SHG. A work group implies two or more individuals who routinely function like a team, and interdependent in achievement of a common goal, and may or may not work next to one another or in the same department. This kind of work groups are ever present in business domains. They represent a part of a business or the business itself. Transparency implies visibility of information related to financial and non-financial matters of the work group and its stakeholders. In the context of a business where several levels of administration are present, directions of transparency may take four directions: upwards, downwards, inwards, and outwards. *Upwards transparency* is meant to describe a hierarchical principal actor situation where the subordinate actor's actions (or transactions) can be observed by the principal; *Downwards transparency* is the opposite of upwards, ie. when the principal can be observed by subordinate actors; *Inwards transparency* is the transparency to all the insiders of the work group and *Outwards transparency* is the transparency to all the outsiders of the work group. Financial reporting is tracking of monetary data and non-financial reporting is task's execution-status data. We can classify transparency

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 265–266, 2012.

into three degrees : *opaqueness, translucency and clarity*. Opaqueness is when a work group does not disclose any information to its stakeholders and hence a opaque work group is not a transparent work group. Translucency is when a work group discloses its information partially. Hence, a translucent work group still cannot be called a transparent work group. Clarity is when a work group discloses all of its information. Only the work group having clarity degree of transparency is the transparent work group. Each task (w) will have three phases viz. pre-activity, per-activity and post-activity and transparency is defined on these three phases. Interestingly, the degree of transparency is directly proportional to the phases of activity. Prior to the pre-activity phase, the degree of transparency is opaque (or *null*). If the transparency conditions are met in the pre-activity phase, the degree becomes *translucent*. If the transparency conditions are met at per-activity phase and post-activity phases, then the degree of transparency becomes *clear*. As it is observed that after each activity phase, the degree of transparency is increasing, it is of interest to us to define the primary constituents of each activity phase which affect the degree of transparency. At each activity phase we define the transparency dimensions that are required. Though transparency dimensions are task- and situation-dependent, the following dimensions are mandatory. *Recognition of responsibilities and interdependencies* are primarily of concern before the realization of a given activity (pre-activity). *Recognition of status and problems* are primarily related to transparency into an ongoing activity (per-activity). Similarly, *understanding of performance* and *feedback* are related to post-activity transparency. There are three main perspectives in Transparency: *static , dynamic and radical*. In Static Transparency, the flow of information is mainly unidirectional i.e. from the work group to stakeholders. In *Dynamic Transparency* the work gorup and its stakeholders can exchange, share and compare information and adapt its online behavior and electronic requests and queries to the answers and reactions of respective counterparts. *Radical Transparency* refers to the capability of a firm's top management to employ internet-based technologies, such as rss, blogs and collaborative websites, in order to create a direct and continuous dialogue with customers and other stakeholders. Implementation of transparency in such a work group can be made using Member Behavior Model [MBM] and Task Execution Cycle [TEC]. MBM outlines the behavior of every member in a work group and also of the work group. MBM provides an imprint of generic behavior of each member (or work group) and TEC gives the different states of an executing task. Relation between MBM and TEC is that TEC is an integral part of ExTsk of the MBM. Transparency metric quantifies the levels of transparency of the work group or a member based on the transparency messages. Each transparency message can also be provided with a value corresponding to its priority. Thus, depending on the number of messages and the values associated with the messages, transparency can be computed. Since each activity has three phases i.e. pre, per and post , the transparency is a sum of transparencies of these three phases. Thus the present paper successfully *defines* and outlines *implementation* of Transparency.

Keywords: transparency, social values, ethics.

Acknowledgments. I am greatly thankful to Prof. Hrushikesha Mohanty for all his guidance and showing me the research directions.

A New Hierarchical Structure of Star Graphs and Applications

Wei Shi, Feng Luo, and Pradip Srimani

School of Computing, Clemson University, Clemson, SC 29634–0974

Abstract. A star graph S_n [1], of order n, is defined to be a symmetric graph $G = (V, E)$ where V is the set of $n!$ vertices, each representing a distinct permutation of n elements and E is the set of symmetric edges such that two permutations (nodes) are connected by an edge iff one can be reached from the other by interchanging its first symbol with any other symbol. The star graph S_n is a $(n - 1)$-regular graph with $n!$ nodes and $n!(n - 1)/2$ edges. Recursive hierarchical structure is one of the most attractive and well known properties of star graphs. A dimension n star graph can be divided into n substars of dimension $n-1$ by grouping the nodes with the same symbol at the ith position together, $2 \leq i \leq n$ [see Figure 1].

In this paper, we propose a new recursive hierarchical structure of star graphs. The objective is to redesign shortest routing in star graphs in the light of this new structure and design new efficient algorithms for shortest path multicast algorithms [2] adaptibe to bandwidth and latency requirements.

New Hierarchical Structure

In the new hierarchical structure, the $n!$ n des in a dimension n star graph are divided into n groups of nodes by different positions of a certain symbol. The symbol can be any one of the n symbols used in the permutation. For example, if we pick symbol A when dividing a 4 dimension star graph, the first group will contain nodes: *abcd*, *acbd*, *adbc*, *abdc*, *acdb*, and *adcb* [See Figure 2].

Definition: Consider a star graph S_n of dimension n and any arbitrary symbol T. S_n can be divided into n groups of nodes according to different position of symbol T in the label. The *root group* consists of nodes that have symbol T at the first position. The *ith leaf group* consists of nodes that have symbol T at the ith position, $2 \leq i \leq n$.

Properties

1. The nodes in each *leaf group* form a star graph of dimension one less than the original star graph.
2. There does NOT exist a direct link between any two nodes in the *root group.*
3. For each node in the root group, it has exactly one link to some node from each of the leaf group.
4. There does not exist a direct link between any two nodes from different leaf groups, and any path between these two nodes always includes at least one node from the root group.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 267–268, 2012.

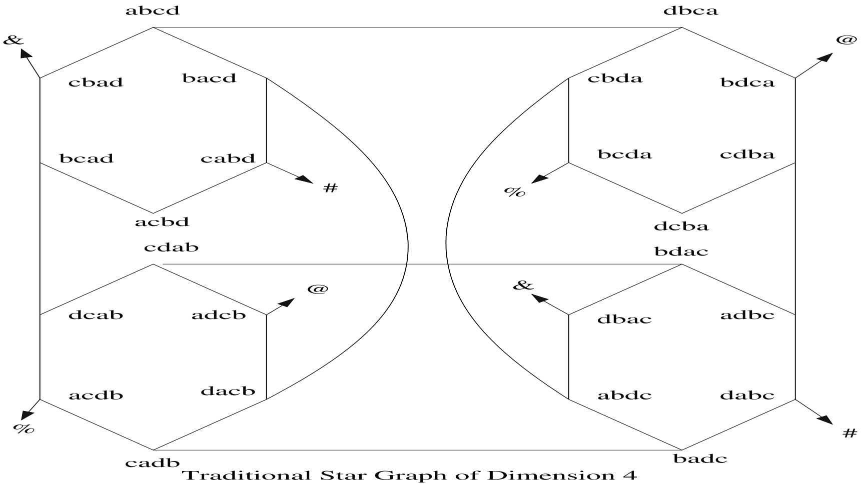


Fig. 1. Traditional Hierarchy in S_4

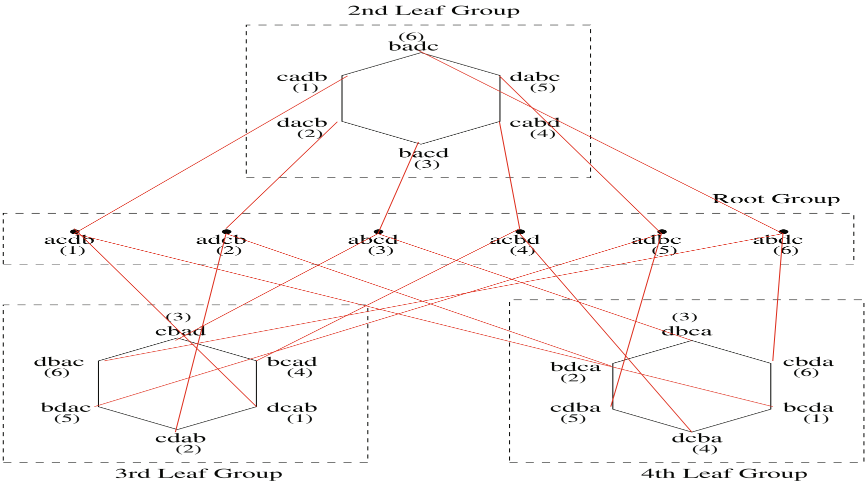


Fig. 2. Example of S_4 Grouped using Symbol a

Acknowledgement. This work is supported by NSF grant # CCF-0832582.

References

1. Akers, S.B., Krishnamurthy, B.: A group-theoretic model for symmetric interconnection networks. IEEE Transactions on Computers 38(4), 555–566 (1989)
2. McKinley, P.K., Xu, H., Esfahanian, A., Ni, L.M.: Unicast-based multicast communication in wormhole-routed networks. IEEE Transactions on Parallel and Distributed Systems 5(12), 1252–1265 (1994)

Incremental Discovery of Sequential Pattern from Semi-structured Document Using Grammatical Inference

Ramesh Thakur[1], Suresh Jain[2], and Narendra S. Chaudhari[3]

[1] IIPS, DAVV, Indore, India
[2] KCB Technical Academy, Indore, India
[3] Indian Institute Of Technology, Indore

Abstract. On the World Wide Web a large numbers of information is available in the form of semi-structured format. Knowledge discovery in semi-structured document has been recognized as promising task. Since semi structured document is typically hidden within HTML formatting intended for human viewing the details of which vary widely from site to site and frequent changes made to their formatting so we can't construct a global schema, discovery of interesting rules form it is complex and tedious process. Most of the existing system uses hand-coded wrappers to extract information, which is monotonous and time consuming. An intelligent and automated method is needed for their processing. Learning grammatical information from given sample of semi-structured documents has attracted lots of attention in the past decades. To understand "what say the data" is necessary to know the structure of data to discover the syntactic-semantic knowledge of its language.

The problem of learning the correct grammar for the unknown language form finite example of the language is known as grammatical inference problem. In automated grammar learning, the task is to infer grammar rules from given information about the target language. If example belongs to the target language it is called positive example otherwise it is called negative example. In this paper we propose a grammar inference methodology to automate the construction of grammar rules and facilitate the process of information extraction. We are using hybrid technique of association analysis and sequential algorithm to generate context free grammar rules from semi-structured document (HTML document).

Our algorithm that infers a sequential pattern from a sequence[1] of discrete HTML tags. The basic insight is that sub-string is selected on the basis of high support factor[2] by taking entire sentences into account. Which appears more frequently in string can be replaced by a grammatical rule that generate the sub-string, and this process is repeated many times, producing a single length rules of the sequence. The result is strictly a context-free grammar rules, which provide a compact summary of corpora that aids understanding of its properties.

Keywords: Knowledge discovery, sequence mining and grammar inference.

[1] Sequence: A sequence is an ordered list of alphabet symbols. We denote a sequence by <s1s2….sn) where si is a symbol. A sequence <a1, a2, …. An) is a subsequence of another sequence < b1, b2, ……bm> if there exist integers i1<i2<….in such that a1=bi1, a2=bi2, ……an=bin. For example, the sequence bob is subsequence of bbobbb.

[2] Support factor: The support Factor (SFβ) for sub-sequences in corpora C.

$$SF\beta = \sum_{i=1}^{N} \text{count of } \beta \text{ in sentence} \times \text{length of } / \text{ length of sentence}$$

Where N= number of sentences in Corpora C and β is a candidate sub-sequence for replacement.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, p. 269, 2012.

Group Associated Petri Nets in Bio Computing

K. Thirusangu[1], D. Gnanaraj Thomas[2], and B.J. Balamurugan[3]

[1] Department of Mathematics, SIVET College, Gowrivakkam, Chennai 600 073, India
kthirusangu@gmail.com

[2] Department of Mathematics, Madras Christian College, Chennai - 600 059, India
dgthomasmcc@yahoo.com

[3] Department of Mathematics, Agni College of Technology, Chennai 603 103, India
balamuruganbj@yahoo.com

A Petri net can be represented as a particular kind of bipartite graph consisting of two kinds of nodes called places and transitions. Directed arcs are used to connect places to transitions (output of places) and to connect transitions to places (input of places). The study of structural properties and behavioral properties for the bounded conflict free Petri net has been made utilizing siphons and traps [2].

Nitrogen is an essential element for all living things because it is a principal component of proteins and nucleic acids. Since enzymes carry out almost all of the chemical reactions in our body, every one can understand the importance of nitrogen [1].

In this paper we construct a bounded conflict free Petri net for a given group (Z_n, A) with a generating set. We prove that the resulting bounded conflict free Petri net associated with this group has subsets of places which are both siphon and trap whose input transitions equal to output transitions and both of them equal to the set of all transitions. This leads us to establish that the dual of this constructed bounded conflict free Petri net represents a Nitogen cycle.

Theorem 1. *There exists a bounded conflict free Petri net for every group (Z_n, A) where A a generating set.*

Proof. Let $(Z_{2k}; A)$ be a group with generating set $A = \{a, b, b+k\}$ where a, b, k are integers. Take the elements of Z_{2k} as the transitions of the bounded conflict free Petri net. Since Z_{2k} has $2k$ elements, we have $|T| = 2k$. Moreover, $A \subseteq Z_{2k}$. Now let us introduce places as follows. For every $t_i \in Z_{2k}$ and $s_k \in A$ such that $t_i + s_k = t_j (mod\, 2k)$, make a place p such that $^{\bullet}p = t_i$ and $p^{\bullet} = t_j$. Also deposit tokens in a place p if p is the input of $s_i + s_j$, for every $s_i, s_j \in A$. Since the generating set A has 3 elements, we have each transition has exactly 3 inputs and 3 outputs. Thus we have constructed a bounded conflict free Petri net with initial marking.

Example 1. The dual of bounded conflict free Petri net for the group $(Z_8; 5, 3, 7)$ is shown in Fig. 1 which describes the nitrogen cycle.

Here the places $p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7$ represent respectively atmospheric N_2, animals, nitrites NO_2, organic N_2 (proteins, amino acids), decomposers, ammonium NH_4^+, plants, nitrates NO_3. The set of transitions $t_1, t_2, \ldots, t_{24}$ are as follows.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 270–271, 2012.

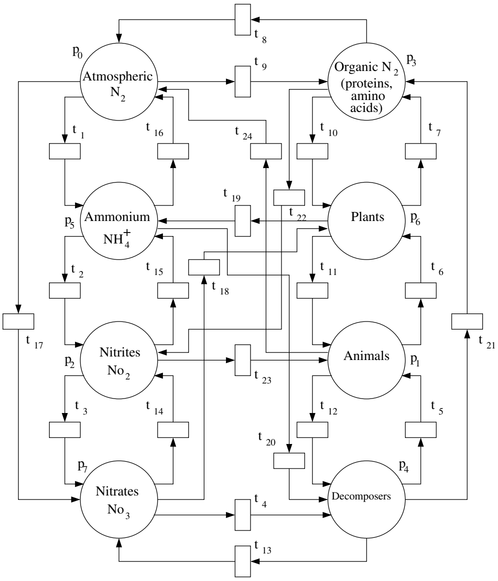


Fig. 1. The dual of bounded conflict free Petri net for Nitrogen cycle

t_1 - Nitrogen fixation by soil bacteria, t_2 - Nitrification by Nitrifying bacteria 1, t_3 - Nitrification by Nitrifying bacteria 2, t_4 - Decomposition by microbes 1, t_5 - Consumption of organic compounds, t_6 - Nitrogen uptake from the death and decay animals, t_7 - Death and Decay, t_8 - Vaporization process 1, t_9 - Nitrogen uptake by soil organism, t_{10} - Nitrogen uptake by plants, t_{11} - Consumption of plants by animals, t_{12} - Decomposition by microbes 2, t_{13} - Mineralization process, t_{14} - Denitrification 1, t_{15} - Ammonification process 1, t_{16} - Vaporization process 2, t_{17} - Oxidation process, t_{18} - Assimilation, t_{19} - Ammonification process 2, t_{20} - Salt formation, t_{21} - Reduction process, t_{22} - Nitrification by Nitrifying bacteria 3, t_{23} - Reassimilation, t_{24} - Denitrification 2.

References

1. Medici, L.O., Azevedo, R.A., Smith, R.J., Lea, P.J.: The influence of nitrogen supply on antioxidant enzymes in plant roots. Funct. Plant. Biol. 31, 1–9 (2004)
2. Thirusangu, K., Rajeswari, R., Balasangu, K.: Bounded Conflict Free Petri Nets Relative to (Zn, A). International Journal of Combinatorial Graph Theory and Applications (IJCGTA) 1(2), 99–108 (2008)

Concept Map Based Service Specification and Discovery*

Supriya Vaddi

Department of Computer and Information Sciences
University of Hyderabad, India
supriyavaddi@gmail.com

Web services have gained popularity with increasing number of businesses available on web. Huge increase in the number of services available on web makes service discovery a difficult problem. The process of discovery has graduated from syntax based to semantic based search. And there has been a spurt of research in specification and discovery of web services. The research work can be broadly classified to the following categories (1) WSDL Input Output Match (2) WSDL Input Output Precondition & Effect (Capability based) Match (3) Function semantics based Matching.

OWLS [1] has been successful for semantic specification and discovery of web services. This requires to define domain ontology that is agreed upon by service providers and consumers. Making an ontology for a domain brings in constraints in specifications, as all are limited to use the ontology that is agreed upon . The attempt has been made to overcome the limitation allowing users to generate new ontology combining the concepts available in ontology. WSMO [3] is one such attempt. WSMO is higher in abstraction than OWLS for providing means to generate new concept from given ones; thus allowing users unlimited capability for generating newer concepts.

This capability while is a boon, also can be a bane for difficulty in tracing concepts and getting a meaning as they are discrete without having defined inter-concept relations.

WSMO consortium is working on to standardize concept based service specification. And academia as well as professionals are adding new dimensions to WSMO [2], [4] . At this juncture we propose an idea of concept map that helps in providing a structural form to service specification that is understandable and traceable in process of service development as well as execution. In this paper we propose some relations that help to connect concepts and to generate concept maps for service specification. And these maps can also be used for service discovery.

Concept Map: Concept map is an association of basic concepts and that is achieved by operations viz Includes, a-kind-of(akf), Leads-to and Has-With are proposed here to specify certain aspects of web services; While the first three concepts help to build inter concept relations, the last one describes a concept

* This work is sponsored by DST-ITPAR (Indo Trento Project on Advanced Research) for 'Distributed Software Engineering', I sincerely thank Prof. Hrushikesha Mohanty for guiding me in the project.

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 272–273, 2012.

ascribing its attributes and corresponding value. Diagrammatically concepts are represented by solid oval shapes inscribed with concept names. Dashed oval shapes present attributes and their values. Below we present concept map specifying a restaurant service. The map is self describing with domain ontology and association operations.

ABC Restaurant service is an instantiation that includes instantiation of other services viz. Party Service, Dining Service, SocialFunction service. The links are labeled with operations and constraints of necessary (e.g a party service with bill amount ≥ 3000 rupees leads to a concessional service.) Further, these operations thrust a structure on concepts.

For example, *includes* operator brings in granularity and hierarchical structure into service specification. Whereas, '*a-kind-of*' operation defines peer relations proposing an alternative to a service as the operator establishes a similarity (it could have degree of similarity) between two services; hence are called peers. The structure emerging for these two operations enables one to traverse in top-to-bottom (or vice-versa) among concepts of different granularity and left-to-right (or vice-versa) among peer services existing at the same level of hierarchy. Again leads-to operator helps to specify emerging service (could be dynamically conditional) for leads-to operation.

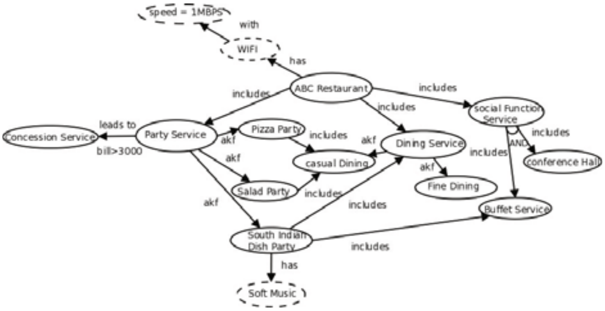


Fig. 1. Concept Map of an ABC Restaurant Service

Service discovery can be carried out by traversing concept maps and matching with user requirements. The proposed method enables user to discover alternative choices in case of failure in exact match.

References

1. Semantic Markup for Web Services (OWL-S), W3C Member Submission, November 22 (2004), http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/
2. de Bruijn, J., Lausen, H., Polleres, A., Fensel, D.: The Web Service Modeling Language WSML: An Overview. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 590–604. Springer, Heidelberg (2006)
3. Roman, D., Lausen, H., Keller, U., et al.: Web Service Modeling Ontology (WSMO). WSMO Final Draft October 21 (2006), http://www.wsmo.org/TR/d2/v1.3/
4. Wang, H.H., Gibbins, N., Payne, T.R., Redavid, D.: A formal model of the Semantic Web Service Ontology (WSMO). Information Systems 37(1), 33–60 (2012); available online since August 2011

Semantic Search Using Constrained Spread Activation for Semantic Digital Library

Sandeep Vasani, Mohit Pandey, and Minal Bhise

Dhirubhai Ambani Institute of Information and Communication Technology,
Gandhinagar, Gujarat-382007, India
{sandeep.vasani,mohitpandey31}@gmail.com,
minal.bhise@daiict.ac.in

Abstract. The work presented includes a prototype that demonstrates semantic search using constrained spread activation for relationship inference for semantic digital library domain. A java applet forms the basic user interface for this prototype. The user submits search query which is expanded using semantic digital library domain ontology. Based on the expanded query, initial nodes are activated and the activation is made to travel to other nodes using the constrained spread algorithm. The spread is constrained by making use of the distance constraint, which is supplied by the user. . The semantic search proposed here is the combination of spread activation techniques with traditional search engines techniques to obtain its results.

Keywords: Semantic Search, Ontology, Constrained Spread Activation.

1 Semantic Search System Implementation

The general architecture of the proposed system is shown in Figure 1. The initial query submitted by user to the search engine is refined and then used to activate Root Nodes which have activation level 100. From the root node activation is spread to all its connecting nodes. With each activated node its activation level is indicated in the final output. To control the spreading of activation the prototype uses distance constraint which puts a constraint on the number of hops from the root node. This process is shown in Figure 2.

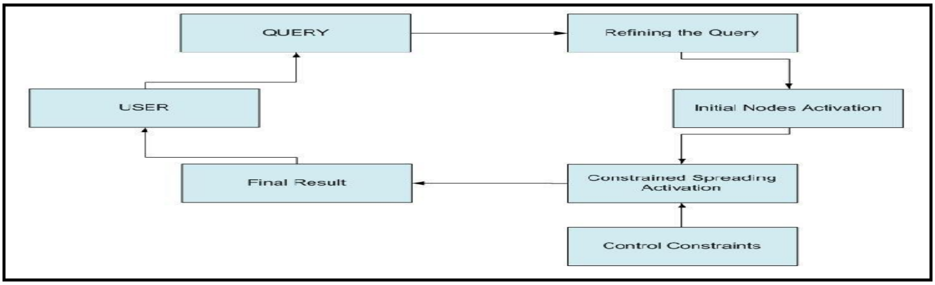
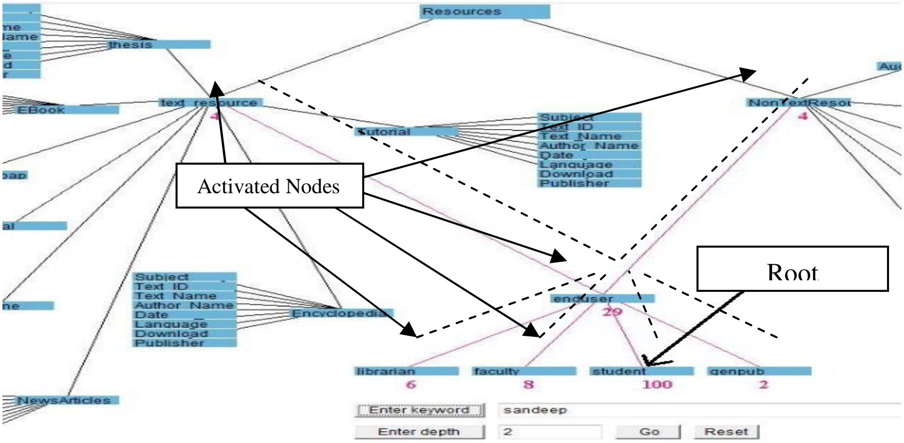


Fig. 1. Architecture of Proposed Model

R. Ramanujam and S. Ramaswamy (Eds.): ICDCIT 2012, LNCS 7154, pp. 274–275, 2012.

Constrained Spreading Activation in the semantic network is done using the breadth first traversal. First we need to find the root node(s). In our prototype the root nodes are found out from the search keywords. This is done by making use of the ontology. To show user that these nodes are root nodes and activated nodes we put the activation value near those nodes as shown in Figure 2.



Enter keyword sandeep

Enter depth 2

Fig. 2. Example of searching in the prototype

As input the user is asked to give depth till which s/he wants to spread activation. This input serves as the constraint to stop the spreading activation. For implementing the depth constraint we make use of difference in queue size. After finding all the root nodes we en-queue them and find the size of the queue. This is the size of queue at level 1. Next we perform the de-queue operation for size number of times. Whenever a de-queue operation is performed, we en-queue the node's neighbor. Once de-queue has been done for the required number of times, we find the size of the queue this is size of our level 2 and process goes on till we reach the required depth

For spread activation to neighboring nodes the following formula is used:

$$Activation_{i,j} = (a * weight_{i,j}) \; / \; (\sum_{k=1}^{j-1} weight_k + \sum_{k=j+1}^{n} weight_k)$$

Where **a**: Firing variable which depends on the activation level of the node, **activation**$_{x,y}$: Amount of activation sent from node x to node y when node x fires, **weight**$_{x,\, y}$: Strength of link between x and y , **n:** Number of nodes connected to node x.

2 Conclusion

A semantic search system to carry out search using constrained spread activation and enhance the search quality using ontology based information retrieval has been proposed here. Ontology is used to better understand the user intent and give more relevant answers. The usage of constrained spread activation gives the user the option of going for either a generalized or a specific search. The relevance feedback mechanism incorporated in the system ensures that the system does not get misdirected as the user queries sometimes could be ambiguous.

Author Index