Yifeng Zhou
Thomas Kunz (Eds.)

LNICST 223

Ad Hoc Networks

9th International Conference, AdHocNets 2017
Niagara Falls, ON, Canada, September 28–29, 2017
Proceedings

EAI

Springer

Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 223

More information about this series at http://www.springer.com/series/8197

Yifeng Zhou · Thomas Kunz (Eds.)

Ad Hoc Networks

9th International Conference, AdHocNets 2017
Niagara Falls, ON, Canada, September 28–29, 2017
Proceedings

Springer

Editors
Yifeng Zhou
Communications Research Centre
Ottawa, ON
Canada

Thomas Kunz
Carleton University
Ottawa, ON
Canada

ISSN 1867-8211 ISSN 1867-822X (electronic)
Lecture Notes of the Institute for Computer Sciences, Social Informatics
and Telecommunications Engineering
ISBN 978-3-319-74438-4 ISBN 978-3-319-74439-1 (eBook)
https://doi.org/10.1007/978-3-319-74439-1

Library of Congress Control Number: 2017964217

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The EAI International Conference on Ad Hoc Networks (AdHocNets) is a major annual international event in the ad hoc networking community. This year's AdHocNets conference was held in Niagara Falls, Ontario, Canada, eight years after its inauguration in the same city. The aim of the AdHocNets conferences is to provide a forum to bring together researchers from academia and industry as well as government to meet and exchange ideas and discuss recent research work on all aspects of ad hoc networking.

Over the last two decades, many efforts have been devoted to producing enormous contributions addressing the fundamental issues of ad hoc networking including modeling, protocol and algorithm design, security architectures and mechanisms, etc. Recent development in this area has focused more on addressing the challenges related to real and commercial applications including the Internet of Things (IoT), smart grid, and device-to-device (D2D) communications in 5G mobile systems. All these aim to extend the huge potentials of ad hoc networking beyond experiments and laboratories.

This volume of LNICST includes all the technical papers presented at AdHocNets 2017, which includes regular papers as well as invited papers from renowned researchers in this field. The invited papers provide visions, trends, challenges, and opportunities in the area of ad hoc networking and its applications. It is our hope that the proceedings will be a useful and timely reference for researchers in their effort to understand the real-world challenges for ad hoc networking, and to develop innovative solutions in addressing these challenges.

September 2017

Yifeng Zhou
Thomas Kunz

Organization

Steering Committee

Steering Committee Chair

Imrich Chlamtac	Create-Net and University of Trento, Italy

Steering Committee

Shiwen Mao	Auburn University, USA
Jun Zheng	Southeast University, China

Organizing Committee

General Chair

Yifeng Zhou	Communications Research Centre, Canada

Technical Program Committee Chair

Thomas Kunz	Carleton University, Canada

Publicity and Social Media Chair

Feng Yan	Southeast University, China

Publications Chair

Marc St-Hilaire	Carleton University, Canada

Web Chair

Jing Wang	University of Ottawa, Canada

Local Chair

Jiali Shang	Remote Sensing Applications Development, Agriculture and Agri-Food Canada, Canada

Conference Manager

Katarína Antalová	European Alliance for Innovation

Technical Program Committee

David Brown	Defence R&D Canada
Claude Chaudet	Telecom ParisTech, France
Yin Chen	Keio University, Japan

Stefano Chessa	University of Pisa, Italy
Omer Farooq	University College Cork, Ireland
Antoine Gallais	Université de Strasbourg, France
Raffaele Gravina	University of Calabria, Italy
Sunil Kumar	San Diego State University, USA
Chung-Horng Lung	Carleton University, Canada
St-Hilaire Marc	Carleton University, Canada
Nicholas Mastronarde	University at Buffalo, USA
Nathalie Mitton	Inria Lille - Nord Europe, France
Kamesh Namuduri	University of North Texas, USA
Amiya Nayak	University of Ottawa, Canada
Minming Ni	Beijing Jiaotong University, China
Symeon Papavassiliou	National Technical University of Athens NTUA, Greece
Scott Pudlewski	Air Force Research Laboratory
Pratap Kumar Sahu	University of Montreal, Canada
Ramon Sanchez-Iborra	Universidad Politecnica de Cartagena, Spain
Ronggong Song	DRDC Ottawa, Canada
Mehdi Sookhak	Carleton University, Canada
Alex Sprintson	Texas A&M University, USA
Yoshihiro Sugaya	Tohoku University, Japan
Manabu Tsukada	University of Tokyo, Japan
Paul Ward	University of Waterloo, Canada
Kui Wu	University of Victoria, Canada
Bo Xiao	East China Normal University, China
Salim Zabir	National Institute of Technology, Tsuruoka College, Japan

Contents

Underwater Networking

Security

Vehicular Networks

Ariel Networks and Routing

Cellular Networks, Sensor Networks

Underwater Networking

Doppler Effect in the Underwater Acoustic Ultra Low Frequency Band

Abdel-Mehsen Ahmad[1], Michel Barbeau[2], Joaquin Garcia-Alfaro[3](✉),
Jamil Kassem[1], Evangelos Kranakis[2], and Steven Porretta[2]

[1] School of Engineering, Lebanese International University, Bekaa, Lebanon
[2] School of Computer Science, Carleton University, Ottawa, ON K1S 5B6, Canada
[3] Telecom SudParis, CNRS Samovar, UMR 5157, Evry, France
joaquin.garcia_alfaro@telecom-sudparis.eu

Abstract. We address communications between Autonomous Underwater Vehicles (AUVs), Underwater Sensors (USs) and remote operators. We assume the use of acoustic waves. Due to the Doppler effect, the communication frequency depends on the relative motion between the participants. We are interested in the Ultra Low Frequency (ULF) range, from 0.3 to 3 kHz. We relate the Doppler effect to the half-power bandwidth, versus distance. Numeric simulations are conducted. We show that the Doppler shift is significant with respect to the half-power bandwidth in the ULF band, for long distance communications.

1 Introduction

Autonomous Underwater Vehicles (AUVs) and Underwater Sensors (USs) use acoustic waves to communicate. We are interested in the Ultra Low Frequency (ULF) range, 0.3 to 3 kHz (kHz), underwater communications. The ULF band, as suggested by Stojanovic [1], is interesting because the attenuation is lower, relative to higher frequencies. Hence, there is more potential for long range communications. For instance, Freitag *et al.* [2] have been able to make contact at a distance of 400 km at 900 Hz. On the other hand, the half-power bandwidth is narrow in the ULF band. As a consequence, solely extremely low rate data streams can be supported. Another communication impairment is the Doppler effect. It is created by relative motions between acoustic sources and receivers. Given the narrow half-power bandwidth and slow propagation speed of underwater acoustic waves, one may expect a significant Doppler effect in the ULF band. The goal of this work is to characterize the importance of the Doppler effect in various underwater communication scenarios in the ULF band. Some questions addressed are: What is the maximum Doppler shift that can be expected on underwater links in the ULF band? What is the maximum frequency drift that can happen during the reception of a data frame? Through a number of scenarios, we show that the Doppler shift is significant in the ULF band for long distances, relative to the narrow half-power bandwidth.

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 3–12, 2018.
https://doi.org/10.1007/978-3-319-74439-1_1

Section 2 provides background on ULF underwater acoustic communications. Section 3 discusses the Doppler effect. Section 4 presents our experimental scenarios and results. Section 5 concludes.

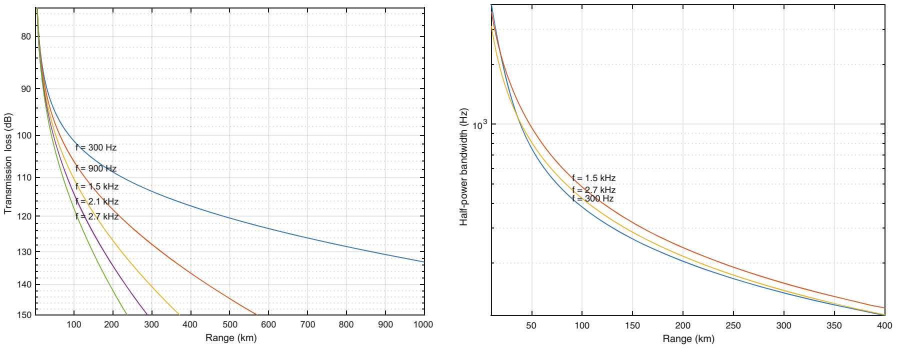
2 ULF Underwater Acoustic Communications

Attenuation is an important underwater acoustic communication impairment. The main causes are conversion of acoustic energy into heat and geometrical spreading. The magnitude of attenuation is represented in the Thorp's model [3–5]. For long distance underwater communications, the ULF band is preferable because there is less attenuation at the lower end of the acoustic spectrum.

Figure 1(a) plots the attenuation as a function of distance for selected frequencies in the ULF band. Realistically, for long range underwater acoustic communications, solely the use of low frequencies can be envisioned. For instance, Freitag *et al.* [2] have been able to achieve communication over a 400 km range at 900 Hz.

Another important fact is the gradient of the attenuation versus frequency. The transmission loss rapidly increases for higher frequencies. It limits the operating bandwidth. This constraint is captured by the concept of half-power bandwidth. The half-power bandwidth is commonly used to define cutoff frequencies and bandwidths of filters by using frequency response curves, using 3 dB points in the frequency response of a band-pass filter [6].

Figure 1(b) shows the half-power bandwidth for selected ULF frequencies versus distance. Firstly, the relationship between frequency, half-power bandwidth and distance is not linear. Secondly, at very long ranges (e.g., 400 km), the half-power bandwidth is very narrow, i.e., around 100 Hz.



(a) Attenuation.

(b) Half-power bandwidth.

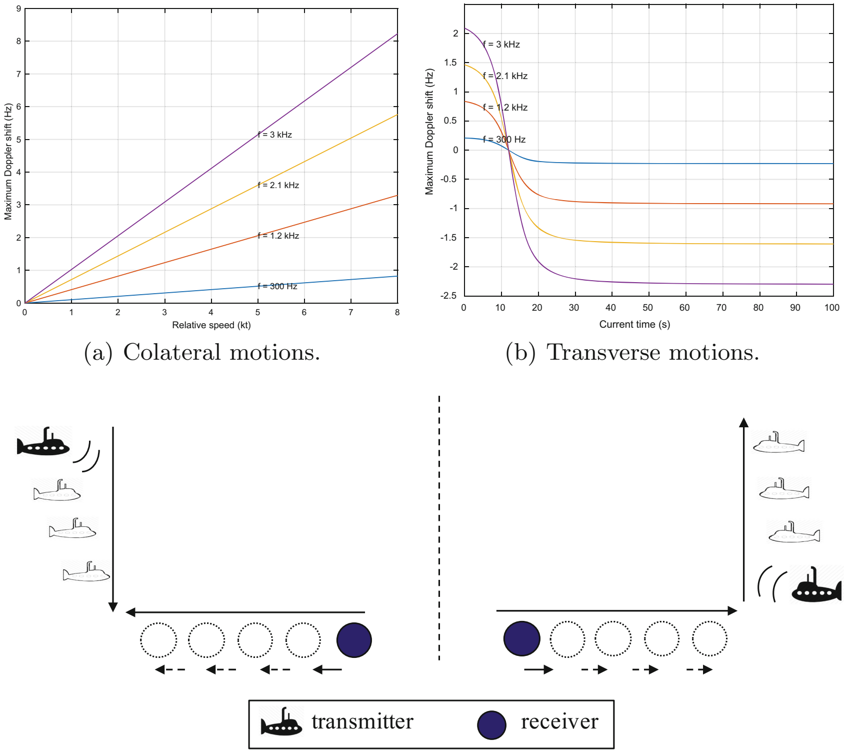
Fig. 1. (a) Attenuation and (b) half-power bandwidth for selected frequencies in the underwater acoustic ULF band. We can observe the relation between the half-power bandwidth and the frequency, with respect to the range. For a range less than 17 km, the half-power bandwidth is better for low frequencies (300 Hz) than for higher frequencies with the same range.

3 Doppler Effect

The Doppler effect shifts the frequency, from the receiver point of view, because of a transmitter-receiver delay change during data transmission. This happens because either the transmitter or the receiver are mobile. Their relative separation distance is not constant. Let v (m/s) be the relative velocity between a transmitter and a receiver. It is positive when they are getting closer, negative when moving away. Let c be the signal propagation speed (m/s). At nominal frequency f_0 Hz, the variation of frequency due to the Doppler effect is [7]:

$$\delta f = f_0 \frac{v}{c} \text{ Hz} \tag{1}$$

Figure 2 depicts the maximum Doppler shift for selected frequencies in the ULF range. We assume that the relative speed varies from zero to eight knots. This range is consistent with the values reported by Robert *et al.* [8] about speed range of AUVs. Figure 2(a) shows that the Doppler shift turns out to be



(a) Colateral motions.

(b) Transverse motions.

(c) Scenario for the transverse motion of a transmitter and a receiver.

Fig. 2. Maximum Doppler shift in the ULF range, assuming mobile transmitter and receiver moving either on the same axis (a) or in transverse directions (b). The scenario for the transverse movement between the transmitter and receiver is depicted in (c).

linear with respect to the relative speed, if we assume that both transmitter and receiver are moving along the same axis. Figure 2(b) makes the assumption that the transmitter and receiver move in transverse directions (i.e., transmitter and receiver move in opposite directions with respect to a reference axis) with constant speeds and as in the scenario shown in Fig. 2(c). After nine seconds, both of them arrive at the same point, where the Doppler shift becomes null. Then, they move away from each other, causing an increase in the Doppler shift.

4 Experimental Scenarios and Results

Experiments are conducted using numeric simulations via BELLHOP [9,10]. We assume situations in which one mobile transmitter (representing an AUV) is continuously transmitting acoustic waves at a specific frequency. At an initial distance, the waves are processed by an array of receivers (underwater sensors equipped with acoustic hydrophones). The array of receivers also move with respect to a mobility model (e.g., a sinusoidal movement). At each instance, one or more receivers process a series of multipath components that are summed up together at the receiver side. These multipath components consist of a straight line-of-sight (LOS) ray and multiple reflected and refracted rays. The rays are generated using BELLHOP. Each ray comes with different delay shifts, causing different frequency shifts. Our goal is to study variations of Doppler shifts and estimate the detection accuracy in a series of communication scenarios. The source code of the simulations is available online at http://j.mp/UWtmpgit. Next, we present our main scenarios.

4.1 Doppler Shift for a Transmitter-Receiver Pair

Figure 3(a) depicts our first scenario. It consists of two underwater devices: one acting as a transmitter (T) and the other acting as a receiver (R). R moves according to a sinusoidal model along the z axis (such that $z = A \sin\left(\frac{t \cdot \pi}{60}\right)$). The trajectory of T is based on the Caruso *et al.* model [11,12], which simulates a movement with one degree of freedom. The Doppler shift is derived according to Eq. (1) (c is 1500 m/s). The distance between the two devices is derived by substituting their three-dimensional coordinates in the equation $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$. Change of distances are used to compute the relative velocity between T and R. The data is produced at one sample per second.

Figure 4 plots the Doppler shifts for times $t = 0$ to $t = 50$ s. Figure 4(a), the amplitude A of R is 10 m. The figure shows a slight frequency shift, that is one Hz at its maximum. In Fig. 4(b), the amplitude A of R is 50 m. The Doppler shift becomes noticeable, as it peaks at 35 Hz.

Scenario 1.1 – Mobile Transmitter, Stationary Receiver. In Fig. 5(a), the transmitter is assumed to follow a sinusoidal motion with frequency $f_m = 0.1$ Hz and amplitude A of 10 m. The receiver is stationary. The transmitter is moving

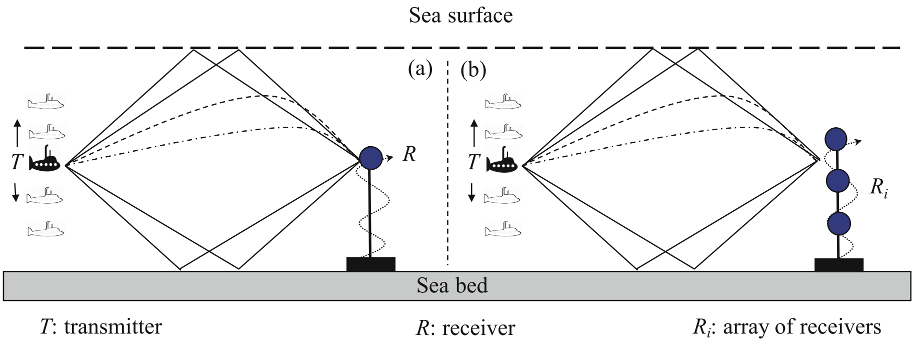
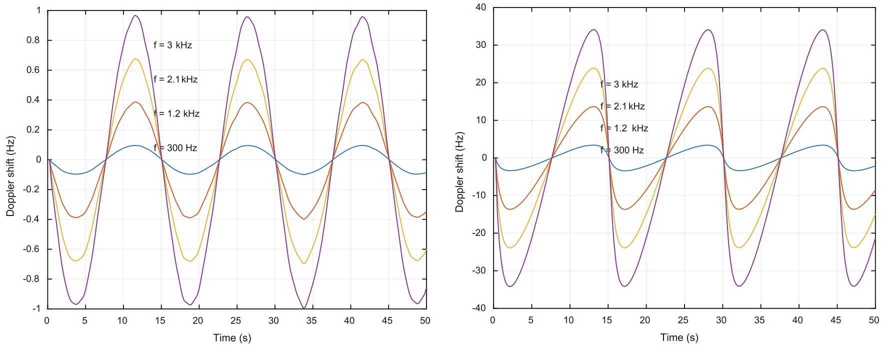


Fig. 3. Experimental scenarios. (a) Communication between one transmitter and one receiver. (b) Communication between one transmitter and one array of receivers. The receivers moves according to a sinusoidal model along the z axis. The transmitter moves according to the model in [11,12], which simulates a one-degree freedom of movement. Straight lines represent either reflected or refracted rays. Dashed lines represent eigenrays, i.e., no reflections nor refractions. Some videocaptures of the assumed mobility patterns are available at: http://j.mp/UWtransmitters and http://j.mp/UWreceivers.



(a) R's motion amplitude A is 10 m.

(b) R's motion amplitude A is 50 m.

Fig. 4. Doppler shift at different receiver positions for selected ULF frequencies. The receiver moves according to a sinusoidal law.

along a line with an invariant angle β relative to the x-axis. This line makes a variable angle α with a transmitter-receiver line. Since the transmitter is moving along an angle β with respect to the x-axis, we can compute the horizontal position x of the transmitter as follows:

$$x = \cos\beta \cdot A \cdot \cos\left(\frac{\pi t}{f_m}\right) \text{ m} \tag{2}$$

and its depth d as:

$$d = \sin\beta \cdot A \cdot \cos\left(\frac{\pi t}{f_m}\right) \text{ m} \tag{3}$$

The frequency at the receiver f_r is computed using the following formula:

$$f_r = \frac{c}{c - v_s \cos(\alpha)} \cdot f_s \text{ Hz} \tag{4}$$

where f_s is the frequency of the transmitter, c is the signal propagation speed (assumed to be 1500 m/s), and v_s is the velocity of the transmitter. The value of v_s is obtained by computing $\frac{\delta d}{\delta t}$, where δd is the change of distance during an interval of time t.

Scenario 1.2 – Mobile Receiver, Stationary Transmitter. In Fig. 5(b), the receiver is moving with velocity v_r along a line with constant angle β, relative to the x-axis. Variable α is the angle between this line and transmitter-receiver line. The receiver is assumed to follow a sinusoidal motion with frequency 0.1 Hz and amplitude of 10 m. The transmitter is stationary. Same operations as in Scenario 1.1 are computed, except for deriving the frequency at the receiver f_r, which is computed using the following equation:

$$f_r = \left(1 + \frac{v_r \cdot \cos\alpha}{c}\right) \cdot f_s \tag{5}$$

Figures 6 and 7 show the simulation results. Figure 6 shows the variation of both angle and velocity of movement. The moving receiver and moving transmitter scenarios produced the same results. Figure 7 shows the variation of the Doppler shift for four β angles: (0°, 40°, 80° and 90°). In Fig. 7(f) velocity is increased 40×.

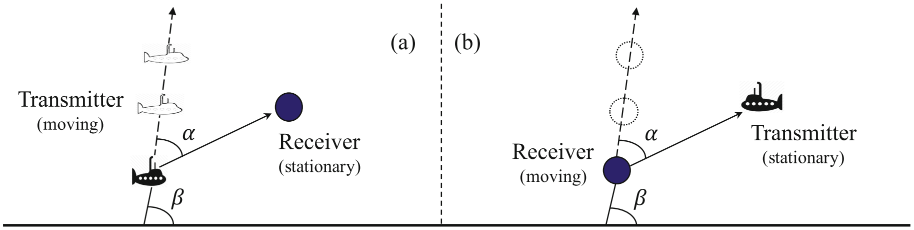
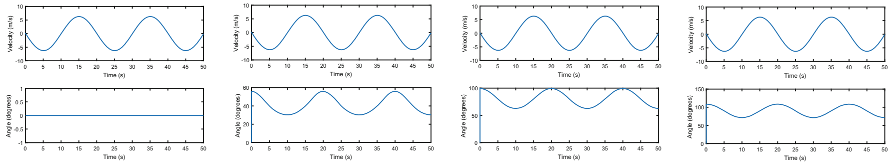
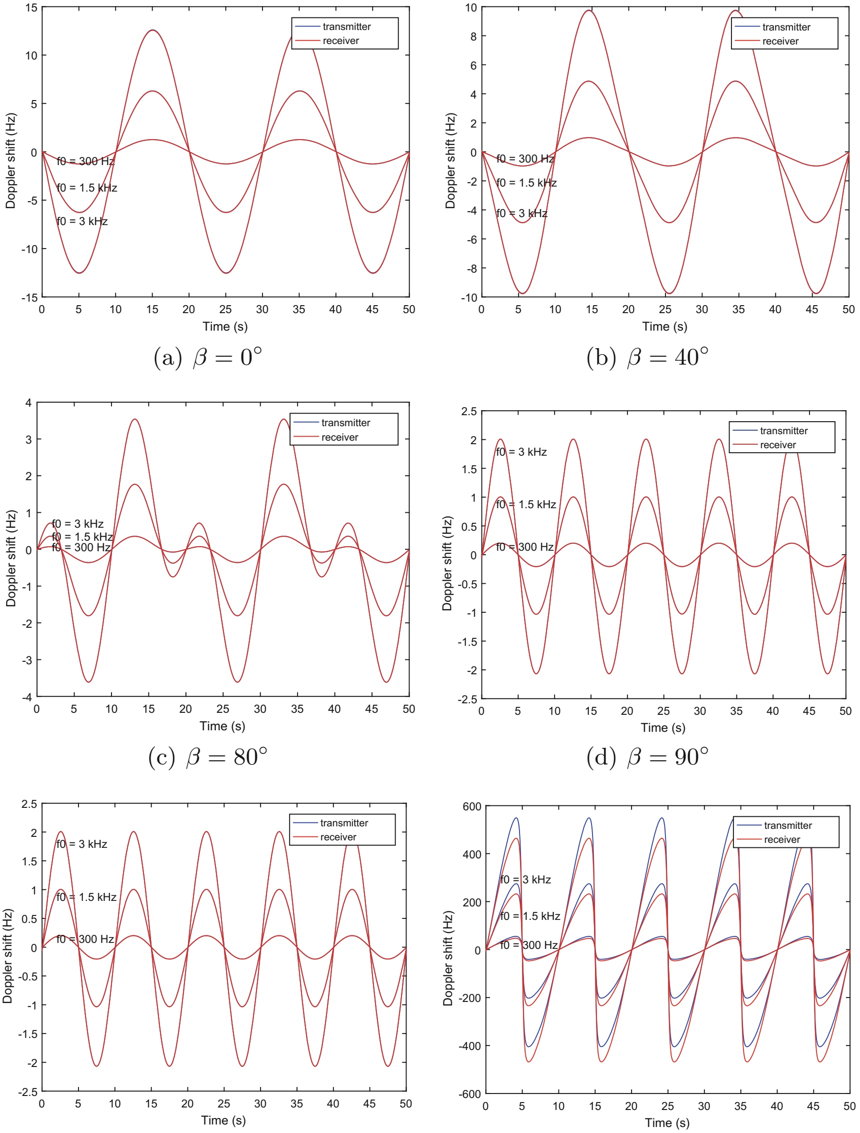


Fig. 5. (a) Scenario 1.1, moving transmitter and stationary receiver. (b) Scenario 1.2, moving receiver and stationary transmitter.



(a) $\beta = 0°$ (b) $\beta = 40°$ (c) $\beta = 80°$ (d) $\beta = 90°$

Fig. 6. Mobility characteristics for the Doppler shift experiments assuming 1.5 kHz for the frequency; and 0 to 90° as angle β.

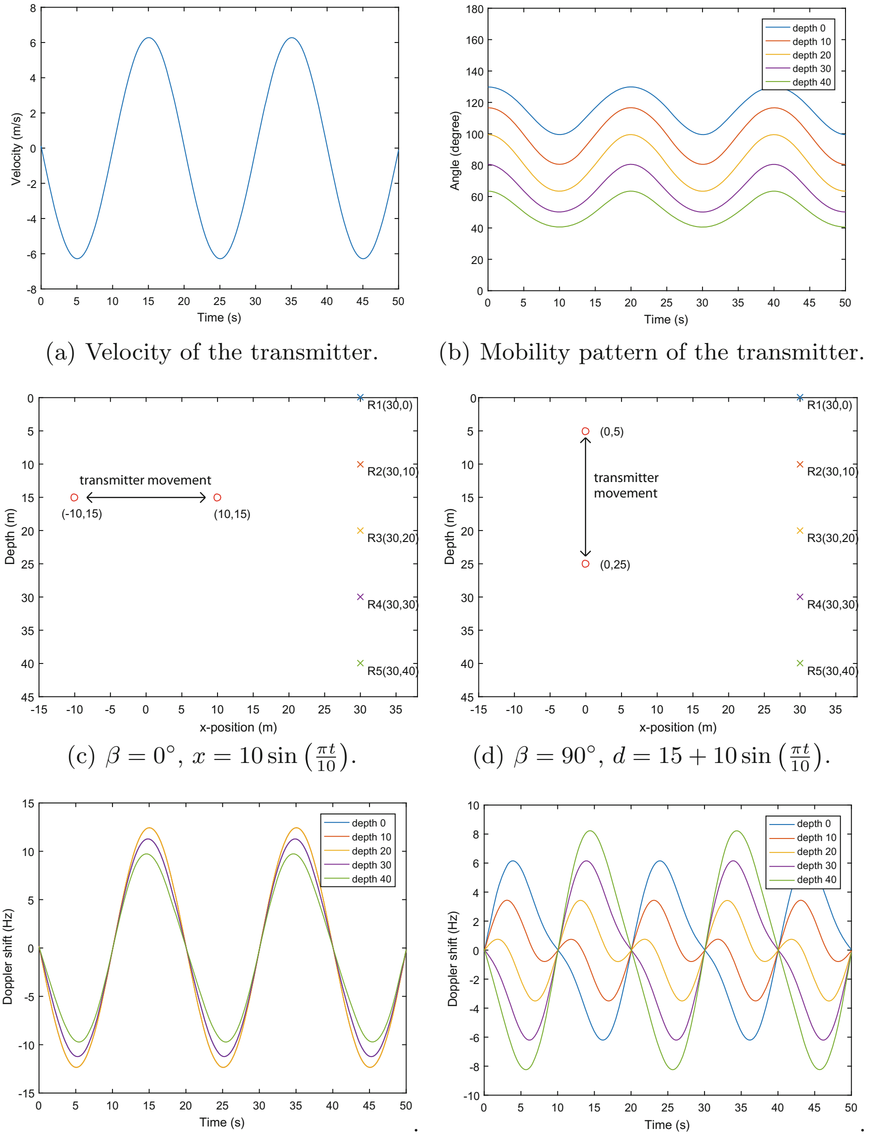


(a) $\beta = 0^\circ$ (b) $\beta = 40^\circ$

(c) $\beta = 80^\circ$ (d) $\beta = 90^\circ$

(e) Normal velocity (f) 40x velocity increase

Fig. 7. Doppler shift experiments at 1.5 kHz and angles 0° to 90°.



(a) Velocity of the transmitter.

(b) Mobility pattern of the transmitter.

(c) $\beta = 0°$, $x = 10\sin\left(\frac{\pi t}{10}\right)$.

(d) $\beta = 90°$, $d = 15 + 10\sin\left(\frac{\pi t}{10}\right)$.

(e) Shifts, $\beta = 0°$, $x = 10\sin\left(\frac{\pi t}{10}\right)$.

(f) Shifts, $\beta = 90°$, $d = 15 + 10\sin\left(\frac{\pi t}{10}\right)$.

Fig. 8. Doppler shifts affecting transmitters (a, b) moving along either the x-axis (c), (e) or y-axis (d), (f).

4.2 Doppler Shifts and Attenuation Between One Transmitter and Several Receivers

We study the Doppler shift assuming the existence of multiple receivers, as depicted in Fig. 3(b). Different scenarios are discussed.

Assuming that the transmitter is positioned at 15 m deep, five receivers are placed 30 m away from the transmitter, at depths zero, 10, 20, 30, and 40 m. The array of receivers is stationary. The following cases are considered:

1. The transmitter is moving along a line parallel to the x-axis, where $x = 10\cos(0.1\pi t)$ m. Depth is constant at 15 m.
2. The transmitter is moving along a line parallel to the vertical axis, where depth $d = 15 + 10\cos(0.1\pi t)$ m and $x = 0$ m.

Let v be the velocity of the transmitter. The following equation is used to compute the Doppler shift:

$$\delta f = \frac{c}{c - v\cos\alpha} \tag{6}$$

where α is the angle between the line along which the transmitter is moving and transmitter to receiver line.

Figure 8 shows the results of the simulations and the velocity and mobility patterns assumed during the experiments. When angle β is set to 90°, receivers are far from the transmitter, i.e., at distance greater than the amplitude of the sinusoidal motion of the transmitter, each receiver is either above or below the transmitter. If the transmitter is moving up, then it gets closer to the receiver placed at the higher depth. It moves away from the receivers placed at the bottom of the array. It is reflected in the Doppler shift. When the motion of the transmitter is along the x-axis, all the receivers experience similar delays. At depths 10 to 20 m, the Doppler shift is the same as at depths 0 and 30 m—since the transmitter is moving in the middle at depth 15 m.

5 Conclusion

We have addressed acoustic communications between AUVs, USs and remote operators. We studied scenarios comprising one transmitter and one or several receivers. Due to the mobility of nodes, the Doppler effect changes the communication frequency. We focused on the ULF band, i.e., the frequency range 0.3 to 3 kHz. Numeric simulations confirm the importance of the Doppler shift. We have a maximum Doppler shift of 10 Hz in the scenarios we studied. It is negligible for short and medium ranges. It is, however, significant with respect to the half-power bandwidth for long distance communications (400 km). It corresponds to 10% of the half-power bandwidth. Since attenuation also depends on frequency [3–5], a positive Doppler shift increases the frequency and augments the attenuation, and vice-versa. In our simulations, the Doppler effect on the attenuation bandwidth is not significant. The source code of the simulations is available online at http://j.mp/UWtmpgit.

References

1. Nordrum, A.: NATO Unveils JANUS, First Standardized Acoustic Protocol for Undersea Systems (2017). http://spectrum.ieee.org/tech-talk/telecom/wireless/nato-develops-first-standardized-acoustic-signal-for-underwater-communications
2. Freitag, L., Partan, J., Koski, P., Singh, S.: Long range acoustic communications and navigation in the Arctic. In: OCEANS 2015 - MTS/IEEE Washington, pp. 1–5, October 2015
3. Thorp, W.H.: Analytic description of the low frequency attenuation coefficient. J. Acoust. Soc. Am. **42**, 270 (1967)
4. Thorp, W.H.: Deep ocean sound attenuation in the sub and low kilocycle per second region. J. Acoust. Soc. Am. **38**(4), 648–654 (1965)
5. Thorp, W.H., Browning, D.G.: Attenuation of low frequency sound in the ocean. J. Sound Vib. **26**, 576–578 (1973)
6. Baisheng, W.: A correction of the half-power bandwidth method for estimating damping. Arch. Appl. Mech. **85**(2), 315–320 (2015)
7. Lurton, X.: An Introduction to Underwater Acoustics: Principles and Applications. Springer, Heidelberg (2002)
8. Button, R.W., Kamp, J., Curtin, T.B., Dryden, J.: A survey of missions for unmanned undersea vehicles (2009)
9. Porter, M.B.: The BELLHOP manual and user's guide (2011). http://oalib.hlsresearch.com/Rays/index.html
10. Rodriguez, O.C.: General description of the BELLHOP ray tracing program (2008). http://oalib.hlsresearch.com/Rays/index.html
11. Caruso, A., Paparella, F., Vieira, L.F.M., Erol, M., Gerla, M.: The meandering current mobility model and its impact on underwater mobile sensor networks. In: 27th Conference on Computer Communications (INFOCOM 2008), pp. 221–225. IEEE (2008)
12. Caruso, A.: Simple jet meandering model library (2014). https://github.com/antoniocaruso/smm

The Sound of Communication in Underwater Acoustic Sensor Networks

(Position Paper)

Michel Barbeau[1], Joaquin Garcia-Alfaro[2], Evangelos Kranakis[1](✉), and Steven Porretta[1]

[1] School of Computer Science, Carleton University, Ottawa, ON K1S 5B6, Canada
kranakis@scs.carleton.ca

[2] Telecom SudParis, CNRS Samovar, UMR 5157, Evry, France

Abstract. Underwater environments have never been much of a constraint to the rich animal life they support at all depths of our seas and oceans. Indeed, nature has taken advantage of this environment to develop a rich variety of efficient communication strategies through evolutionary change and adaptation. The wealth of knowledge to be discovered will continue to dazzle and fascinate the world. For underwater sensor network communication, acoustic signalling is the preferred choice for designers because sound propagation is the most efficient when compared to other forms, like thermal, light, and electromagnetic. It is within this *acoustic* environment that researchers have to innovate and develop new ideas and methodologies so as to advance the state-of-the-art. In this paper, several fundamental issues and connections are discussed that arise in the study of underwater wireless sensor networks. A variety of ideas and solutions for further research is proposed and fundamental issues in topology control, directional underwater transducers, and monitoring and surveillance are discussed.

Keywords: Directional hydrophone and vibrator
Monitoring and surveillance · Neighbour discovery
Underwater acoustic communications

1 Introduction

Sound is very important for communication in the animal world. It helps animals to become aware of events that occur all around, regardless of where attention is focused. With respect to their land counterparts, sea mammals are even more dependent on sound for communication and sensing because of the special circumstances involved in the nature of signals underwater affecting the propagation of light, smell, and other senses. One must take into account that light propagation suffers from scattering due to reflection and refraction. Smell

The research of Michel Barbeau and Evangelos Kranakis is supported in part by Natural Sciences and Engineering Research Council (NSERC) Discovery grants.

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 13–23, 2018.
https://doi.org/10.1007/978-3-319-74439-1_2

is affected by molecular diffusion due to temperature, viscosity of the fluid, and size of the particles. As a consequence, sight and smell could be ineffective and rather much less suited for communication in the seas when compared to sound. Sound has another advantage because water molecules lose less energy as they vibrate. This paper explores how sound can be used to effectively communicate and build underwater networks. In Sect. 2, we introduce the nature of sound in the underwater environment. We explore the transmission of messages using sound in Sect. 3. A model for directional underwater communications is presented in Sect. 4. We conclude in Sect. 5.

2 Nature of Communication Underwater

You can calculate the speed v (m/s) of sound by taking the square root of the ratio of the pressure p (Pa) of the medium, inside which it travels, divided by the density ρ (kg/m^3) of this medium, namely

$$v = \sqrt{\frac{p}{\rho}} \text{ m/s.} \tag{1}$$

In the air, it is approximately only 343 m/s (at sea level). However, despite the fact that the propagation of sound underwater is affected by temperature, salinity, hydrostatic pressure and other factors, its speed in the ocean varies from 1,450 m/s to 1,540 m/s. It is more than four times higher than its speed in the air. Also note how pressure affects the speed of sound. Approximately 1,6 m/s per 100 m downwards is added to the velocity due to the increase in hydrostatic pressure.

Mammalian evolution has created numerous adaptations so as to exploit the propagation of sound underwater. Acoustic communication in the seas is entirely different from the more familiar terrestrial. Moreover, in marine life, mechanisms used to produce sound vary widely even from one family of sea animals (such as whales, dolphins, and porpoises) to another. This is documented extensively in the scientific marine biology literature. For example, it is well known that the humpback whales are producing regular and predictable sounds known as *songs* to communicate male fitness to females. The clicking sequences of dolphins and sperm whales are thought to be individualized rhythmic sequences communicating the identity of a single mammal to others in its group. They allow groups to coordinate foraging activities. Furthermore, communication can reach large distances with sperm whales being the undisputed vocal champions that can give a powerfully deafening directional sonar of 240 dB.

One cannot but marvel at the astonishing variety of sound based communication mechanisms that have evolved throughout sea life to communicate, attract mates, defend territory, sense surroundings and find food [6]. (See Ref. [18] where you can play recordings of all kinds of underwater animals, from whales and shrimps to oysters.) Although whales can communicate long distance with their powerful sounds, at the opposite scale Patek [20,25] reports that the *spiny lobster* emits Near Field Communication (NFC) signals (that propagate no more than a meter) every time it throws off its exoskeleton. The very unique sound it generates (by using its body as a violin) protects the *naked* lobster against its enemies while at the same time the short distance of propagation prevents it from advertising its presence further away!

3 Transmission of Sound

What technical issues do we encounter in transmitting messages underwater? How can we take them into account and at the same time improve our communication capabilities? We discuss how sonar measurements are made underwater as well as the impact of waveguides (communication tunnels) for connectivity.

3.1 Sonar Measurements

Sonar (also called *echolocation*) refers to the principle of detecting and localizing objects by sound. When referring to animals, it is also called *biological sonar* or *biosonar*. SONAR is an acronym for SOund Navigation And Ranging [3]. It is a technique that uses underwater sound propagation to navigate, communicate with or detect objects (such as submarines and mines) on or under the surface of the water by projecting sound and detecting the echoes from the objects.

The key to measuring the intensity and pressure of acoustic waves is based on using the concept of decibel (dB). Since in underwater acoustics, the primary interest is often in ratios rather than in absolute quantities this gives a convenient way for expressing changes (usually large) in pressure. Given two powers P_1 and P_2 (Watts), with power ratio P_1/P_2, we use the *decibel* expression $10\log_{10}\left(P_1/P_2\right)$ dB. When an acoustic wave propagates in a medium, acoustic energy is being transmitted. The amount of energy per second crossing a unit area is called the *intensity* of the wave. The unit of intensity in underwater acoustics is defined as the intensity of a plane wave having a pressure p of one micropascal (μPa). The relationship between acoustic pressure p and intensity I (Watts/m^2) is $I = p^2/(\rho v)$ Watts/m^2, where ρ (kg/m^3) is the density of water and v (m/s) is the speed of sound.

The intensity ratio I_1/I_2 is defined in decibels similarly to the power ratio, i.e., the intensity ratio in dB is equal to $10\log_{10}\left(I_1/I_2\right)$ dB. The basic measurement in acoustics is based on pressure and not on intensity. Most hydrophones used in underwater measurements are sensitive to pressure, particle velocity, or pressure gradient. It follows from from the above that the pressure ratio in decibels is expressed as $20\log_{10}(P_1/P_2)$.

3.2 Impact of Temperature and Pressure on Sound

Underwater, propagation of sound is three dimensional. It propagates in all directions from its source. During transmission sound dissipates. Understanding its behaviour is complicated by such features as suspended particles, air bubbles, plankton, and even the swim-bladders of swimming fish.

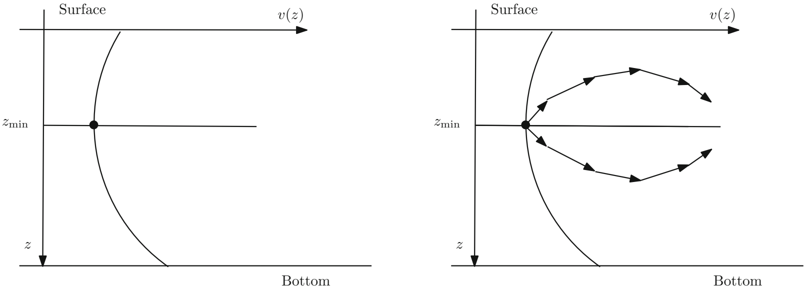


Fig. 1. *Left:* Diagram of the speed of sound $v(z)$ as a function of the depth z. *Right:* Due to refraction, a sound waveguide is formed whose walls delimit the propagation of sound when the emitting source is placed at depth $z_{\min}$.

The speed of sound in the ocean varies, see Eq. (1). It depends on temperature, salinity, pressure and other factors. Note that the pressure $p(z)$ (μPa) is monotone increasing as a function of the depth z(m). Also temperature affects the speed $v(z)$ of sound, as a function of the depth z. The interplay of these two factors affect $v(z)$ that has a representation resembling the plot depicted in the left part of Fig. 1. Closer to the surface, the speed $v(z)$ of sound is more affected by temperature. It decreases as we move deeper. As we move even deeper, the impact of pressure overtakes temperature. The speed of sound increases. Eventually, temperature and pressure balance out at a certain depth $z_{\min}$. The resulting speed of sound $v(z_{\min})$ at $z_{\min}$ is the minimum possible. This depth $z_{\min}$ also depends on the oceanic temperature. It can be up to 2 km in warmer waters while it is closer to the surface in the Arctic.

3.3 Impact of Refraction and Waveguides

Sound propagating in the ocean can sometimes be detected thousands of kms away from the source. Does the ocean contain a channel (or *acoustic waveguide*) through which sound can propagate with little attenuation? Indeed, it is not difficult to speculate that a natural channel is formed between the surface of water and bottom of the ocean. But what mechanism do sound waves obey in such long-distance propagation? The basic principle is that transmission of sound along a waveguide is based on the reflection of waves along its boundaries which prevents scattering. It would seem as if sound propagates along a *narrow tube* reflecting along its boundaries. But how are such gigantic waveguides formed underwater and where are its boundaries?

It turns out that refraction plays a crucial role in the formation of waveguides. Assume a sound source placed at depth $z_{\min}$, see right part of Fig. 1. Consider the sound beams emanating from it. Because of refraction, the propagation of the sound depends on the angle of the beam with respect to the horizontal. A beam propagating along a horizontal line is straight. A beam leaving $z_{\min}$ at an angle bends. However, since the speed of sound increases both up and down from the point $z_{\min}$, sound beams bend towards the horizontal. As a consequence this gives rise to a waveguide whose *walls* are formed by the layers of water at the depths where the sound beams reflect. For additional details see [2] (Chap. 3: The Oceanic Phone Booth) as well as Porretta's recent thesis [21].

To understand better the formation of waveguides, let $v(0)$ and $v(f)$ be the speeds of sound in the surface and bottom of the sea, respectively. It turns out that two types of waveguides may be formed depending on the relative sizes of the speeds $v(0)$ and $v(f)$.

Case $v(f) > v(0)$*:* This usually occurs in deep water. On the one hand, when the water surface is calm the sound is reflected from the surface but is refracted at sea bed. In fact one can use Snell's law to determine what portion of the sound beam is *captured* by the channel [2]. On the other hand, when the water surface is rough the sound scatters from it. The rays leaving the surface at large angles reach the bottom and are absorbed there. However, because of refraction the channel captures those rays that do not reach the rough surface [2].

Case $v(f) < v(0)$*:* This usually occurs in shallow water. In this case the sound refracted at the bottom does not reach the surface [2].

Schmidt and Schneider [10,23,24] documented the existence of a waveguide in the Beaufort Sea, called the Beaufort Lens. Due to a flow of warm water entering through the Bering Strait, from the Pacific Ocean. A sound speed minimum is created at low depth, around 80 m. Sound energy is trapped in the resulting waveguide. Long range (100 km) communication, without ice interaction, is possible using the waveguide.

The principle can be studied through simulation. Figure 2(Left) shows a Sound Speed Profile (SSP), artificially created to better illustrate the idea. It plots the speed of sound (x-axis) as a function of depth (y-axis). There is a sound speed minimum at 500 m deep. This minimum creates a waveguide in which the acoustic energy propagates without interaction with sea surface or sea bed. Figure 2(Right) shows the result of a BELLHOP [22] simulation of the

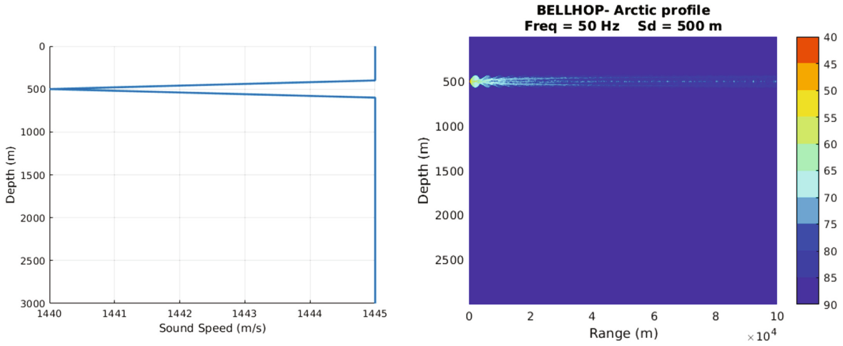


Fig. 2. *Left:* Sound speed profile with local minimum. *Right:* Transmission loss (dB) versus distance and depth, at 50 Hz and from zero to 100 km.

waveguide at the acoustic signal frequency 50 Hz. The source-receiver separation distance is from zero to 100 km. The x-axis represents range (m). The y-axis represents depth. A signal source is placed on the left, at range zero and depth 500 m. On a dB scale, color-coding represents signal attenuation as a function of location. In underwater, attenuation is proportional to frequency. Hence, attenuation is weaker on the left, staring at 70 dB on short range. The simulations demonstrate that signals propagating through a waveguide theoretically persist across long distances.

4 Directional Underwater Transducers

Directional antennae are widely used in wireless communication. They are versatile. They improve overall energy consumption [14]. This is rather easy to motivate. The transmission cost is proportional to the area covered by the antenna. Thus, the energy cost of an omnidirectional antenna with range r (m) is proportional to the area of a disk of radius r, that is, πr^2 m^2. By comparison the signal from a directional antenna of beam-width ϕ radians reaches much further, with the same energy consumption, namely it has range $r\sqrt{2\pi/\phi}$ m that, depending on the beam-width ϕ, can be significantly larger than r. They have numerous applications. They may enhance network capacity [9,26], reach further than omnidirectional antennae for detection and surveillance purposes, improve topology control and stability [8], and offer the potential for mitigating various security threats [11], just to mention a few applications.

A significant amount of research has been dedicated to the 2D model of directional antennae. In this model, the antennae are located in a planar terrain. To establish a network, antennae need to communicate with each other. To this end, two basic antennae communication models are employed. Consider two directional antennae: a sender and a receiver. In the *symmetric* model, communication is possible if the sender and receiver are within the range (determined by respective lobes) of each other, see [1,17]. In the *asymmetric* model, the sender can transmit directly a message to the receiver (provided the receiver is within the range of the sender) but the receiver may not be able to send directly a message to the sender, see [4,7,13,16]. In a way, the asymmetric model is less rigid than the symmetric one, but the receiver must seek a (alternate) path in the network if it also wants to talk to the sender.

4.1 3D Underwater Transducer Model

Underwater communications differ from classical wireless. Rather than electromagnetic waves, mechanical acoustic waves are used. The transducers, converting electrical energy to mechanical energy and vice versa, are vibrators and hydrophones. Transmission is done with mechanical vibrators. Reception is done with mechanical hydrophones. Hereafter, we discuss a 3D underwater transducer model. We identify vibrators and hydrophones. We model a *three dimensional* directional underwater transducer as a spherical sector of solid angle Ω (Fig. 3,

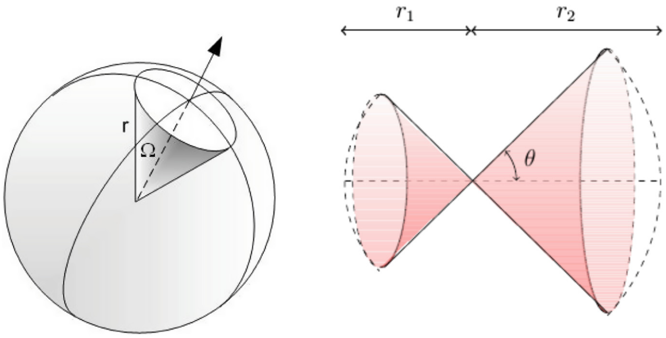


Fig. 3. Spherical underwater transducer radiation patterns. Solid (represented by Ω) and apex (represented by 2θ) angles in the 3D directional antenna model. The antenna in the left picture has a single lobe while in the right picture it has a double lobe.

left) and apex 2θ (Fig. 3, right). The *solid* angle Ω of a solid spherical sector is defined as the ratio of the area of the spherical surface and square of the radius of the sphere of which it forms part. The *apex* angle of a spherical sector with solid angle Ω is defined as the maximum planar angle between any two generatrices of the spherical sector. It is usually represented by 2θ, see [12] for additional details. The apex 2θ and solid angle Ω are related by the following important identity due to Archimedes. $\Omega = 2\pi(1 - \cos\theta)$; this Formula gives a method to compute a 3D angle with the help of a 2D angle.

4.2 Communication Model

How can we model communication using directional underwater transducers? We distinguish two types of connectivity: *symmetric* and *asymmetric*. In symmetric communication, two underwater transducers communicate if they are within range of each other, see Fig. 4(Left). This also means they can send messages directly to each other. Contrast this with asymmetric communication (Fig. 4(Right)) in which a vibrator S can talk to a hydrophone R only if there exists a sequence of $(\mathit{vibrator}, \mathit{hydrophone})$ pairs $S \rightarrow S'$ such that $S := S_1 \rightarrow S_2, S_2 \rightarrow S_3, \ldots, S_{k-1} \rightarrow S_k := R$ and moreover so that each hydrophone in this sequence is within the range of a vibrator. Thus, to establish bidirectional communication between S, R in the asymmetric communication model not only a *path* must be found between source S and destination R; in addition, a *path* must be found in the reverse direction from destination R to source S. Despite this difficulty it is still possible to provide algorithms that can establish bidirectional communication [4,7,13,16] with constant stretch factor.

4.3 Neighbour Discovery

How does a underwater node discover its neighbour(s)? The neighbour discovery process usually entails the exchange and subsequent confirmation of identities

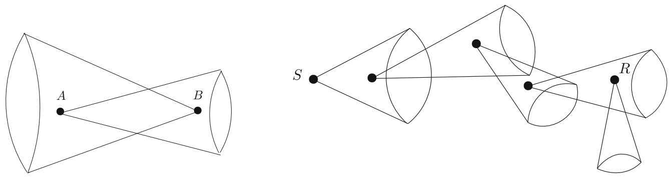


Fig. 4. Left: Symmetric communication with two directional underwater transducers centered at points A and B. Right: Asymmetric communication with directional underwater transducers: A directed communication path from vibrator S to hydrophone R.

between two adjacent nodes so that they can identify each other. To achieve this, omnidirectional underwater transducers must be, at a minimum, within range of each other (so that they can receive each other's messages). Thus, neighbour discovery for directional underwater transducers in the *symmetric* model requires not only that they must be facing each other but also be within range of each other. In the *asymmetric* model a communication path must be found between a sender and a receiver. Regardless of the communication model being used, the underwater nodes must execute an algorithm to discover their neighbour(s). To simplify matters, let us look first at 2D. Consider two directional nodes u and v with beam-width ϕ_u and ϕ_v, respectively. To communicate their underwater transducer must be facing each other. For each node u, let d_u be an integer delay parameter and k_u be defined so that $\phi_u = \frac{2\pi}{k_u}$ and consider Algorithm 1.

Algorithm 1. Underwater Transducer Rotation Algorithm $ARA(d_u, k_u)$

```
Start at a given orientation
while true do
    for $i \rightarrow 0$ to $d_u - 1$ do
        Send messages to neighbour(s)
        Listen for messages from neighbour(s)
    Rotate transducer beam one sector counter-clockwise
```

It can be shown (see [5] for details) that for a set S of synchronous nodes by appropriately choosing (either deterministically or at random) the delay d_u, for $u \in S$, Algorithm 1 ensures that every pair of underwater transducers within range will discover each other.

An approach to solving the neighbour discovery problem for underwater nodes is to adapt the previous approach, except that underwater transducer rotations must be done in 3D. The key idea is to use a partition of 3D space in a geodesic grid Moreover, like in Algorithm 1, underwater transducers would somehow have to rotate over a well-defined domain specified by a geodesic grid. The rotation mechanism (speed and direction of rotation) may depend on some knowledge of the environment and on the depth (see [15] for a related study).

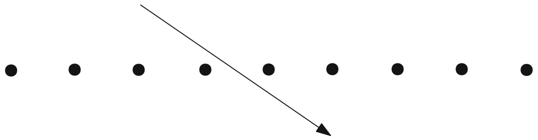


Fig. 5. An array of hydrophones and a passing underwater autonomous vehicle.

4.4 Monitoring and Surveillance

A potential application would be in establishing a wireless underwater networked system of communicating nodes to form an effective monitoring and surveillance network. Figure 5 depicts a linear array of underwater hydrophones and a passing autonomous vehicle. Each hydrophone has the ability not only to detect a passing autonomous vehicle, but also to discover its neighbours and transmit messages to them (other nodes within its range). Further, and unlike the scheme proposed by [19] which is static and not immune to transducer failures, the resulting array is dynamic and fault tolerant thus also adapting to a changing communication environment.

5 Conclusion

Research in UWANs requires a multidisciplinary approach involving scientists and engineers of widely varying academic backgrounds, experience and expertise. In this paper we looked at some characteristics of underwater communication and how they can be used so as to develop methodologies for better and more reliable connectivity. Further, we discussed the possibility of designing a wireless networked system based on underwater hydrophones to support monitoring and surveillance services. The ultimate goal would be to aid the design of surveillance underwater wireless acoustic networks in (harsh) underwater environments.

References

1. Aschner, R., Katz, M.J., Morgenstern, G.: Symmetric connectivity with directional antennas. Comput. Geom. **46**(9), 1017–1026 (2013)
2. Aslamazov, L., Varlamov, A.: The Wonders of Physics. World Scientific Publishing Co., Inc., Singapore (2012)
3. Au, W.W.L.: The Sonar of Dolphins. Springer, New York (2012). https://doi.org/10.1007/978-1-4612-4356-4
4. Caragiannis, I., Kaklamanis, C., Kranakis, E., Krizanc, D., Wiese, A.: Communication in wireless networks with directional antennas. In: Proceedings of the SPAA, pp. 344–351 (2008)
5. Du, J., Kranakis, E., Ponce, O.M., Rajsbaum, S.: Neighbor discovery in a sensor network with directional antennae. In: Erlebach, T., Nikoletseas, S., Orponen, P. (eds.) ALGOSENSORS 2011. LNCS, vol. 7111, pp. 57–71. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28209-6_6

6. Durrani, M., Kalaugher, L.: Furry Logic: The Physics of Animal Life. Bloomsbury Sigma (2017)
7. Eftekhari Hesari, M., Kranakis, E., MacQuarrie, F., Morales Ponce, O.: Strong connectivity of sensor networks with double antennae. Theor. Comput. Sci. **610**, 192–203 (2016)
8. Gupta, H., Kumar, U., Das, S.R.: A topology control approach to using directional antennas in wireless mesh networks. In: IEEE International Conference on Communications, vol. 9(06), pp. 4083–4088 (2006)
9. Gupta, P., Kumar, P.R.: The capacity of wireless networks. IEEE Trans. Inf. Theory **46**(2), 388–404 (2000)
10. Howe, T.: A modal analysis of acoustic propagation in the changing arctic environment. Master's thesis, MIT, Department of Mechanical Engineering (2016)
11. Hu, L., Evans, D.: Using directional antennas to prevent wormhole attacks. In: Network and Distributed System Security Symposium (NDSS) (2004)
12. Kranakis, E., Krizanc, D., Modi, A., Morales Ponce, O.: Connectivity trade-offs in 3D wireless sensor networks using directional antennae. In: 25th IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2011, Anchorage, Alaska, USA. Conference Proceedings, 16–20 May 2011, pp. 345–351 (2011)
13. Kranakis, E., Krizanc, D., Morales Ponce, O.: Maintaining connectivity in sensor networks using directional antennae. In: Nikoletseas, S.E., Rolim, J.D.P. (eds.) Theoretical Aspects of Distributed Computing in Sensor Networks. Monographs in Theoretical Computer Science, pp. 59–84. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-14849-1_3
14. Kranakis, E., Krizanc, D., Williams, E.: Directional versus omnidirectional antennas for energy consumption and k-connectivity of networks of sensors. In: Higashino, T. (ed.) OPODIS 2004. LNCS, vol. 3544, pp. 357–368. Springer, Heidelberg (2005). https://doi.org/10.1007/11516798_26
15. Kranakis, E., MacQuarrie, F., Travizani Maffra, I.K., Morales Ponce, O.: Strong connectivity of wireless sensor networks with double directional antennae in 3D. In: Cichoń, J., Gębala, M., Klonowski, M. (eds.) ADHOC-NOW 2013. LNCS, vol. 7960, pp. 257–268. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39247-4_22
16. Kranakis, E., MacQuarrie, F., Morales Ponce, O.: Connectivity and stretch factor trade-offs in wireless sensor networks with directional antennae. Theor. Comput. Sci. **590**, 55–72 (2015)
17. Montemanni, R., Gambardella, L.M.: Minimum power symmetric connectivity problem in wireless networks: a new approach. In: Belding-Royer, E.M., Al Agha, K., Pujolle, G. (eds.) MWCN 2004. IFIP AICT, vol. 162, pp. 497–508. Springer, Boston (2005). https://doi.org/10.1007/0-387-23150-1_42
18. University of Rhode Island Graduate School of Oceanography. Discovery of sound in the sea. http://www.dosits.org/. Accessed 03 June 2017
19. Otnes, R., Voldhaug, J.E., Haavik, S.: On communication requirements in underwater surveillance networks. In: OCEANS 2008-MTS/IEEE Kobe Techno-Ocean, pp. 1–7. IEEE (2008)
20. Patek, S.N.: First-person: The benefits of "strange" science, 14 March 2016. Duke Magazine: http://dukemagazine.duke.edu/. Accessed 03 Mar 2017
21. Porretta, S.: Environmental communication optimization in underwater acoustic sensor networks, Masters Thesis in Computer Science, Carleton University (2017)
22. Porter, M.B.: The BELLHOP manual and user's guide: Preliminary draft (2011)

23. Poulsen, A., Schmidt, H.: Acoustic noise properties in the rapidly changing Arctic Ocean. In: Proceedings of the 22nd International Congress on Acoustics (ICA), pp. 1–4, August 2016
24. Schmidt, H., Schneider, T.: Acoustic communication and navigation in the new arctic; a model case for environmental adaptation. In: 2016 IEEE Third Underwater Communications and Networking Conference (UComms), pp. 1–4, August 2016
25. Staaterman, E.R., Claverie, T., Patek, S.N.: Disentangling defense: the function of spiny lobster sounds. Behaviour **147**(2), 235–258 (2010)
26. Yi, S., Pei, Y., Kalyanaraman, S., Azimi-Sadjadi, B.: How is the capacity of ad hoc networks improved with directional antennas? Wireless Netw. **13**(5), 635–648 (2007)

Applying Utility Theory to Improve Autonomous Underwater Vehicle Mission Playload Planning and Replanning

Valerie Winschel(✉)

George Mason University, 4400 University Drive, Fairfax, VA 22030, USA
valerie.winschel@gmail.com

Abstract. This work presents a method by which utility theory can be applied to the payload decision making processes of autonomous underwater vehicles (AUV) for mission planning and replanning purposes. Such an application allows AUVs to determine the 'best' payload to use for a specific mission without operator intervention, thus improving AUV reliability while the vehicle is out of communication range from the operator. Because 'best' is subjective, focusing on relevant payload attributes and tailoring these functions to individual operator preferences ensures a unique vehicle makes decisions that align with a unique operator's preferences. The creation of these functions is an iterative process that involves interviewing an individual operator to determine the form and weight of that operator's preferences based upon theoretical payload attributes, followed by the testing of the resulting equation using actual payload attributes. Contained in this paper are example utility functions that take into consideration three attributes each for describing the decision making preferences for three AUV operators when determining the appropriate side scan sonar to use to perform a specific seabed imagining mission. These three functions were tested and determined to produce results that align with the individual operators' preferences, thus validating the appropriateness of these equations for these operators on this mission.

Keywords: Autonomous underwater vehicle · AUV
Unmanned underwater vehicle · UUV · Decision theory · Utility theory
Planning

1 Introduction

Militaries, commercial vendors, and research organizations are increasingly relying on unmanned vehicles to perform missions that are dull, dirty, dangerous, or otherwise cost prohibitive. These vehicles perform missions in the air, on land, on water, and undersea. Whereas surface and air vehicles can readily communicate with external information sources, such as human operators or satellites, undersea vehicles have limited communication capabilities, due to the electromagnetic signal-attenuating properties of water [1]. As a result, underwater vehicles not otherwise tethered to an operator must be capable of autonomous decision making.

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 24–33, 2018.
https://doi.org/10.1007/978-3-319-74439-1_3

To program an AUV with a mission, a human operator inputs parameters into the vehicle's controller, prior to launch. These parameters may include: type of mission, actions to be performed, payloads to be used, and mission location, among others. The vehicle uses this input to plan and continuously replan the mission as the missions progresses. The vehicles are, however, unable to plan or replan in a manner that violates these operator-provided inputs. In the event the specified payload fails during a mission, therefore, the AUV is generally unable to replan a mission around an alternate payload. When these payload failures occur, the AUV instead aborts that portion of the mission and either moves to another, pre-defined mission or returns to the surface of the water to await further instruction from a human operator.

Aborted missions create sunk cost or otherwise fail to capitalize on the benefits of the original mission. To minimize the occurrence of these aborted missions, this paper presents a method for allowing the vehicle, rather than the operator, to decide what payload to use for a given mission using mission-specific utility functions tailored to the unique operator, based upon previously-established decision making preferences of that operator.

2 Related Work

Work in the area of mission planning and replanning around a degraded payload has been conducted by researchers at Heriot-Watt University [1, 2]. One of the most comprehensive of these published works is Pedro Patron's 2010 doctoral thesis, from which many other works followed [3]. Much of Patron's work and the work of his coauthors focuses on semantic-based goal-oriented mission replanning around a degraded payload, specifically, a side looking sonar with one of two transducers failed [1–4]. In various published works, Patron and his coauthors propose approaching mission planning around a completely failed payload by searching for redundant components or platforms, however, no significant work has been published using this approach [3].

3 Approach

The focus and approach of this paper is the use of known attributes of possible mission payloads, operator-defined missions, and previously-defined operator preferences to create a set of utility functions that can order, from most preferred to least preferred, alternate payloads for a series of AUV missions. Because 'best' is subjective, these sets of equations will be tailored to individual operators so that a given AUV makes the 'best' decision for its operator. To accomplish this, an operator is interviewed to determine the form of his preferences for a specific attribute of a payload. He is then interviewed to determine the weight of his preferences for various attributes relative to others. Combined, these interviews allow for the creation of a utility function for a specific mission. The validity of this function is tested by asking the operator to rank example payloads, while also having the function calculate the ranks of these payloads. If the operator and equation do not provide the same results, the process is repeated, until the results match. Once the equation and operator provide the same results, the equation is sufficient at replicating operator decision making preferences for that mission.

4 Mission Definition

To provide the vehicle with the flexibility to plan a mission around a payload or replan a mission around a failed payload, the AUV must be programmed with a mission goal or end state, rather than a specific payload and associated payload action. By applying a goal-specific utility function to that mission, the vehicle will then be able to determine which payload will best allow it to meet that goal. How well a payload performs an action to accomplish a mission goal depends on what that goal is. As such, each goal will have a unique utility function taking into consideration a unique combination of attributes—or technical specifications—that are relevant for that goal. The vehicle can be programmed with a series of these utility functions. After an operator has defined the mission goal, the vehicle can then select the proper function for the decision to be made, apply the function, and calculate the highest-rated payload for mission performance. For multi-attribute utility theory with mutual preferential independence, the general form of a utility function is:

$$\left(\mathrm{x}_1, \ldots, \mathrm{x}_n\right) = \sum\nolimits_{i=1}^{n} \mathrm{k}_i * \mathrm{u}_i\left(\mathrm{x}_i\right) \tag{1}$$

where k_i is a scaling constant and $\mathrm{u}_i\left(\mathrm{x}_i\right)$ is a function of each attribute i [5]. To determine the form of each $\mathrm{u}_i\left(\mathrm{x}_i\right)$, utilities of 0 to 1 (i.e. least useful to most useful) are assigned for the range of possible attribute values. The utility mid-value point (i.e. the amount of an attribute where utility equals 0.5) is identified, based upon the preference of the operator for whom the function is being developed. This process is repeated for mid-value points for the upper and lower halves, quarters, eighths, and so on until a sufficient amount of precision is captured such that a trendline can be fitted through these mid-value points, the equation of which, $\mathrm{u}_i\left(\mathrm{x}_i\right)$, is the normalized utility function for that attribute. This process is repeated for all attributes, providing the basis for a utility function.

Next, a value for each k_i must be determined. To do this, a set of i equations is needed to solve for all constants. For the first equation, the rules of additive value due to mutual preferential independence provide that the sum of all values of k equals 1. Two attribute profiles for hypothetical payloads are then compared. The value of one attribute in one profile is manipulated until indifference (i.e. the point at which improving one profile or another is equally valued) is identified. This process is repeated until the set of functions can be solved for all values of k [5].

Once all equations for $\mathrm{u}_i\left(\mathrm{x}_i\right)$ and all values for k_i are known, the goal-specific mission utility function is created. By entering attribute values for the payloads under consideration into this equation, the utilities for those payloads can then be calculated. The payload with the highest utility is ‘best’ and the vehicle can plan or replan the mission around this payload.

5 Results

Although a complete set of equations for all possible missions spanning all applicable attributes would be required to allow a vehicle to autonomously plan and replan all possible missions around all available payloads for one operator, this paper establishes one utility function for one mission goal based upon three attributes to demonstrate the viability of applying utility theory to AUV decision making. The process is repeated for a total of three operators to show that different operators have different preferences.

5.1 General Form of a Utility Function

To create an example utility function, a goal must be defined, such as: imaging a given area of the flat ocean floor via survey to locate an object the size of a small automobile using side scan sonar, while traveling at 4 knots and maintaining an elevation of 100 m. Next, relevant attributes that influence the decision are considered, such as sonar frequency, transmit bandwidth, and ping rate [6, 7]. Assuming a scenario in which all sonars are otherwise equal, the general form of the overarching utility function will take the form:

$$\mathrm{u} = \left[\mathrm{k}_F * \mathrm{u}_F(\mathrm{F})\right] + \left[\mathrm{k}_P * \mathrm{u}_P(\mathrm{P})\right] + \left[\mathrm{k}_B * \mathrm{u}_B(\mathrm{B})\right] \tag{2}$$

5.2 Operator-Specific Utility Function

For this paper, three AUV operators were interviewed to determine their payload decision making preferences. Operator 1 has project engineer experience with the U.S. Navy's Salvage and Diving Command. Operator 2 has technical engineering experience with the U.S. Naval Sea Systems Command. Operator 3 has project management experience with a commercial vehicle operation company. The operators were presented with the abovementioned assumptions and attributes under consideration, with all other attributes behind held constant. Given these assumptions and using the technical specifications of commercially available sonars to approximately bound the equations, attribute-specific utility functions for Operator 1 were plotted as shown in Figs. 1, 2, and 3:

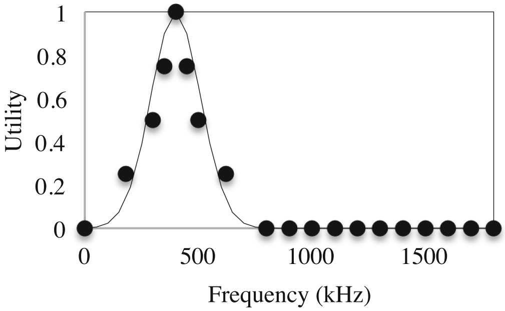


Fig. 1. Operator 1 frequency utility as a function of frequency (kHz).

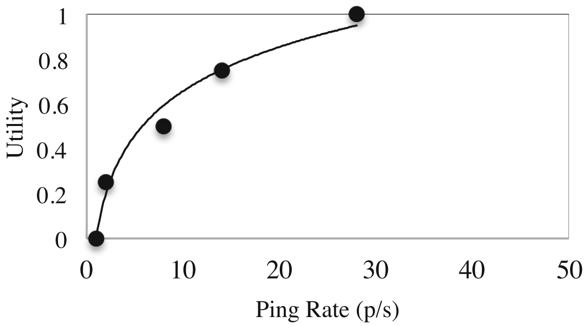


Fig. 2. Operator 1 ping rate utility as a function of ping rate (ping/s).

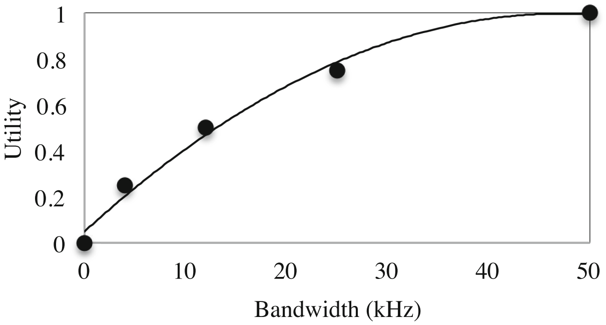


Fig. 3. Operator 1 bandwidth utility as a function of bandwidth (B).

The equations for the trendlines fitted through these plots are:

$$u_F = e^{\frac{-(F-400)^2}{24,000}} \tag{3}$$

$$u_P = \begin{cases} -0.0004B^2 + 0.04B + 0.05 & \text{for } B \leq 20 \\ 1 & \text{for } B > 20 \end{cases} \tag{4}$$

$$u_B = 0.28 \ln P + 0.004 \tag{5}$$

For Operator 2, these functions are represented by Figs. 4, 5, and 6:

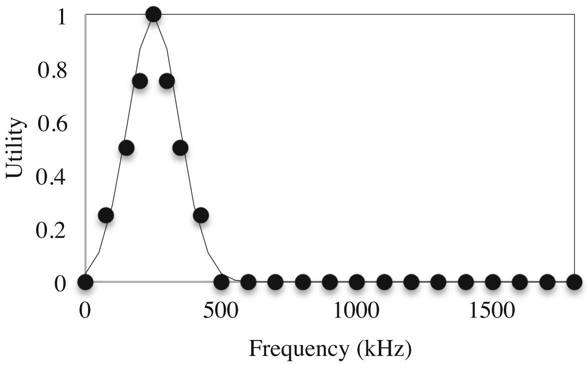


Fig. 4. Operator 2 frequency utility as a function of frequency (kHz).

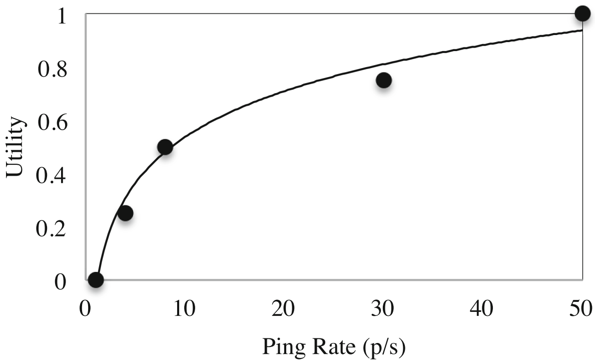


Fig. 5. Operator 2 ping rate utility as a function of ping rate (ping/s).

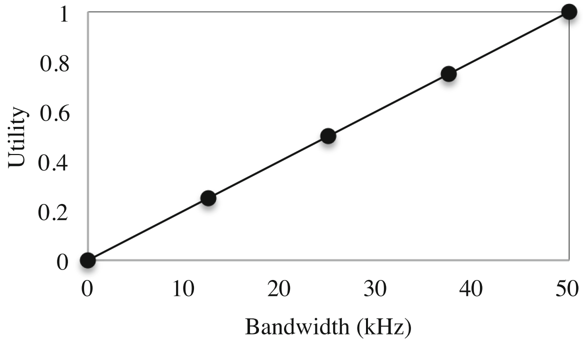


Fig. 6. Operator 2 bandwidth utility as a function of bandwidth (B).

And:

$$u_F = e^{\frac{-(F-250)^2}{18,000}} \tag{6}$$

$$u_P = 0.25\ln P - .04 \tag{7}$$

$$u_B = 0.02B \tag{8}$$

For Operator 3, these plots and functions are depicted in Figs. 7, 8, and 9:

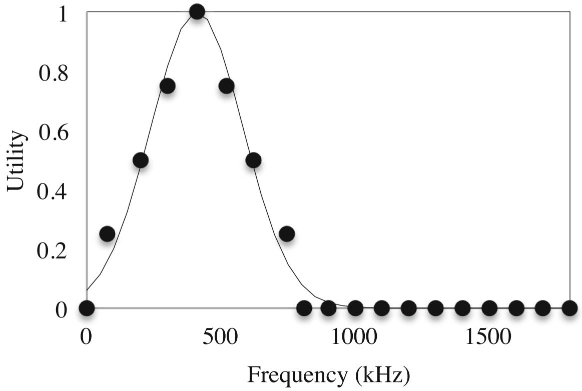


Fig. 7. Operator 3 frequency utility as a function of frequency (kHz).

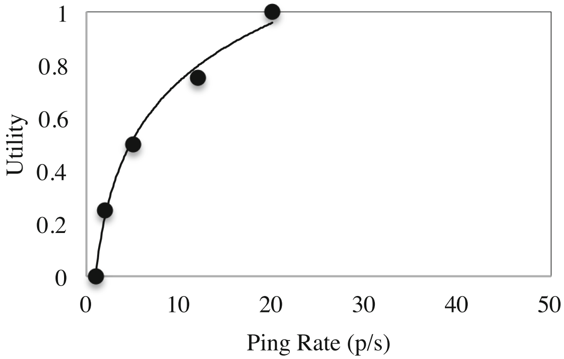


Fig. 8. Operator 3 ping rate utility as a function of ping rate (ping/s).

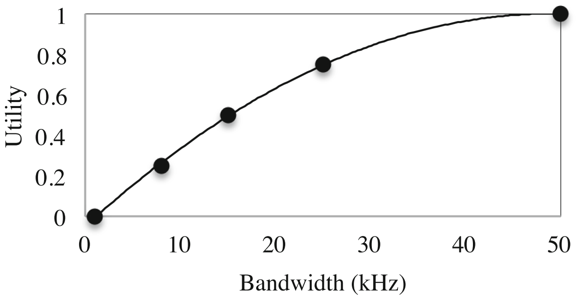


Fig. 9. Operator 3 bandwidth utility as a function of bandwidth (B).

And:

$$u_F = e^{\frac{-(F-410)^2}{60,000}} \tag{9}$$

$$u_P = \begin{cases} -0.0004B^2 + 0.05B + 0.04 & \text{for } B \le 20 \\ 1 & \text{for } B > 20 \end{cases} \tag{10}$$

$$u_B = 0.31 \ln P + 0.003 \tag{11}$$

Once attribute-specific utility functions are established, additional interviews of the operators 1, 2, and 3, reveal the weights for each attribute utility to be shown in Eqs. 12, 13, and 14, respectively:

$$u_{Operator\,1} = 0.73u_F + 0.13u_P + 0.14u_B \tag{12}$$

$$u_{Operator\,2} = 0.85u_F + 0.07u_P + 0.08u_B \tag{13}$$

$$u_{Operator\,3} = 0.71u_F + 0.15u_P + 0.14u_B \tag{14}$$

Of note, operators 1 and 3 have similar project management backgrounds and produced similar utility functions, both in weight and form. Operator 2, however, has a stronger technical background and placed more emphasis than the other operators on the technical impact of frequency on resolution.

To test the validity of these equations for these operators, the theoretical sonars depicted in Table 1 were passed through Eqs. 12, 13, and 14.

Table 1. Theoretical sonars

Sonar	Frequency (kHz)	Ping rate (ping/s)	Bandwidth (kHz)
1	300	5	50
2	410	10	50
3	500	20	10

A ranking of most preferred to least preferred of these theoretical sonars was calculated for each operator using the above equations. Concurrently, these theoretical sonars were presented to the operators, who then ranked these sonars. The equation rankings and operator rankings from most preferred to least preferred are listed in Tables 2, 3, and 4, with calculated utility listed in parentheses next to each equation ranking:

Table 2. Operator 1 rankings of theoretical sonars.

	First preference	Second preference	Third preference
Equation ranking	Sonar 2 (0.75)	Sonar 1 (0.94)	Sonar 3 (0.65)
Operator ranking	Sonar 2	Sonar 1	Sonar 3

Table 3. Operator 2 rankings of theoretical sonars.

	First preference	Second preference	Third preference
Equation ranking	Sonar 1 (0.85)	Sonar 2 (0.33)	Sonar 3 (0.09)
Operator ranking	Sonar 1	Sonar 2	Sonar 3

Table 4. Operator 3 rankings of theoretical sonars.

	First preference	Second preference	Third preference
Equation ranking	Sonar 2 (0.87)	Sonar 1 (0.96)	Sonar 3 (0.81)
Operator ranking	Sonar 2	Sonar 3	Sonar 1

The equations for Operators 1 and 2 produced test results that matched the operator rankings on the first iteration. The equation for Operator 3, however, did not initially provide results that matched operator ranking. After presenting Operator 3 with the calculated utilities for the three sonars—specifically, how similarly-ranked sonars 2 and 3 were, Operator 3 conceded that he equally or nearly-equally preferred the equation ranking and his original ranking and agreed to the equation rankings. Because these two sets of rankings—from the equations and from the operators—eventually matched, these equations are reasonable bases on which to build more complete utility functions of side scan sonars for this mission.

6 Discussion, Conclusion, and Future Work

By applying utility theory to the decision making processes of AUVs in mission planning and replanning around failed payloads, AUVs can more robustly make decisions without real-time assistance from human operators. Instead, operator preference, as defined prior to mission performance, will influence these decisions. By taking into account multiple applicable attributes and multiple applicable types of missions, the vehicles can be programmed to handle a variety of alternate payload decisions, thus improving the likelihood that the mission goal is met. As the examples presented in this paper demonstrate, different operators will have different preferences. As such, the equations must be tailored to specific operators.

To transition this work from the theoretical to the practical, future efforts should include reconfiguring the logic of a vehicle controller to accept mission goals, rather than specific mission payloads and associated activities, as well as allowing the vehicle to calculate the utility of each payload for meeting mission goals, as defined by the equations created by the process outlined in this paper. Such modifications would allow the vehicle to initially plan the mission around the 'best' payload available and replan the mission around the next 'best' payload, in the event the initially-selected payload fails.

References

1. Seto, M.L. (ed.): Marine Robot Autonomy. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-5659-9
2. Patron, P., Miguelanez, E., Petillot, Y.R., Lane, D.M., Salvi, J.: Adaptive mission plan diagnosis and repair for fault recovery in autonomous underwater vehicles (2008). http://osl.eps.hw.ac.uk/files/uploads/publications/planning.pdf
3. Patron, P.: Semantic-based adaptive mission planning for unmanned underwater vehicles. Doctoral thesis (2010). http://www.ros.hw.ac.uk/handle/10399/2373
4. Naval Research Advisory Committee: How autonomy can transform naval operations (2012). http://www.nrac.navy.mil/docs/NRAC_Final_Report-Autonomy_NOV2012.pdf
5. Keeney, R., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Cambridge University Press, Cambridge (1976)
6. Kessel, R., Pastore, T.: A comparative analysis of side-looking sonars for rapid classification of underwater intruders (NURC-PR-2008-010). In: Proceedings of the 1st International Conference and Exhibition on Waterside Security (WSS 2008), 25–28 August 2008, Technical University of Denmark, Copenhagen, Denmark (2008)
7. Pinto, M.: High resolution sonar (tutorial slides). In: OCEANS 2010 IEEE Sidney, Australia, 24–27 May 2010 (2010)

Security

Challenges of Misbehavior Detection in Industrial Wireless Networks

Sebastian Henningsen(✉), Stefan Dietzel, and Björn Scheuermann

Humboldt-Universität zu Berlin, Berlin, Germany
{sebastian.henningsen,stefan.dietzel}@hu-berlin.de,
scheuermann@informatik.hu-berlin.de

Abstract. In recent years, wireless technologies are increasingly adopted in many application domains that were either unconnected before or exclusively used cable networks. This paradigm shift towards – often ad-hoc – wireless communication has led to significant benefits in terms of flexibility and mobility. Alongside with these benefits, however, arise new attack vectors, which cannot be mitigated by traditional security measures. Hence, mechanisms that are orthogonal to cryptographic security techniques are necessary in order to detect adversaries. In traditional networks, such mechanisms are subsumed under the term “intrusion detection system,” and many proposals have been implemented for different application domains. More recently, the term “misbehavior detection” has been coined to encompass detection mechanisms especially for attacks in wireless networks. In this paper, we use industrial wireless networks as an exemplary application domain to discuss new directions and future challenges in detecting insider attacks. To that end, we review existing work on intrusion detection in mobile ad-hoc networks. We focus on physical-layer-based detection mechanisms as these are a particularly interesting research direction that had not been reasonable before widespread use of wireless technology.

Keywords: Physical-layer security · Wireless security
Industrial wireless networks · Intrusion detection

1 Introduction

Cyber-physical systems, such as, power plants, intelligent transportation systems, connected cars, or industrial automation systems, were traditionally mostly autonomous. As their connectivity increases, a constant threat arises by both insider and outsider adversaries with varying motivations. One source of motivation is industrial espionage, where a competitor tries to obtain secret intellectual property in order to void a technological advantage. Also, acts of sabotage in the name of national security have been observed. Famously, the Stuxnet virus [1] presumably targeted nuclear facilities in Iran. Attacks on cyber-physical systems may lead to severe disadvantages for companies, and they may even endanger lives.

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 37–46, 2018.
https://doi.org/10.1007/978-3-319-74439-1_4

Importantly, most attacks on industrial communication are mounted from within the network. As recent attack examples demonstrate, adversaries often infiltrate their target systems and then act as insiders, using compromised systems within the network's trusted perimeter. Therefore, additional layers of defense are necessary when cryptographic protection, like encryption and signatures, are compromised. Mechanisms are required that inspect messages and message sequences in order to ensure their plausibility and consistency. These mechanisms are usually subsumed under the term Intrusion Detection System (IDS). In industrial networks, such systems are important to detect active insider attacks. As existing general-purpose IDS cannot simply be converted for use in industrial settings, researchers have made tailored proposals that include application semantics (e.g., [2–4]). These proposals typically apply to wired industrial communication architectures, such as the supervisory control and data acquisition (SCADA) system.

Whereas these example attacks and tailored IDS target traditional IT infrastructure that is mostly wired, recent advances in wireless technology have led to a pervasive adoption of wireless systems – including their use for industrial communication. Depending on the use case, the industrial communication networks' structure differs widely between closed-loop systems and ad-hoc topologies. Common to all applications are tight resource constraints, for example, on the latency or expected throughput.

Fundamentally, the shift towards wireless connectivity opens up new attack vectors due to the wireless medium's broadcast nature. Eavesdropping and jamming of packets, for instance, are simplified significantly. Moreover, with increased flexibility, industrial communication systems become more complex and require more maintenance. This additional maintenance extends the risk of inside adversaries, such as disguised IT-maintenance personnel. To counteract these novel attack forms, the term "misbehavior detection" emerged, which can be seen as a subform of IDS for wireless systems [5–7]. Example mechanisms for misbehavior detection are two-step ACKs [8], monitoring watchdogs [9] and automatic feature correlation [10]. These approaches, however, may not always be suited for particular situations. For example, due to resource constraints, misbehavior detection in industrial wireless communication systems is challenging and closely related to existing approaches for mobile ad-hoc networks (MANETs). Instead, lightweight misbehavior detection based on physical-layer characteristics can be used. These physical-layer characteristics depend on the sender's location and are, in contrast to the communication's content, not spoofable by an adversary. Misbehavior detection mechanisms can leverage these channel characteristics for adversary detection – a practice located in the field of physical-layer security. In particular, lightweight security based on physical-layer properties is well-suited for this task.

Therefore, in this paper, we discuss the potential and challenges of misbehavior detection mechanisms in industrial wireless communication. We elaborate on these challenges (Sect. 2) before turning to existing approaches for attack detection algorithms in MANETs in Sect. 3. In particular, we state their assumptions and limitations and discuss whether ideas can be transferred industrial wireless use cases. We identify the challenge of trusted information dissemination in multi-hop networks with potential adversaries, which we investigate in Sect. 4 alongside with suggestions for possible solutions. We summarize pointers for future research and conclude the paper in Sect. 5.

2 Challenges of Industrial Misbehavior Detection

Due to new attack vectors, the development of misbehavior detection frameworks for emerging industrial networks is important to add an additional layer of security to these systems. Traditional wired industrial automation systems and emerging – more general and flexible – wireless systems are fundamentally different. Therefore, existing IDS techniques cannot easily be transferred to the new use case, necessitating novel mechanisms.

Challenges arise due to the variety of different application scenarios and industrial wireless networks' contradictory requirements. On the one hand, use cases in industrial wireless networks range from closed-loop systems with low latency and low packet error rate, to multi-hop information propagation from sensors to data sinks. These applications differ not only in their respective scenario but also in requirements and communication characteristics. In closed-loop systems, for example, communication is synchronized and optimized for maximum throughput and minimal packet error rate, while not providing flexibility in terms of joining/parting nodes. Multi-hop information propagation, as the other extreme, is flexible in the topology and can easily tolerate packet errors but cannot achieve minimal delay. Hence, the challenge when developing misbehavior detection techniques is to not only cope with these differences but rather leverage these characteristics. Domain-specific properties should be used to develop tailored mechanisms for each use case in order to achieve maximal detection performance and security. On the other hand, contradictory requirements regarding the properties of industrial wireless networks lead to additional challenges in the design of misbehavior detection systems. Closed-loop, low-latency wireless networks, for example, aim at achieving wire-equivalent performance while providing more security than traditional wire-based systems. Hence, the security mechanisms must be as lightweight as possible but still maintain a high security standard – a tradeoff that has to be accounted for in the development of suitable misbehavior detection approaches. Thus, the challenge when designing these approaches lies in this tradeoff and finding the equilibrium between performance and security.

Therefore, lightweight misbehavior detection systems are necessary, which inflict as little resource usage as possible while still maintaining a sufficient level of security. Particularly promising approaches for this task are passive physical-layer security mechanisms, which monitor the communication and only send messages in the presence of suspicious activities. The resource-friendly availability of these channel measurements makes them well suited for resource-constrained networks, such as, industrial automation systems or MANETs. Passive physical-layer security has been employed for misbehavior detection in MANETs, especially for Sybil attack detection [11–13] and classical intrusion detection [14,15]. Two questions arise: can these techniques be transferred from MANETs to industrial communication systems? And, what are the potential limitations and future research directions?

3 New Challenges – Old Solutions?

In the following, we discuss existing physical-layer approaches for MANETs and their applicability to industrial wireless networks. Physical-layer security leverages physical properties to enhance security, which involves diverse topics such as distance bounding [16], key derivation [17], node authentication [18], and intrusion detection [19]. We distinguish between active and passive mechanisms: active physical-layer security algorithms require additional actions, such as, a challenge-response communication in distance bounding. Passive mechanisms, in contrast, simply monitor the desired physical properties of the communication channel between the monitoring node and the sender. The channel between sender and receiver determines how wireless signals are altered by the environment through, for instance, reflections and diffractions. Due to these environmental effects, each location is unique in terms of how the signal arrives at the receiver [20]. Moreover, the pairwise channel between sender and receiver cannot be estimated through eavesdropping at different locations and is, therefore, considered a shared secret [17]. The channel properties are estimated by the receiver at the beginning of each transmission sequence; hence, these measurements are readily available without inducing additional network load – a significant advantage over traditional cryptographic methods.

The applicability of some existing works is hindered by strong assumptions on the attacker or the network in general. Newsome *et al.* [12], for example, propose a radio resources testing scheme based on dividing the frequency band into subchannels for each neighbor. Under the assumption that the attacker radio can only listen and send on a single frequency, Sybil nodes will be detected. Besides the limitation in network bandwidth, an attacker may use multiple antennas to circumvent this mechanism. Similarly, in [21,22], a low-mobility network is assumed, which is not generally the case for industrial communication systems.

In their seminal work, Demirbas *et al.* [23] propose a detection mechanism based on Received Signal Strength Indicator (RSSI) measurements. Their work is an implementation of [24], where a framework of four collaborating honest nodes is used to locate a node. It is proven that no node can hide its position when monitored by four honest nodes. The authors point out that exactly locating a node is not necessary in order to detect Sybil attacks; instead, it suffices to process the measured RSSI values directly. The approach is applicable in general scenarios, though in both works, the problem of information dissemination and selection of honest nodes is not tackled.

Sybil detection using RSSI measurements has subsequently been adopted in a number of works [21,22,25,26]. These approaches, however, are based on strong assumptions regarding the mobility in the network or the attacker's capabilities. In [22], for example, it is assumed that the attacker cannot control transmission power, whereas Wang *et al.* [25] assume no mobility in the network. Although Wang *et al.*'s assumptions do not hold in industrial communication systems, the proposed hierarchical approach to Sybil detection is insightful and can be easily transferred to other application domains. Information is shared by flooding, which is robust but suffers from low performance.

An analytical justification of Sybil detection via channel measurements is given by Xiao *et al.* [20], who theoretically analyze the Channel Impulse Response (CIR). It is shown that the CIR quickly de-correlates with distance. Based on this theoretical analysis, a hypothesis test – Sybil attacker present or not – is proposed. The necessary parameters for this hypothesis test are derived from the theoretical model. Chen *et al.* [27,28] extend this theoretical treatment to Received Signal Strength. The conducted analysis provides valuable insights not only for MANETs but also for other application domains. The attacker detection in [18,20,29,30] is based on hypothesis testing of only the last measurement, which may not be enough if large channel fluctuations occur. Subsequent works are based on a k-means cluster analysis by maintaining a sliding window of past measurements, which is more suitable for the industrial use case. Again, when measurements from multiple nodes are combined, the focus lies on the detection algorithm rather than information dissemination.

The field of secure localization is also closely related to the detection of Sybil attacks, since an attacker cannot forge multiple fake identities if each network node can be localized. Localization techniques that were not designed with an adversarial setting in mind, however, suffer shortcomings and potential security weaknesses in the presence of an attacker [31,32]. Hence, secure localization approaches can either cope with attacker-injected outliers [33] or use unforgeable physical properties to make attacks impossible [32]. Thwarting attacks by making algorithms more robust is a promising idea but a difficult task and requires careful consideration of the use case and attacker model at hand, thus providing interesting research challenges. In general, since secure localization schemes are oftentimes based on active measurements [32,34], they are not particularly well suited for enhancing the security in industrial wireless networks.

Passive physical-layer techniques based on channel properties are not only employed for Sybil detection, but also for misbehavior detection in general. In fact, in an IDS, the detection of Sybil attacks and impersonation attacks is one of many monitoring tasks. In these wireless intrusion detection systems, passive physical-layer security techniques are used as one possible source of information [14,15]. A key observation of wireless intrusion detection systems is that a single metric is not enough for adequate attacker detection. When detecting jammers, for example, Xu *et al.* have shown that with a single metric, such as RSSI, not all potential jamming strategies can be detected [35,36]. Instead, the authors propose a combination of different indicators, such as RSSI, packet delivery ratio, and node location, to detect and mitigate jamming attacks. The combination significantly improves the detection performance, an observation that applies to other use cases, as well [9,32,37]. Hence, relying on a single metric for intrusion detection is not enough, instead one has to take into account all available information.

4 New Solutions Are Necessary

As we have seen, a number of channel-based physical-layer security approaches for MANETs exist. Most of these ideas require additional research to transform existing knowledge into misbehavior detection techniques in industrial wireless networks. A promising general idea is to leverage the wireless medium's broadcast nature by aggregating data from multiple nodes for improved detection accuracy. Common to all algorithms that rely on multiple cooperating nodes or distributed measurements is the challenge of information dissemination. Especially in an adversarial setting, trusted communication among the monitoring nodes is a vital aspect of these algorithms.

Depending on the attacker model, this problem may be easily solvable with traditional cryptographic mechanisms. If we, however, assume an inside attacker with access to key material, as motivated in Sect. 1, authenticity and integrity cannot be guaranteed anymore. Attackers could easily jam the communication of an honest node [38] and inject their own data instead. These challenges have been investigated separately in great detail and suitable solutions have been proposed for the individual aspects: cryptographic mechanisms at least ensure authenticity as well as integrity, jamming can be detected in many cases [35], and cloned/overtaken nodes can be detected [39,40]. However, individual solutions assume different attacker models and thus can not necessarily be combined.

In the end, given an attacker with the ability to inject/jam packets, distributed information sharing is closely related to the notion of trust. Trust can be node-centric or data-centric, i.e., to which degree the bearer of information is trusted and how plausible the data is. Especially in multi-hop networks in the presence of an attacker, the trust in a certain data item quickly decreases in the number of hops between sender and receiver. In particular, the trust in received information depends on both, the data itself, as well as the nodes traversed on the routing path. When designing distributed algorithms, these complex trust relationships should therefore explicitly be taken into account in order to make the system more resilient against attacks.

While the notion of trust is intuitive to most, several questions arise in the context of communication networks: How to quantify and compute trust? How to include trust into, e.g., routing and attacker detection decisions? How to deal with inconsistent "trust views" or opinions between the nodes? Most Sybil attack detection algorithms are based on statistical methods, such as hypothesis testing [18,20,29,30] or machine learning algorithms [27]. Hence, their output is not binary but probabilistic. Although this output could serve as the basis for trust derivation, it is not trivial to cope with statistical outliers nor to normalize the resulting values. Optimally, the trust should depend on the situation rather than the underlying algorithm; thus, some sort of output normalization is necessary. Additionally, the data should contribute to the trust computation as well. Hence, every node must have some model, theoretical or empirical, for each neighbor, specifying the expected range of data. Since the nodes are often resource constrained, one has to face a tradeoff between accuracy and cost of these models. Moreover, the transitive computation of trust over multiple hops is a difficult challenge. If a node does not trust a direct neighbor, any information transferred via that neighbor is questionable. One is tempted to simply dismiss these transitive relationships and focus on the direct neighbors instead, which would, however, give an attacker an easy leverage to disturb the network. Last but not least, computing trust for more than just immediate neighbors is a difficult task: one has to design suitable metrics and is still facing the problem of obtaining reliable information on multi-hop neighbors.

Including computed trust values into routing and attacker detection decisions poses another challenge. When incorporating data from multiple sources in distributed algorithms, the trust in each particular source should influence the decision. In attack detection, for example, one could weigh each received (and missing) data item according to the computed trust associated with that packet. However, again, an attacker could exploit that mechanism to alter the decisions of the detection algorithm by decreasing the trust in honest nodes. A possible remedy is the development of robust algorithms [33], which are able to deal with or even detect intentional outliers. When every node maintains its own trust view on the network, inconsistencies can arise, which could cause difficulties when nodes are to agree on a consensus in a distributed fashion. Naturally, the trust has to be taken into account in this process. In situations without trust, this is normally achieved by Byzantine commitment protocols [41]. The combination of trust values for each packet, together with traditional Byzantine commitment protocols and the establishment of fundamental bounds on this procedure, are also a challenging research question.

A potential method to formally capture trust relationships in general is subjective logic [42]. Subjective logic is a powerful and flexible framework for reasoning under uncertainty, in which facts or measurements are extended to opinions in order to incorporate said uncertainty. It provides notions for data-centric trust and node-centric trust, as well as operators to combine opinions, which can be used to model trust relationships. In particular, subjective logic provides several operators in order to reduce trust chains to a single logical opinion. The choice of operators depends on the setting at hand and requires careful consideration.

5 Concluding Outlook on Future Research

In this paper we discussed the challenges of misbehavior detection in industrial wireless networks. These challenges arise in multiple aspects, from jamming detection and prevention to malicious packet injection; hence, a holistic perspective is required to design suitable solutions. Although it is reasonable to focus on a particular aspect in research, a heterogeneous landscape of different assumptions and models makes it difficult to combine individual ideas. We believe that a harmonization of models is necessary so that the design of a complex systems from individual parts is facilitated.

As we have seen in Sect. 3, physical-layer information, which is readily available without any additional cost, has been successfully leveraged for attacker detection. Due to firmware constraints in commercial off-the-shelf hardware, often only coarse-grained information, such as RSSI, is reported to higher layers. However, more fine-grained CIR information is available, which would significantly increase the performance of detection algorithms [43]. Thus, in our view, the development of suitable firmwares capable of providing more information is an important step towards future solutions.

Last but not least, we discussed information dissemination in adversarial multi-hop environments. We propose to include the notion of trust into algorithms and protocols in order to dependably spread information. Possible directions for future research include (1) the computation of the trust itself, especially over multiple hops, (2) the incorporation of trust into existing protocols, and (3) trust-based distributed consensus.

Acknowledgments. The work was partly funded by the German Federal Ministry of Education and Research under BMBF grant agreement no. 16KIS0222.

References

1. Langner, R.: Stuxnet: dissecting a cyberwarfare weapon. IEEE Secur. Priv. **9**(3), 49–51 (2011)
2. Hadžiosmanović, D., Sommer, R., Zambon, E., et al.: Through the eye of the PLC: semantic security monitoring for industrial processes. In: Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014. ACM (2014)
3. Hadziosmanovic, D., Bolzoni, D., Etalle, S., et al.: Challenges and opportunities in securing industrial control systems. In: 2012 Complexity in Engineering (COMPENG), Proceedings, June 2012
4. Kargl, F., van der Heijden, R.W., König, H., et al.: Insights on the security and dependability of industrial control systems. IEEE Secur. Priv. **12**(6), 75–78 (2014)
5. Radosavac, S., Baras, J.S., Koutsopoulos, I.: A framework for MAC protocol misbehavior detection in wireless networks. In: Proceedings of the 4th ACM Workshop on Wireless Security. ACM (2005)
6. Sarafijanovic, S., Le Boudec, J.-Y.: An artificial immune system approach with secondary response for misbehavior detection in mobile ad hoc networks. IEEE Trans. Neural Netw. **16**(5), 1076–1087 (2005)
7. Butun, I., Morgera, S.D., Sankar, R.: A survey of intrusion detection systems in wireless sensor networks. IEEE Commun. Surv. Tutorials **16**(1), 266–282 (2014)
8. Liu, K., Deng, J., Varshney, P.K., et al.: An acknowledgment-based approach for the detection of routing misbehavior in MANETs. IEEE Trans. Mobile Comput. **6**(5), 536–550 (2007)
9. Marti, S., Giuli, T.J., Lai, K., et al.: Mitigating routing misbehavior in mobile ad hoc networks. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom 2000. ACM (2000)

10. Huang, Y.-A., Lee, W.: A cooperative intrusion detection system for ad hoc networks. In: Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks. ACM (2003)
11. Yang, H., Luo, H., Ye, F., et al.: Security in mobile ad hoc networks: challenges and solutions. IEEE Wirel. Commun. **11**(1), 38–47 (2004)
12. Newsome, J., Shi, E., Song, D., et al.: The Sybil attack in sensor networks: analysis & defenses. In: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, IPSN 2004. ACM (2004)
13. Sanzgiri, K., Dahill, B., Levine, B.N., et al.: A secure routing protocol for ad hoc networks. In: 10th IEEE International Conference on Network Protocols, Proceedings. IEEE (2002)
14. Onat, I., Miri, A.: An intrusion detection system for wireless sensor networks. In: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, vol. 3. IEEE (2005)
15. Bhuse, V., Gupta, A.: Anomaly intrusion detection in wireless sensor networks. J. High Speed Netw. **15**(1), 33–51 (2006)
16. Brands, S., Chaum, D.: Distance-bounding protocols. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 344–359. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48285-7_30
17. Jana, S., Premnath, S.N., Clark, M., et al.: On the effectiveness of secret key extraction from wireless signal strength in real environments. ACM, MobiCom (2009)
18. Xiao, L., Reznik, A., Trappe, W., et al.: PHY-authentication protocol for spoofing detection in wireless networks. In: Global Telecommunications Conference (GLOBECOM 2010). IEEE (2010)
19. Faria, D.B., Cheriton, D.R.: Detecting identity-based attacks in wireless networks using signalprints. In: Proceedings of the 5th ACM Workshop on Wireless Security. ACM (2006)
20. Xiao, L., Greenstein, L.J., Mandayam, N.B., et al.: Using the physical layer for wireless authentication in time-variant channels. IEEE Trans. Wirel. Commun. **7**(7), 2571–2579 (2008)
21. Abbas, S., Merabti, M., Llewellyn-Jones, D.: Signal strength based Sybil attack detection in wireless ad hoc networks. In: 2009 Second International Conference on Developments in eSystems Engineering. IEEE (2009)
22. Abbas, S., Merabti, M., Llewellyn-Jones, D., et al.: Lightweight Sybil attack detection in MANETs. IEEE Syst. J. **7**(2), 236–248 (2013)
23. Demirbas, M., Song, Y.: An RSSI-based scheme for Sybil attack detection in wireless sensor networks. In: WOWMOM. IEEE (2006)
24. Zhong, S., Li, L., Liu, Y.G., et al.: Privacy-preserving location-based services for mobile users in wireless networks. Department of Computer Science, Yale University, Technical report ALEU/DCS/TR-1297 (2004)
25. Wang, J., Yang, G., Sun, Y., et al.: Sybil attack detection based on RSSI for wireless sensor network. In: International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2007. IEEE (2007)
26. Jan, M.A., Nanda, P., He, X., et al.: A Sybil attack detection scheme for a centralized clustering-based hierarchical network. In: Trustcom/BigDataSE/ISPA, vol. 1. IEEE (2015)
27. Chen, Y., Yang, J., Trappe, W., et al.: Detecting and localizing identity-based attacks in wireless and sensor networks. IEEE Trans. Veh. Technol. **59**(5), 2418–2434 (2010)

28. Chen, Y., Xu, W., Trappe, W., Zhang, Y.: Detecting and localizing wireless spoofing attacks. In: Securing Emerging Wireless Systems. Springer, Boston (2009)
29. Xiao, L., Greenstein, L., Mandayam, N., et al.: A physical-layer technique to enhance authentication for mobile terminals. In: Proceedings of ICC. IEEE (2008)
30. Xiao, L., Greenstein, L.J., Mandayam, N.B., et al.: Channel-based detection of Sybil attacks in wireless networks. IEEE Trans. Inf. Forensics Secur. **4**(3), 492–503 (2009)
31. Du, X., Chen, H.-H.: Security in wireless sensor networks. IEEE Wirel. Commun. **15**(4), 60–66 (2008)
32. Sastry, N., Shankar, U., Wagner, D.: Secure verification of location claims. In: Proceedings of the 2nd ACM Workshop on Wireless Security. ACM (2003)
33. Li, Z., Trappe, W., Zhang, Y., et al.: Robust statistical methods for securing wireless localization in sensor networks. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks. IEEE (2005)
34. Capkun, S., Hubaux, J.-P.: Secure positioning in wireless networks. IEEE J. Sel. Areas Commun. **24**(2), 221–232 (2006)
35. Xu, W., Trappe, W., Zhang, Y., et al.: The feasibility of launching and detecting jamming attacks in wireless networks. ACM, MobiHoc (2005)
36. Xu, W., Ma, K., Trappe, W., et al.: Jamming sensor networks: attack and defense strategies. Netw. Mag. Global Internetwkg. **20**(3), 41–47 (2006)
37. Glynos, D., Kotzanikolaou, P., Douligeris, C.: Preventing impersonation attacks in MANET with multi-factor authentication. In: Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks. IEEE (2005)
38. Wilhelm, M., Martinovic, I., Schmitt, J.B., et al.: Short paper: reactive jamming in wireless networks: how realistic is the threat? In: Proceedings of the Fourth ACM Conference on Wireless Network Security. In: WiSec 2011. ACM (2011)
39. Conti, M., Di Pietro, R., Mancini, L., et al.: Distributed detection of clone attacks in wireless sensor networks. IEEE Trans. Dependable Secure Comput. **8**(5), 685–698 (2011)
40. Parno, B., Perrig, A., Gligor, V.: Distributed detection of node replication attacks in sensor networks. In: 2005 IEEE Symposium on Security and Privacy. IEEE (2005)
41. Lamport, L., Shostak, R., Pease, M.: The Byzantine generals problem. Trans. Prog. Lang. Syst. (TOPLAS) **4**(3), 328–401 (1982)
42. Jøsang, A.: Subjective Logic - A Formalism for Reasoning Under Uncertainty. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-42337-1
43. Yang, Z., Zhou, Z., Liu, Y.: From RSSI to CSI: indoor localization via channel response. ACM CSUR **46**(2), 25 (2013)

A New Look at an Old Attack: ARP Spoofing to Create Routing Loops in Ad Hoc Networks

J. David Brown(✉) and Tricia J. Willink

Defence R&D Canada, Ottawa, Canada
{david.brown,tricia.willink}@drdc-rddc.gc.ca

Abstract. This paper examines a new application of the well-known ARP spoofing (or ARP cache poisoning) attack. Traditionally, ARP spoofing has been applied in local area networks to allow an attacker to achieve a man-in-the-middle position against target hosts, or to implement a denial of service by routing messages to non-existent hardware addresses. In this paper, we introduce a variant of ARP spoofing in which a routing loop is created in a target wireless ad hoc network. The routing loop not only results in a denial of service against the targeted hosts, but creates a resource consumption attack, where the targets waste power and occupy the channel, precluding its use by legitimate traffic. We show experimental results of an implementation and provide suggestions as to how to prevent, detect, or mitigate the attack.

Keywords: Denial of service · ARP spoofing · Ad hoc networks
Sensor networks · Routing loops · Resource consumption · DoS defences

1 Introduction

ARP cache poisoning, or ARP spoofing, is a well-known network attack technique against local area networks (LANs) in which an attacker sends spoofed Address Resolution Protocol (ARP) messages to one or more target hosts. ARP spoofing can be performed as the first step in a larger attack, where the end goal of the attacker could be to achieve a man-in-the-middle position between two hosts or to cause a denial of service (DoS) against one or more hosts. The Address Resolution Protocol is vulnerable to spoofing because ARP messages include no authentication (see RFC 826, and updates in RFC 5227, 5494 [1–3]) and thus any host connected to the target network can emit an ARP request or response purporting to come from another host. This technique has been recognized for nearly 20 years, and it remains an area of interest as evidenced by continued activity in the security community, examining techniques to detect it and mitigate it—see for instance [4–7].

While ARP spoofing is usually discussed in the context of wired LANs, it is arguably more damaging—and easier to perform—in wireless ad hoc networks, where hosts are expected to leave and join frequently, the physical communication medium is easily accessible, and there is no central entity for security co-ordination; [14] enunciates the devastating effects of ARP poisoning on ad hoc networks. The "gold standard" of protection against ARP spoofing—namely, hard-coding the MAC and IP address pair

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 47–59, 2018.
https://doi.org/10.1007/978-3-319-74439-1_5

associations in each host—is impractical for many ad hoc network use cases, since it is often the case that the complete set of hosts that participate in an ad hoc network is not known *a priori*. Other existing defences against ARP spoofing rely on making modifications to existing protocols [15]; employing schemes or tools that perform passive monitoring of traffic or internal system parameters [8, 9]; or modifying operating system (OS) configurations. Although these defences are simple and practical, the reality is that they are often not implemented, leaving networks vulnerable [10]. As ad hoc networks become increasingly pervasive—in applications including sensor networks and for devices residing on the “Internet of Things”—it is likely that many of these networks will be developed and deployed without such defences in place since the driver for many industries will be in developing devices of low cost, low complexity, and interoperability. In fact, although it has been pointed out that ARP is not truly suited to ad hoc networks [11], the protocol will no doubt continue to be used in many implementations despite competing suggestions and algorithms.

In this paper, a novel use of the ARP spoofing technique is presented that can create a powerful DoS attack against a target ad hoc network. An attacker injects spoofed ARP packets into the ad hoc network such that a “routing loop” is formed between two or more hosts; as a result, an IP packet directed through any of the affected hosts oscillates “forever” in a loop—or until the packet’s time to live (TTL) expires. In this fashion, the attacker exerts relatively little effort (in terms of power resources) but creates a situation where the target network exhausts its own resources and floods the shared wireless channel. The attack is unique to ad hoc networks and does not port directly to the wired case, since in an ad hoc network all hosts on a common subnet can act as routers as well as endpoints, thus presenting the opportunity for creating routing loops among the hosts themselves. This is different from the standard set of ARP spoofing attacks, which generally force hosts to route through the attacker (creating a man-in-the-middle) or direct hosts to route to non-existing addresses (see [14]). While directing a host to a non-existing address results in a link failure (and a DoS against the host), the attack proposed here causes hosts to continue transmitting duplicate copies of packets—effectively depleting battery life and consuming channel resources, thus denying them to other non-targeted hosts.

The remainder of the paper is organized as follows. In Sect. 2, a brief review of the standard ARP spoofing attack is provided followed by a walk-through of a simple example of the new ARP-route-looping attack. Section 3 discusses how the ARP-route-looping attack could be applied to ad hoc networks in general, and identifies required topology pre-conditions that target networks must satisfy in order to allow for a successful effect. Section 4 provides the results of an experiment conducted on an ad hoc network comprised of Android smartphones, showing the effect of ARP-route-looping in a real-world scenario. Finally, Sect. 5 provides suggested defences and mitigations against the attack, with concluding remarks in Sect. 6.

2 The ARP-Route-Looping Attack

This section describes the ARP-route-looping attack by looking at a simple walk-through example. First, a brief description of traditional ARP spoofing is provided.

2.1 Traditional ARP Spoofing

Consider a simple IP network of two hosts, Alice and Bob, where Alice and Bob communicate over a wireless interface. When Alice sends a message to Bob, the message consists of a packet containing Bob's network IP address, denoted here as IP_B. As the packet travels down Alice's protocol stack, Alice's OS adds a hardware (or MAC) address for Bob, denoted here as MAC_B. Alice's OS obtains Bob's MAC from the local ARP cache, which contains a mapping of IPs to MACs for the hosts in the network. If the ARP cache does not contain an entry for Bob, Alice must broadcast an ARP request and wait for Bob's ARP reply (which contains MAC_B). In a traditional ARP spoofing attack, an attacker (Eve) sends spoofed ARP reply messages into the network to mislead Alice and Bob about the mappings of IPs to MACs.

In this scenario, Eve sends ARP spoofing messages to Alice indicating that Bob has hardware address MAC_E: Eve's MAC address. We use the notation Tx(E, A, <IP_B, MAC_E>) to denote that Eve (host h_E) sends a message to Alice (host h_A), where the message consists of an ARP spoof mapping IP_B to MAC_E. Likewise, Eve sends Tx(E, B, <IP_A, MAC_E>), indicating to Bob that Alice has hardware address MAC_E. Thus, Alice unwittingly sends traffic destined for Bob to MAC_E (and Bob sends traffic for Alice to MAC_E). Even in a wireless setting where Alice and Bob can hear all the traffic in the network, they will not process frames addressed to MAC_E (and will only process frames addressed to their own MAC addresses). Thus, once the poisoning is complete, Eve acts as a relay for all traffic between Alice and Bob and can modify, re-route, or drop packets as desired.

2.2 A Simple Example of ARP-Route-Looping

The ARP-route-looping attack is easily explained using a simple example. Consider an ad hoc network of four hosts (A, B, C and D), which we denote as h_A, h_B, h_C, and h_D. In this example, the ad hoc network is a complete graph, meaning that every host is within range of every other host. Thus in the absence any disruptions, all hosts can communicate with one another directly (i.e., without requiring multi-hop routes). The simple network is depicted in Fig. 1, where links are shown as light blue lines.

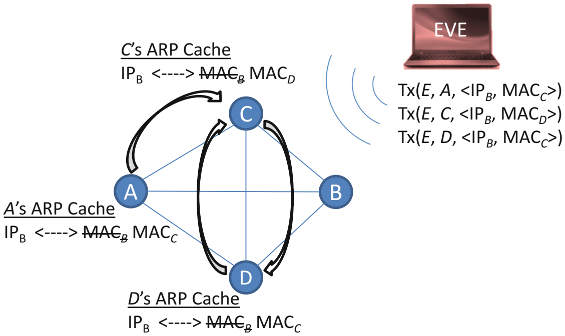

EVE
C's ARP Cache
IP$_B$ <----> ~~MAC$_B$~~ MAC$_D$
Tx(E, A, <IP$_B$, MAC$_C$>)
Tx(E, C, <IP$_B$, MAC$_D$>)
Tx(E, D, <IP$_B$, MAC$_C$>)
C
A
B
D
A's ARP Cache
IP$_B$ <----> ~~MAC$_B$~~ MAC$_C$
D's ARP Cache
IP$_B$ <----> ~~MAC$_B$~~ MAC$_C$

Fig. 1. An example ARP-route-looping attack. (Color figure online)

Suppose an attacker, Eve, wants to disrupt communication between h$_A$ and h$_B$ using ARP-route-looping. Initially, hosts h$_A$, h$_C$, and h$_D$ all have direct routes to h$_B$, along with ARP caches correctly mapping IP$_B$ to MAC$_B$. First, Eve sends Tx(E, A, <IP$_B$, MAC$_C$>) to h$_A$, poisoning the ARP cache of h$_A$ such that the MAC of h$_C$ is associated with the IP for h$_B$. Thus, whenever h$_A$ wants to send any unicast messages to h$_B$, h$_A$ will address the messages with the IP address of h$_B$, but the MAC address of h$_C$. Next, Eve sends Tx(E, C, <IP$_B$, MAC$_D$>), poisoning the ARP cache of h$_C$ such that all traffic from h$_C$ intended for h$_B$ will be sent to h$_D$. Finally, Eve sends Tx(E, D, <IP$_B$, MAC$_C$>), poisoning the ARP cache of h$_D$ such that all traffic from h$_D$ intended for h$_B$ will be sent to h$_C$.

Eve's activities are now finished and the conditions for ARP-route-looping have been set. Consider the following steps that inform the flow of a unicast packet that h$_A$ sends to h$_B$ in this scenario:

1. h$_A$ constructs a packet and inserts the IP address for h$_B$;
2. h$_A$ consults its routing table and determines that it has a direct route to h$_B$;
3. h$_A$ adds the MAC address for h$_B$ to the packet; h$_A$ consults its ARP cache to determine the MAC address for h$_B$;
4. h$_A$ inserts the (poisoned) entry MAC$_C$ and sends the packet;
5. h$_C$ receives the packet, examines the IP address and finds that the packet is destined for h$_B$; since the network is ad hoc, h$_C$ has an IP-forwarding capability and so forwards the packet to h$_B$ (the intended destination according to the IP address);
6. h$_C$ has a direct link to h$_B$ according to its routing table, so it must update the MAC address with the entry for h$_B$ before forwarding the packet; h$_C$ consults its ARP cache for the MAC of h$_B$ and inserts the (poisoned) entry MAC$_D$, then forwards the packet;
7. h$_D$ receives the packet, examines the IP address and finds that the packet is destined for h$_B$ and so forwards the packet to h$_B$;
8. h$_D$ consults its ARP cache to determine the MAC address for h$_B$ and inserts (poisoned) entry MAC$_C$.

At this point, the cycle repeats and returns to step 5, with h$_C$ once again receiving the packet. The packet will continue to cycle between h$_C$ and h$_D$ until the TTL counter for the packet reaches zero. This not only creates a DoS between hosts h$_A$ and h$_B$, but also creates a situation where h_C and h_D occupy the channel (precluding legitimate usage) and exhaust resources by transmitting duplicate packets in a loop.

When the TTL expires, h_D sends an ICMP time exceeded message to h_A (the source of the initial packet), which indicates that the TTL field in the IP header has reached zero. If desired, Eve can set conditions such that this ICMP message follows a routing loop as well, making it loop between hosts h_C and h_B until its own TTL reaches zero.

3 Generalized ARP-Route-Looping in Ad Hoc Networks

The ARP-route-looping attack was introduced in Sect. 2 for a specific example network. This section describes the technique in general terms and discusses how to determine which hosts to poison and what content to place in the spoofing messages.

3.1 Notation and Assumptions

We begin by introducing additional notation to describe the general ARP-route-looping attack. Consider a network of n hosts, where the network is represented as a graph. If an edge exists between any two hosts, h_i and h_j, they are “neighbours”. The neighbour-set of any host h_i is denoted by $\boldsymbol{N}(i)$, and consists of the set of all neighbours of h_i. Note that sets are denoted with bold italicized typeface, and hosts in a set are denoted by their indices for simplicity (e.g., we write $\{A\}$ instead of $\{h_A\}$). In the network from Fig. 1, for instance, $\boldsymbol{N}(C) = \{A, B, D\}$. We denote the relative complement of a set $\boldsymbol{N}(i)$ and a set of hosts $\boldsymbol{H}$ by $\boldsymbol{N}(\mathrm{i})\backslash\boldsymbol{H}$: this is the set of hosts in $\boldsymbol{N}(i)$ excluding the hosts in $\boldsymbol{H}$. So, for instance, in Fig. 1, $\boldsymbol{N}(C)\backslash\{A, B\} = \{D\}$.

We say there is a route or path between two hosts, h_i and h_j, if there exists a sequence of edges in the graph connecting h_i and h_j through some set of vertices. Assuming the network employs a shortest-path routing algorithm, we denote by $r(i, j)$ the first host (after h_i) along the route from h_i to h_j. In cases where there is more than one possible shortest path and $r(i,j)$ could have multiple values, for simplicity we select the host with the lowest index.

We remark that to achieve ARP-route-looping, the attacker must transmit a series of ARP spoofing messages to various hosts in the network. In a geographically diffuse ad hoc network it is possible that a single attacking node would be insufficient to reach all target hosts. For the purposes of this paper, however, we assume all spoofing packets arise from a single attacker, Eve, where it is understood that this may in fact consist of multiple co-ordinated transmitting stations. Furthermore, we assume that Eve has knowledge of the adjacency matrix of the graph representing the network—that is, Eve can compute which hosts are neighbours and can compute the shortest path routes between hosts in the network. Admittedly, in a dynamic network, this knowledge may be challenging to achieve; one possible strategy is for Eve to observe routing control messages and attempt to infer the adjacency matrix from these.

Finally, we note that any routing loop will terminate if the looping packet arrives at either the originator of the packet (since a host will not forward a packet for which it is identified as the source IP address) or the intended destination for the packet. Thus, in creating an ARP-route-loop both of these hosts must be avoided.

3.2 Two-Host Loops

Consider a network of n hosts, where Eve wants to implement ARP-route-looping against messages sent from h_A to h_B. The simplest loop is one that involves only two hosts (neither of which is h_A or h_B). An ARP-route-looping attack can be mounted in this case if the following condition holds:

$$\exists i \in \boldsymbol{N}(A)\backslash\{B\} \text{ such that } \boldsymbol{N}(i)\backslash\{A, B\} \neq \emptyset. \tag{1}$$

Condition (1) says that for two-node looping to be possible there must exist a host, h_i, that is a neighbour of h_A, but is not equal to the destination h_B; furthermore, h_i must have at least one neighbour that is equal to neither h_A nor h_B. When this condition is satisfied, we denote by h_j a node in the (non-empty) set $\boldsymbol{N}(i)\backslash\{A, B\}$. To create an ARP-route-loop, Eve can send out the following three ARP spoofing messages: Tx(E, A, <$\mathrm{IP}_{r(A,B)}$, MAC_i>), Tx(E, i, <$\mathrm{IP}_{r(i,B)}$, MAC_j>), Tx(E, j, <$\mathrm{IP}_{r(j,B)}$, MAC_i>). For hosts h_A, h_i, and h_j, these spoofing messages associate the IP for the next hop on the route to h_B with a poisoned MAC address. Eve's work is done and a routing loop is created between hosts h_i and h_j.

3.3 Loops with More Than Two Hosts

Although a loop with only two hosts is sufficient to perform ARP-route-looping, Eve may be interested in creating a situation where a message from h_A to h_B is looped among more than two hosts. This could have the effect of wasting the resources of more hosts in the network, and in the case of a geographically diffuse network an attacker may wish to occupy the channel over a larger geographic area by involving more hosts in the loop.

Ultimately, to achieve a multi-host loop attack involving k hosts ($k > 2$), Eve needs to identify a path in the graph originating at host h_A that contains a loop of k hosts, where host h_B is not part of the path. Finding loops, or cycles, in graphs is a well-studied problem—see, for instance, [12] and references therein. Although some optimizations to the problem exist in certain cases, for small graphs a brute force search is simple and not onerous to implement. A recursive brute force algorithm that Eve can use to find an appropriate cycle of length k is provided below in Algorithm 1, which is reminiscent of a depth-first search algorithm (e.g., [13]) with minor alterations. Note that *a priori*, Eve does not know whether the network contains a loop of length k and thus would run Algorithm 1 for all values k of interest.

Algorithm 1: *LoopSearch*$_k$*(i)*

```
push($\underline{\boldsymbol{S}}$, $i$)
**If** $\underline{\boldsymbol{S}}$ contains a loop of length $k$ **then**
    EXIT → sequence of hosts in $\underline{\boldsymbol{S}}$ is desired answer
**Else**
    $\boldsymbol{Current} = N(i) \setminus \{A, B, \underline{\boldsymbol{S}} \setminus k^{\text{th}}\text{-last})\}$
    **If** ($\boldsymbol{Current} = \emptyset$) or (LoopSearch$_k$ completed for all elements in $\boldsymbol{Current}$)
        pop($\underline{\boldsymbol{S}}$); return
    **Else**
        **For** each element $j$ in $\boldsymbol{Current}$
            LoopSearch_k($j$)
        **End For**
    **End If**
**End if**
**End**
```

Algorithm 1 includes the concept of a stack, which represents an ordered sequence of hosts—we denote the stack as a vector, $\underline{\boldsymbol{S}}$ (where we denote vectors using bold and underlining). The operator push($\underline{\boldsymbol{S}}$, i) means to add host h_i as the last element of vector $\underline{\boldsymbol{S}}$, and the operator pop($\underline{\boldsymbol{S}}$) means to remove the last element of vector $\underline{\boldsymbol{S}}$. When the stack, $\underline{\boldsymbol{S}}$, is used as part of a set, as in $\boldsymbol{C} = \{A, B, \underline{\boldsymbol{S}}\}$, the meaning is that all hosts in the stack are to be included in $\boldsymbol{C}$. When we write "$\underline{\boldsymbol{S}} \setminus k^{\text{th}}$-last", this refers to all hosts in $\underline{\boldsymbol{S}}$ except the k^{th}-last element. To find a loop of length k for a message sent by h_A to h_B, we start with an empty stack (i.e., $\underline{\boldsymbol{S}} = [\ ]$) and run LoopSearch$_k$($A$). Upon completion, either LoopSearch$_k$(A) will terminate with an empty stack (and the network contains no loop of length k) or LoopSearch$_k$(A) will terminate with a non-empty stack, where $\underline{\boldsymbol{S}}$ contains the sequence of hosts containing a length-k loop for traffic originating at h_A.

At a high level, Algorithm 1 works as follows. When LoopSearch is called for any host h_i, the host is pushed onto the stack. Then we check if the stack in its current form contains a loop; if so, we are done. If not, we recursively run LoopSearch again for all the neighbours of h_i, unless those neighbours are A, B, or other values currently in $\underline{\boldsymbol{S}}$ except for the k^{th}-last value in $\underline{\boldsymbol{S}}$ (because we do not want to find any loops in the network except loops of length k). Running LoopSearch in this fashion and excluding A, B, and $\underline{\boldsymbol{S}}$ is a generalization of condition (1) for the k-loop case. Running LoopSearch on h_A searches for a path containing a cycle, where the path originates at host h_A. As an example, running LoopSearch$_4$(A) on the graph in Fig. 2 provides the output: $\underline{\boldsymbol{S}} = [A, 4, 5, 6, 7, 9, 0, 6]$, identifying the 4-host loop among h_6, h_7, h_9, and h_0.

Once Eve has determined a sequence of hosts containing a loop of length k, ARP spoof packets can be crafted and sent. Consider that Eve has run Algorithm 1, which returned a vector of hosts, $\underline{\boldsymbol{S}}$, where $\underline{\boldsymbol{S}}$ contains m elements. In this case, Eve can craft ($m - 1$) ARP spoof messages as follows:

$$\mathrm{Tx}(E, \underline{\boldsymbol{S}}_i, < IP_{r(\underline{\boldsymbol{S}}_i, B)}, \mathrm{MAC}_{\underline{\boldsymbol{S}}(i+1)} >), \ \forall i \in [1, (m-1)], \tag{2}$$

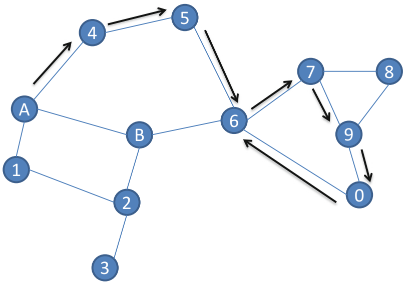


Fig. 2. Finding a loop of length $k = 4$ in a general graph with Algorithm 1.

where $\underline{\boldsymbol{S}}_i$ corresponds to the i^{th} host appearing in vector $\underline{\boldsymbol{S}}$. Thus, to the i^{th} host in $\underline{\boldsymbol{S}}$, Eve sends a spoofed message that maps the IP of the next host *en route* to h_B as corresponding to the MAC of the $(i + 1)^{\text{st}}$ host in $\underline{\boldsymbol{S}}$.

4 Experimental Results

To evaluate the effect of ARP-route-looping on an ad hoc network, we built a test network using Android smartphones (specifically, we used Nexus 5 model smartphones running the Cyanogenmod 13 Android-based operating system). The phones were configured to support ad hoc networking and multi-hop IP forwarding. We constructed the network shown in Fig. 3, where hosts h_A and h_B communicate via a multi-hop route through h_1 and h_2. The hosts in the network ran the optimum link state routing (OLSR) protocol to compute their neighbours and routes. To examine network traffic, we used a laptop running Wireshark in monitor mode as a packet sniffer.

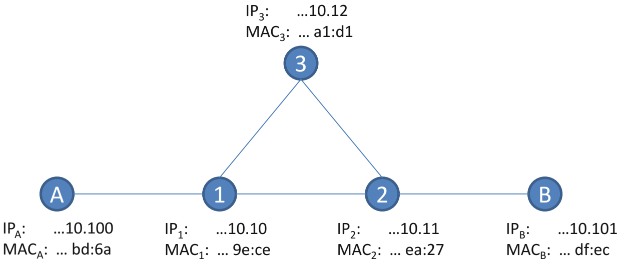
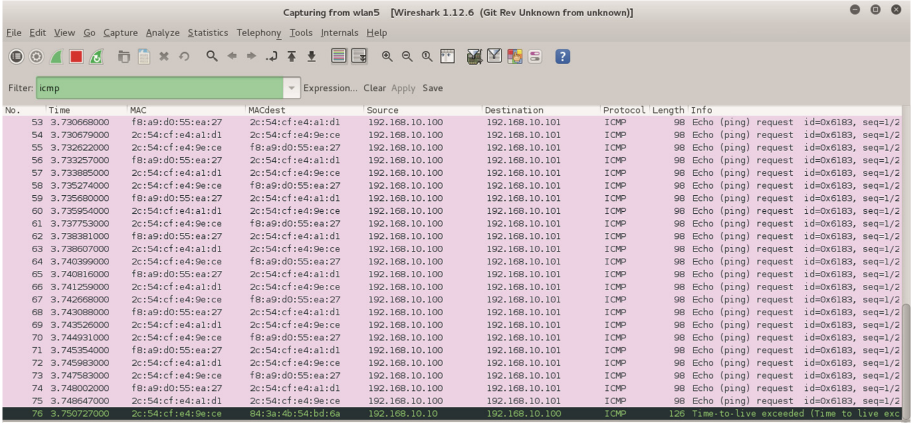


Fig. 3. Experimental network consisting of Android ad hoc hosts; the last two octets of the IPs and MACs are shown for each host.

In this scenario, we examined the ability of ARP-route-looping to disrupt communications from h_A to h_B; to target multiple hosts, we applied Algorithm 1 to find a 3-host loop, yielding $\underline{\boldsymbol{S}} = [A, 1, 2, 3, 1]$ (in this case, of course, we could have found the loop by inspection). To create ARP-route-looping in the network, we sent ARP spoofing messages as follows, based on Eq. (2): Tx(E, A, <IP$_1$, MAC$_1$>), Tx(E, 1, <IP$_2$, MAC$_2$>), Tx(E, 2, <IP$_B$, MAC$_3$>), Tx(E, 3 <IP$_2$, MAC$_1$>).

We expected the spoofing messages to create a routing loop such that all traffic sent from h_A to h_B would cycle around in a loop of hosts h_1, h_2, and h_3 until the TTL of the packet expires. Figure 4 shows a screen capture of our Wireshark packet sniffer when a single ICMP ping is sent from h_A to h_B in the presence of ARP-route-looping. In reading the figure, note that IP_A = 192.168.10.100 and IP_B = 192.168.10.101; for reference, MAC addresses of the hosts in the network are shown in Fig. 3. The single ping from h_A can be seen looping continuously among three hosts (note the MAC addresses in the Wireshark capture). Eventually—after 64 packets have been forwarded around the loop—the TTL of the packet expires and we see the ICMP TTL exceeded message returned to h_A. Not shown in Fig. 4 is the fact that we could have also simultaneously poisoned h_B, h_1, h_2, and h_3 creating a situation where the ICMP TTL exceeded message itself is looped 64 times among h_B, h_2, and h_3.


Capturing from wlan5 [Wireshark 1.12.6 (Git Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: icmp

```
No.  Time         MAC                MACdest            Source          Destination     Protocol Length Info
  53 3.730668000  f8:a9:d0:55:ea:27  2c:54:cf:e4:a1:d1  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  54 3.730679000  2c:54:cf:e4:a1:d1  2c:54:cf:e4:9e:ce  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  55 3.732622000  2c:54:cf:e4:9e:ce  f8:a9:d0:55:ea:27  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  56 3.733257000  f8:a9:d0:55:ea:27  2c:54:cf:e4:a1:d1  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  57 3.733885000  2c:54:cf:e4:a1:d1  2c:54:cf:e4:9e:ce  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  58 3.735274000  2c:54:cf:e4:9e:ce  f8:a9:d0:55:ea:27  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  59 3.735680000  f8:a9:d0:55:ea:27  2c:54:cf:e4:a1:d1  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  60 3.735954000  2c:54:cf:e4:a1:d1  2c:54:cf:e4:9e:ce  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  61 3.737753000  2c:54:cf:e4:9e:ce  f8:a9:d0:55:ea:27  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  62 3.738381000  f8:a9:d0:55:ea:27  2c:54:cf:e4:a1:d1  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  63 3.738607000  2c:54:cf:e4:a1:d1  2c:54:cf:e4:9e:ce  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  64 3.740399000  2c:54:cf:e4:9e:ce  f8:a9:d0:55:ea:27  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  65 3.740816000  f8:a9:d0:55:ea:27  2c:54:cf:e4:a1:d1  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  66 3.741259000  2c:54:cf:e4:a1:d1  2c:54:cf:e4:9e:ce  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  67 3.742668000  2c:54:cf:e4:9e:ce  f8:a9:d0:55:ea:27  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  68 3.743088000  f8:a9:d0:55:ea:27  2c:54:cf:e4:a1:d1  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  69 3.743526000  2c:54:cf:e4:a1:d1  2c:54:cf:e4:9e:ce  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  70 3.744931000  2c:54:cf:e4:9e:ce  f8:a9:d0:55:ea:27  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  71 3.745354000  f8:a9:d0:55:ea:27  2c:54:cf:e4:a1:d1  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  72 3.745983000  2c:54:cf:e4:a1:d1  2c:54:cf:e4:9e:ce  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  73 3.747583000  2c:54:cf:e4:9e:ce  f8:a9:d0:55:ea:27  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  74 3.748002000  f8:a9:d0:55:ea:27  2c:54:cf:e4:a1:d1  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  75 3.748647000  2c:54:cf:e4:a1:d1  2c:54:cf:e4:9e:ce  192.168.10.100  192.168.10.101  ICMP         98 Echo (ping) request  id=0x6183, seq=1/2
  76 3.750727000  2c:54:cf:e4:9e:ce  84:3a:4b:54:bd:6a  192.168.10.10   192.168.10.100  ICMP        126 Time-to-live exceeded (Time to live exc
```

Fig. 4. Wireshark display showing the effect of ARP-route-looping on a ping from h_A to h_B

While the looping of a single ping message is an interesting curiosity, the serious potential detrimental effects of ARP-route-looping are shown in Fig. 5. In our test network, we examined a scenario where host h_A streams UDP traffic at a rate of 10 kB/s to h_B. Before ARP spoofing, the traffic is delivered and the load on the network (i.e., the total number of UDP packets transmitted in the network) is seen to be approximately 15 packets per second—the blue curve in Fig. 5. After creating a 3-host ARP-route loop (among h_1, h_2, and h_3), the load on the network increases to an average of approximately 305 packets per second—the red curve in Fig. 5. Clearly, in this case ARP-route-looping has created a situation where hosts h_1, h_2, and h_3 expend considerably more energy and occupy the channel for a considerably greater period of time than if the loop were not present.

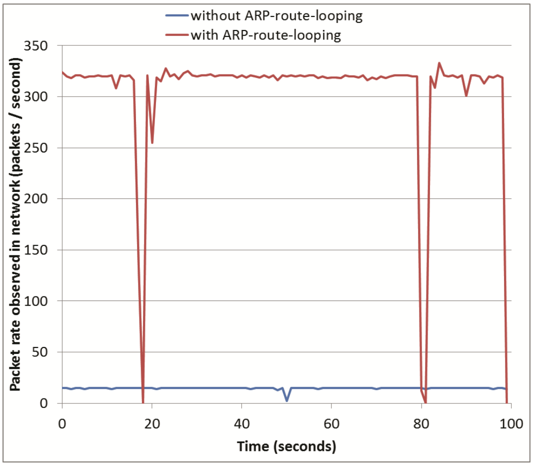


Fig. 5. Packet rate observed for UDP traffic from h_A to h_B with and without ARP-route-looping. (Color figure online)

In general, for an ARP-route-looping attack against host h_A sending to h_B, we expect the packet rate (of affected traffic) to increase by a factor of μ, which we call the traffic multiplication factor. For the UDP example considered here, μ_{UDP} is computed as follows:

$$\mu_{\mathrm{UDP}} = \left(\mathrm{TTL}_{\mathrm{init}} + 1\right) / (\dim(\underline{\boldsymbol{R}}(A, B)) - 1). \tag{3}$$

Here, $\mathrm{TTL}_{\mathrm{init}}$ is the initial time-to-live value for IP packets generated in the network, $\underline{\boldsymbol{R}}(A, B)$ is a vector containing the hosts along the shortest-path route from host h_A to host h_B, and $\dim(\underline{\boldsymbol{R}})$ is the number of elements in vector $\underline{\boldsymbol{R}}$. The numerator contains a "plus one" to account for the additional packet required for the ICMP TTL exceeded message returned to h_A; in the case where ARP-route-looping is also implemented against the return message ICMP TTL exceeded message, then $(\mathrm{TTL}_{\mathrm{init}} + 1)$ is replaced by $(2 \cdot \mathrm{TTL}_{\mathrm{init}})$.

In our test case, $\underline{\boldsymbol{R}}(A, B) = [A, 1, 2, B]$ and $\mathrm{TTL}_{\mathrm{init}} = 64$. Thus we compute $\mu_{\mathrm{UDP,expected}} = (64 + 1)/(4 - 1) = 21.7$. In our UDP example in Fig. 5, we measure the traffic multiplication factor as $\mu_{\mathrm{UDP,observed}}$ = 305 fps/15 fps = 20.3, which is quite close to our expectation.

Finally, we note that Eq. (3) does not account for any return traffic or acknowledgements in computing the denominator. This is valid for the UDP traffic streams considered in Fig. 5; for an ICMP ping, however, we would expect the denominator to be doubled since each ping request results in a ping reply; thus

$$\mu_{\mathrm{ping}} = \left(\mathrm{TTL}_{\mathrm{init}} + 1\right) / (2 \cdot (\dim(\underline{\boldsymbol{R}}(A, B)) - 1)). \tag{4}$$

For TCP, the effect is more complex since it depends at what stage in the TCP session the ARP-route-loop is established. If an ARP-route-loop is created before h_A and h_B

begin a TCP session, then the session is precluded from starting since the initial SYN from h_A will loop in the network without ever reaching h_B. If the session is already established, then TCP traffic—along with myriad retries—will loop among h_1, h_2, and h_3 without ever being delivered to h_B.

5 ARP-Route-Looping Defences

In this section, we briefly outline a few common defences against ARP spoofing, and introduce new mitigations specific to ARP-route-looping.

5.1 Prevention

Since the ARP protocol includes no authentication, other means are needed to prevent an ARP spoofing attack. Well-known methods are listed below.

- Hard-code fixed ARP tables: For all hosts in the network, permanently fix the IP and MAC address mappings and do not use ARP messages. While effective, this may be impractical depending upon the network deployment.
- Dynamic ARP inspection and DHCP snooping: This is a service available on certain switches and validates ARP messages based on previous DHCP assignments. This is not a realistic solution for an ad hoc network, whose hosts likely do not support the service, nor does the network likely utilize a central switch that could drop ARP messages, and it may not employ DHCP.
- Employ a non-standard protocol that includes authentication, e.g., S-ARP [15].
- Do not use ARP: This is a tautological statement, in that by avoiding ARP it is possible to avoid inherent security problems with ARP. For ad hoc networks, however, this is a legitimate and important option; it has been identified in [11] that other techniques should be explored for address resolution.

5.2 Detection

Existing tools such as ARP Watch [8] and ARP Guard [9] allow network administrators to gather a log of IP-MAC address pairs and perform a forensic analysis to determine if ARP spoofing has taken place. These systems identify when IP-MAC pairs have changed and can flag or alert an administrator when this occurs. For unattended ad hoc networks, such systems would provide a means for after-action analysis, but may not be helpful while the attack is occurring. For the specific case of an ARP-route-looping attack in an ad hoc network, we propose the following detection schemes.

- Record/flag duplicate packets: If a host observes that it is forwarding a duplicate packet (i.e., one it has already forwarded), where everything is unchanged with the exception of having a smaller TTL value, this is a strong indication that the host is part of a routing loop. If this behavior is observed over and over again during a short time span, it is all but confirmed.

- Detect duplicate MAC addresses in the ARP cache: If the ARP cache contains two or more different IP addresses corresponding to the same MAC address, this is a red flag in an ad hoc network and may be suggestive that an attacker is attempting to misdirect traffic along unintended routes.

5.3 Mitigation

As noted above, in an ad hoc network without an administrator actively monitoring network health, simply detecting an ARP spoofing attack is not enough—hosts must be able to take action or have methods in place to mitigate the attack as well. For the specific case of ARP-route-looping, we propose two possible mitigation strategies.

- Reduce the default $\mathrm{TTL}_{\mathrm{init}}$: By Eq. (3), if we reduce the initial TTL for packets generated in the network, we in turn will reduce the traffic multiplication factor, μ, thus dampening the severity of the attack. In an ad hoc network with n hosts, if all traffic is expected to be limited to the local network then it is reasonable to set the initial TTL value to n (or less), since packets should not hop through any host more than once. Note that this solution is only practical, however, if it is not expected that traffic will travel outside the ad hoc network.
- Drop duplicate packets: Further to the detection strategy proposed above, not only could hosts detect duplicate packets (where only the TTL value changes), but hosts could drop duplicates when they are seen. This still will not restore connectivity between h_A and h_B, but it will reduce the severity of the resource depletion and channel occupancy.

6 Conclusion

This paper introduced a new application of ARP spoofing that creates routing loops in an ad hoc network, such that hosts in the network continuously forward packets around the network without the packets ever reaching their intended destination. This so-called ARP-route-looping results in hosts depleting their resources—i.e., a resource consumption attack—and increases channel occupancy in the network. In essence, the hosts are misled into a situation where they deny access to the network and resources to one another by amplifying existing network traffic. This is different from typical ARP spoofing attacks, which are often intended to create a man-in-the-middle situation, a situation where traffic is simply dropped, or an ARP flooding situation where the attacker must expend considerable resources.

In this paper we discussed that in ad hoc networks, where every host acts as an endpoint as well as a router, ARP spoofing remains a serious concern and in fact leaves open another vulnerability: ARP-route-looping. While there are many well-known defences against ARP spoofing attacks, unfortunately these are often not implemented. In addition to these well-known preventative techniques, we proposed additional mitigation strategies specific to ARP-route-looping. We hope that this new application will further emphasize the importance of taking steps to avoid this common, yet avoidable, vulnerability.

References

1. Plummer, D.C.: An Ethernet address resolution protocol. RFC 826, November 1982 (1982). http://tools.ietf.org/html/rfc826
2. Cheshire, S.: IPv4 address conflict detection. RFC 5227, July 2008 (2008). http://tools.ietf.org/html/rfc5227
3. Arkko, J., Pignataro, C.: IANA allocation guidelines for the address resolution protocol (ARP). RFC 5494, April 2009 (2009). http://tools.ietf.org/html/rfc5494
4. Mangut, H.A., Al-Nemrat, A., Benzaid, C., Tawil, A.H.: ARP cache poisoning mitigation and forensics investigation. In: Proceedings of 14th IEEE International Conference on Trust, Security, Privacy in Computing and Communications, Helsinki, Finland (2015)
5. Yang, M., Wang, Y., Ding, H.: Design of WinPcap based ARP spoofing defense system. In: Proceedings of 2014 Fourth International Conference on Instrumentation and Measurement, Computer, Communication and Control, Harbin, China (2014)
6. Jinhua, G., Kejian, X.: ARP spoofing detection algorithm using ICMP protocol. In: Proceedings of 2013 International Conference on Computer Communication and Informatics, Coimbatore, India (2013)
7. Salim, H., Li, Z., Tu, H., Guo, Z.: Preventing ARP spoofing attacks through gratuitous decision packet. In: Proceedings of 11th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, Washington DC, USA (2012)
8. LBL Network Research Group, Information and Computing Sciences Division, at Lawrence Berkeley National Laboratory, ARP Watch. http://www.securityfocus.com/tools/142
9. ISL, ARP-Guard. https://www.arp-guard.com/en/arp-guard/product.html
10. Zdrnja, B.: Malicious JavaScript insertion through ARP poisoning attacks. IEEE Secur. Priv. **7**, 72–74 (2009)
11. Carter, C., Yi, S., Kravets, R.: ARP considered harmful: manycast transactions in ad hoc networks. In: Proceedings of 2003 IEEE Wireless Communications and Networking, New Orleans LA, USA (2003)
12. Birmelé, E., et al.: Optimal listing of cycles and st-paths in undirected graphs. In: Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans LA, USA (2013)
13. Shaffer, C.A.: A Practical Introduction to Data Structures and Algorithm Analysis. Virginia Tech, Blacksburg (2010)
14. Sadhir, G., Hu, Y., Perrig, A.: ARP attacks in wireless ad hoc networks (2003). http://dl.icdst.org/pdfs/files/0d65ca5916c99a18d087bad19f6d1d0d.pdf
15. Bruschi, D., Ornaghi, A., Rosti, E.: S-ARP: a secure address resolution protocol. In: Proceedings of the 19th Annual Computer Security Applications Conference (2003)

Investigating Spectrum Sensing Security Threats in Cognitive Radio Networks

Sekgoari Mapunya(✉) and Mthulisi Velempini

Department of Computer Science, University of Limpopo, Polokwane, South Africa
sekgoarimapunya@gmail.com, mthulisi.velempini@ul.ac.za

Abstract. Cognitive Radio Networks (CRN) technology was proposed as a solution to the challenges of overcrowding and underutilization of spectrum bands. CRN is a subset of wireless networks and as such, is susceptible to traditional wireless networks security attacks. In addition, it is also vulnerable to new security attacks such as cooperative sensing related attacks. CRN has an ability to dynamically adapt to the radio environment and thereafter make decisions to access spectrum holes opportunistically.

In this paper, we evaluate spectrum sensing security attacks in CRN. Spectrum sensing is fundamental phase of the cognitive cycle of the CRN however, when compromised; it impacts negatively on the functionality of the cognitive network. Spectrum Sensing Data Falsification (SSDF) attack is one of the security challenges of the CRN and it occurs largely in CRN implementing cooperative spectrum sensing (CSS). CSS is a sensing strategy which increases the detection rate of primary users when secondary users share the sensing data. The SSDF attack degrades the performance of the CRN resulting in the poor utilization of the free spectrum. The study therefore evaluates the Cooperative Neighbouring Cognitive Radio Nodes (COOPON) and the pinokio schemes in a simulated environment. The results show that the COOPON scheme is effective in the mitigation of the effects of malicious users.

Keywords: Cognitive Radio Networks
Cooperative neighbouring cognitive radio nodes
Spectrum Sensing Data Falsification · Pinokio

1 Introduction

The ever-increasing number of wireless devices which utilize free spectrum bands has led to overcrowding of the spectrum while the licensed spectrum is underutilized [1]. The Cognitive Radio Networks (CRN) technology was proposed to address the challenge of spectrum overcrowding and underutilization, where cognitive or secondary users (SU) opportunistically utilize idle spectrum bands licensed to primary users (PU). Spectrum sensing is the most fundamental and vulnerable phase of the cognitive cycle, when cooperative sensing is implemented [2]. CRN is susceptible to both traditional and new security attacks due to its ability to dynamically sense, share, and access the spectrum. This paper focuses on cooperative spectrum sensing (CSS) related security attacks.

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 60–68, 2018.
https://doi.org/10.1007/978-3-319-74439-1_6

In CSS, multiple SUs cooperate in spectrum sensing which makes the network vulnerable to spectrum sensing data falsification attack (SSDF). If spectrum sensing is compromised, it may lead to poor utilization of the spectrum and missed opportunities caused by malicious nodes [3]. The sharing of incorrect spectrum data by malicious nodes is called the SSDF attack which causes interference to both licensed and unlicensed users in CRN [4]. The SSDF attack enables greedy nodes to monopolize the use of the spectrum holes while starving the rest of the nodes. The study investigates spectrum sensing security attacks, the countermeasures and presents the comparative results of security schemes. The study then proposes a security framework for CRN.

2 Related Work

A number of schemes designed to address the effects of the SSDF have been proposed. Most of the schemes implement the data fusion techniques. In [5], a Conditional Frequency Check (CFC) technique based on a Markov Spectrum Model is proposed to mitigate the effects of the Byzantine attacks – the SSDF attacks. With one trusted user, the technique can achieve high detection accuracy of a malicious node without prior knowledge. The assumption of the availability of one trusted node has been adopted in literature. However, when such a trusted node is not available, an additional clustering procedure is required, in attempt to detect the malicious node when the number of non-malicious nodes are more than the malicious ones. The detention window should be wide enough for the scheme to be effective.

In [6], schemes implementing a fusion center (FC) are evaluated. Unfortunately, the schemes are not designed to counter the effects of the SSDF attack. The comparative analysis indicates that the Gaussian assumption is suitable where the SSDF attack is assumed as compared to the Gamma assumption. It was also assumed that the percentage of malicious users (MU) was less than the number of non-malicious users. The algorithm may not perform well as expected if more MUs are considered.

In [7], an extension of the generalized extreme studentised deviation (EGESD) test was proposed to detect selfish nodes in the network. The EGESD was designed to address the limitation of generalized extreme studentised deviation test and it subjects the validity of updates to the Shapiro–Wilk test.

In [8], CRN was implemented in smart home energy management which is susceptible to SSDF. A multi-attribute trust based framework was proposed to facilitate dependable spectrum sensing and to prioritize delay sensitive data transmissions. The evaluation results of the scheme show that it is 91.42% reliability. However, it was assumed that the attacker would always exhibit the always-on attack and different scenarios were not considered.

In [9], a distributed cooperative spectrum sensing (DSCS) with a secure spectrum allocation strategy which is based on the dynamic reputation model and the Vickrey-Clarke-Groves (VCG) was proposed. The evaluation results show that the scheme is effective in addressing the effects of the SSDF attacks. The efficiency of the scheme was compared to the performance of the distributed random scheme in [10].

In [11], a fusion technique is proposed in which spectrum sensing reports are evaluated against a predefined threshold value to detect an attack. The spectrum is said to be occupied by the PU if the reports evaluates to a value which is greater than or equal to the threshold otherwise, it is unoccupied. The change in the value of the threshold has an effect on the results furthermore; it is not optimized for multiple attackers.

In [12], the Weighted Sequential Ration Test (WSRT) is utilized and the scheme consists of reputation maintenance and the hypothesis test. Nodes are assigned a reputation of 0 thereafter with each correct spectrum report the reputation value is incremented by one. The Sequential Probability Ratio Test (SPRT) [13] is then applied. The WSRT differs from the ordinary SPRT because it utilizes a trust-based information fusion scheme. However, there is need to evaluate the efficiency of the scheme.

In [14], a weight based fusion scheme was implemented to counter the effects of a malicious node. It uses trust based and pre-sifting procedures. Permanent malicious nodes are typically of two types, the “Always Yes” and the “Always No”. The “Always Yes” malicious nodes report the presence of the PU which increases the rate of false alarms. The “Always No” advertises the absence of the PU which increases the interference rate. This approach primarily focuses on the pre-filtering of the data to detect the MU and assign the trust value to nodes.

In [15], a detection mechanism that runs in the FC is proposed. The FC detects the attacker by checking mismatches between local decisions and the aggregated decisions and then isolates outliers. The scheme is very effective against Byzantine attacks and it detects MUs within a short time-frame. Unfortunately, the scheme is FC based and infrastructure based.

In [16], a Bayesian detection mechanism that requires the knowledge of prior probabilities of the local spectrum sensing results and the knowledge of prior conditional probabilities of the previous sensing results is proposed. There are a few combination cases that exist between these two cases leading to mismatch in the assignments of the costs. The overall cost is the sum of every cost weighted by the probabilities of the corresponding cases. The scheme cannot detect an SSDF attacker without prior knowledge.

In [17], the Neyman-Pearson Test is proposed that does not require the prior probabilities of final sensing results or any cost associated with each decision case. It defines either a maximum acceptable probability of false alarm or a maximum acceptable probability of missed detection. However, it requires a prior conditional probability of the local sensing.

In [18], a detection technique called pinokio is proposed. Pinokio utilizes a Misbehaviour Detection System (MDS) which profiles the normal behaviour of network nodes based on the training data. The MDS detects MUs by checking the bit rate behavior of nodes. The bit rate has to change occasionally. Nodes not exhibiting the normal expected behavior are classified as outliers. The challenge with the proposed scheme is the assumption that mobile nodes move at a low speed. Higher mobility speeds may impact negatively the performance of the scheme.

COOPON, a simple and efficient detection scheme designed to detect selfish nodes in CRAHN known as SSDF attack is proposed in [19]. The scheme detects the availability of selfish MUs through the help of neighboring nodes. The target SU and its neighbors exchange observed radio environment data, which is evaluated by all SUs to detect selfish malicious nodes. Then, each SU compares the reported data and if there is any difference, a given node is classified as the outlier.

3 Simulation Model

In the section, comparative performance results of the COOPON and Pinokio are presented. The two schemes are designed to counter the effects of the SSDF attacks. The schemes were simulated using the network simulator 2.31 (NS 2.31). Table 1 presents the simulation parameters used in the simulation.

Table 1. Simulation parameters.

Parameter	Values
Antenna type	OmniAntenna
Propagation model	TwoRayGround
Simulation area	500 m * 500 m
Mobility model	Random Waypoint
Node speed	20 m/s
Routing protocol	Ad hoc on-demand multipath distance vector routing
MAC protocol	IEEE 802.11b with extension to support CR networks
Data channel	8
Common control channel	1
Channel data rate	11 M bits/s
Number of SUs	50, 100, 150
Percentage of selfish SU	2%, 10%, 50%, 75%

Table 1 shows the parameters that were used in the modeling of the simulation environment. The simulation time was set to 300 simulation seconds. The cognitive radio network was assumed to be having a transmission radius of 500 m. We considered CRN with eight data channels and one common control channel for the exchanging of control packets between the SUs. The data channel rate was set to 11 Mb/s. It was also assumed that SUs can have at least two neighbors and a maximum of five neighbors.

The detection efficiency of the scheme was measured based on the probability of detection, which is the probability of a CR user positively detecting that a licensed user is present.

4 Results

The detection rate was considered in the evaluation of the performance of the COOPON and Pinokio schemes which were chosen based on the fact that they can be deployed in a cognitive radio ad-hoc networks. The COOPON detects MUs through the implementation of the MDS which profiles the normal behavior of nodes. The MDS detects anomaly behavior by monitoring the bit rate behavior of nodes. There must be periodic change in bit rate which is adjusted continuously by a node. For example, narrow channels use a low bit rate. Nodes which fail to exhibit the expected behavior are classified as outliers. Figure 1 presents the detection rate of the COOPON scheme.

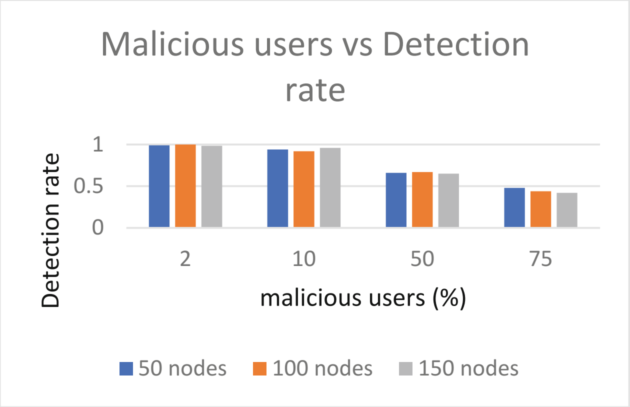


Fig. 1. Detection rate vs. malicious users in COOPON scheme.

To investigate the impact of MUs on the performance of the CRN, the evaluation was performed in network scenarios with 50, 100 and 150 SUs as shown in Fig. 1. It can be seen that the number of users in the network has an impact on the COOPON's detection rate, as the number of nodes increases in the network the detection rate decreases. Figure 2 presents the detection rate of the Pinokio scheme.

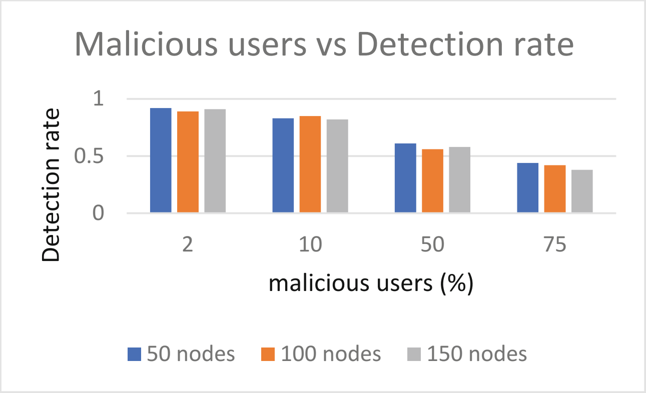


Fig. 2. Detection rate vs. malicious users in Pinokio scheme.

The impact of node density on the performance of the Pinokio was investigated in Fig. 2. The number of nodes was increased from 50 to 100, and then to 150. Figure 2 shows that the density of SUs in the network has a negative impact on the detection rate of the Pinokio scheme. The detection rate decreases as the nodes are increased in the network. In Fig. 3, the comparative results of the two schemes are presented.

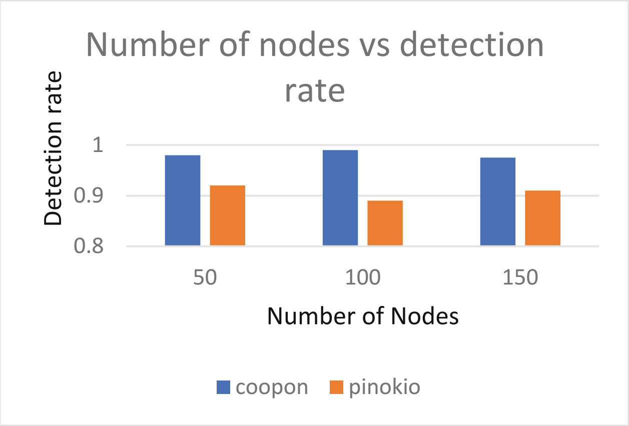


Fig. 3. Number of nodes vs. detection rate with 2% malicious nodes.

Figure 3 shows the comparison of the detection rates of the two schemes when 2% of the total network nodes are malicious nodes. As shown in Fig. 3, the COOPON scheme achieved a higher detection rate than the Pinokio scheme in the three scenarios. It is therefore superior to the Pinoko scheme. Figure 4 considered a network with 10% of the nodes being malicious nodes.

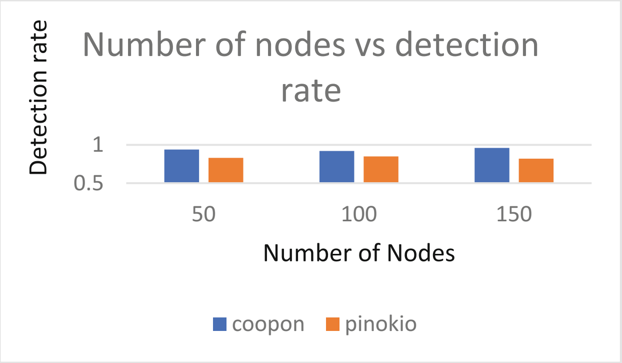


Fig. 4. Number of nodes vs. detection rate with 10% malicious nodes.

Figure 4 show that when there are 10% of malicious nodes and 90% of non-malicious nodes the COOPON scheme outperforms marginally the Pinokio scheme which suggest that the COOPON scheme was degraded severely by the increase in the number of malicious nodes. The results in Fig. 5 confirm this assertion.

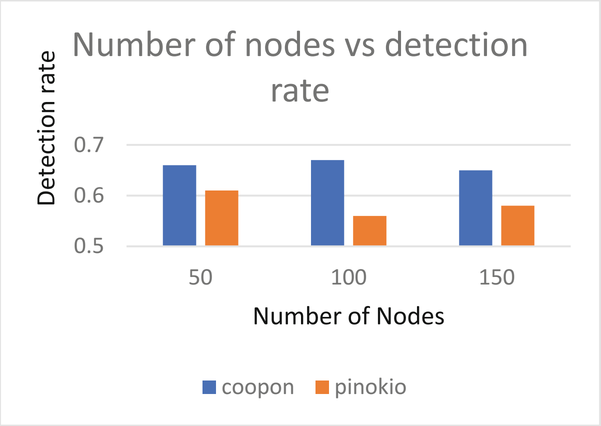


Fig. 5. Number of nodes vs. detection rate with 50% of malicious nodes.

Figure 5 shows the comparison of detection rates of the two schemes when the network consists of 50% malicious nodes. The results show that the performance of the COOPON scheme is marginally better than the performance of the Pinokio scheme as observed in Fig. 4. The degradation in the performance of the COOPON scheme in the presence of increasing number of malicious nodes is evident in Fig. 6.

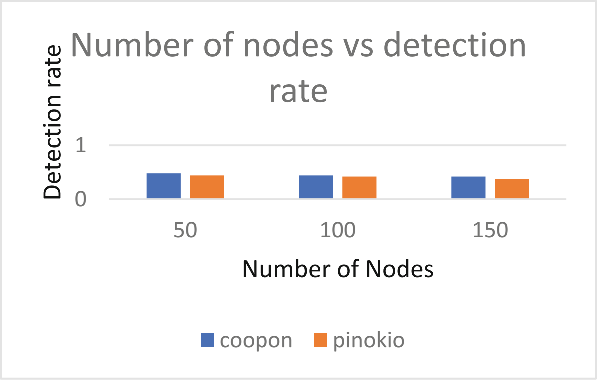


Fig. 6. Number of nodes vs. detection rate with 75% malicious nodes.

In Fig. 6, it can be noted that the performance of the two schemes are almost the same. In this case, 75% of the total nodes were malicious nodes. This proves that the COOPON scheme degrades gracefully with the increasing number of malicious nodes in the network. This indicates that, when the network has a higher percentage of malicious nodes, the COOPON scheme may be outperformed by the Pinokio scheme.

5 Future Work

There is a need to develop a new scheme optimized for the SSDF attacks in cognitive networks. The scheme should be designed to detect many malicious nodes. The scheme should be well designed to ensure that the increasing number of malicious nodes does not degrade its detection rate. We propose a new scheme which employs the extreme studentised deviation test to mitigate the SSDF attack in an ad hoc cognitive radio network. The scheme is designed to counter the effects of a number of malicious nodes. The scheme will be evaluated through numerical and analytical techniques.

6 Conclusion

The comparative evaluation results of the COOPON and Pinokio SSDF attacks mitigation schemes for cognitive radio show that the SSDF countermeasures are also susceptible to the effects of SSDF. Their performance degrades gracefully as the number of the malicious nodes increase in the network. There is need for robust and more resilient SSDF security schemes for better and improved network performance. Alternatively, the current best performing scheme can be modified to enhance the performance of the CRN in the presence of malicious users.

References

1. Monika, B., Chandra, K.R., Kumar, R.R.: Spectrum sensing techniques and issues in cognitive radio. IJETT **4**(4), 695–699 (2013)
2. Dobaria, A., Sodhatar, S.: A literature survey on efficient spectrum utilization: cognitive radio technology. Int. J. Innov. Emerg. Res. Eng. **2**(1), 72–75 (2015)
3. 2015-caps-infocus, 3 December 2015. http://www.capsindia.org/2015-caps-infocus. Accessed 6 June 2016
4. Chetan, M., Subhalakshami, K.: Security issues in cognitive radio. In: Cognitive network: Towards Self-Aware Networks (2007)
5. Xiaofan, H., Huaiyu, D.: A Byzantine attack defender in cognitive radio. In: IEEE International Symposium on Information Theory, Cambridge (2012)
6. Lavanis, N., Jalihal, D.: Performance of p-norm detector in cognitive radio networks with cooperative spectrum sensing in presence of malicious users. Wirel. Commun. Mob. Comput. **2017**(2), 1–8 (2017)
7. Srinu, S., Mishra, A.K.: Efficient elimination of erroneous nodes in cooperative sensing for cognitive radio networks. Comput. Electr. Eng. **52**, 284–292 (2016)
8. Premarathne, U.S., Khalil, I., Atiquzzaman, M.: Trust based reliable transmissions strategies for smart home energy consumption management in cognitive radio based smart grid. Ad Hoc Netw. **41**, 15–29 (2016)
9. Lin, H., Hu, J., Huang, C., Xu, L., Wu, B.: Secure cooperative spectrum sensing and allocation in distributed cognitive radio networks. Int. J. Distrib. Sens. Netw. **2015**, 194–206 (2015)
10. Luo, L., Roy, S.: Analysis of search schemes in cognitive. In: Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2007), pp. 647–654, June 2017

11. Pandharipande, A., Kim, J.M., Mazzarese, D., Ji, B.: Wireless RANs: technology proposal package for IEEE 802.22. In: IEEE 802.22 WG on WRANs (2005)
12. Ruiliang, C., Jung-Min, P., Thomas, H.Y., Jeffrey, H.: Toward secure distributed spectrum sensing in cognitive radio networks. IEEE Commun. Mag. **46**, 50–55 (2008)
13. Shei, Y., Su, Y.T.: A sequential test based cooperative spectrum sensing scheme for cognitive radios. In: 2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, Cannes (2008)
14. Khabbazian, M., Kaligineedi, P., Bhargava, V.: Secure cooperative sensing techniques for cognitive radio systems. In: 2008 IEEE International Conference on Communications, Beijing (2008)
15. Rawat, A.S., Anand, P., Chen, H., Varshney, P.K.: Countering byzantine attacks in cognitive radio networks. In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX (2010)
16. Lu, L., Chang, S.-Y., Zhang, J., Qian, L., Wen, J., Lau, V.K.N., Cheng, R.S., Murch, R.D., Mow, W.H., Letaief, K.B.: Technology proposal clarifications for IEEE 802.22 WRAN systems. In: IEEE P802.22 Wireless RANs, May 2006
17. Hillenbrand, J., Weiss, T., Jondral, F.: Calculation of detection and false alarm probabilities in spectrum pooling systems. IEEE Commun. Lett. **9**(4), 349–351 (2005)
18. Tan, K., Jana, S., Pathak, P., Mohapatra, P.: On insider misbehavior detection in cognitive radio networks. IEEE Netw. **27**(3), 4–9 (2013)
19. Jo, M., Han, L., Kim, D., In, H.P.: Selfish attacks and detection in cognitive radio ad-hoc networks. IEEE Netw. **27**(3), 46–50 (2013)

Integrating Intrusion Response Functionality into the MANET Specific Dynamic Intrusion Detection Hierarchy Architecture

Manpreet Kaur, Dale Lindskog(✉), and Pavol Zavarsky

Concordia University of Edmonton, Edmonton, Canada
mkaur2@student.concordia.ab.ca,
{dale.lindskog,pavol.zavarsky}@concordia.ab.ca

Abstract. In this paper, our interest is intrusion response in mobile ad hoc networks (MANET). All intrusion response systems (IRS) presuppose an underlying intrusion detection system (IDS). We propose improvements to an existing dynamic and hierarchical IDS architecture for MANETs, proposed by Sterne et al. Our improvements are designed to enhance its ability to form an underlying base IDS for an imagined IRS. The enhancements are chosen to overcome the lack of resiliency in the selected architecture, by adding backup cluster heads and a backup root node. Additionally, we also propose revisions designed to avoid giving the root node too much authority over intrusion response, by distributing that power among cluster heads. The root node acts, rather, as an attack information database. The cluster heads, we propose, would make use of a MANET specific intrusion response algorithm proposed and described by Kaur et al.

Keywords: Mobile ad hoc networks · Intrusion detection · Intrusion response Clustering · OLSR

1 Introduction

A MANET is a wireless network which consists of some number of wireless mobile hosts that form a temporary network, a network without the presence of any centralized administration or infrastructure, such as access points, base stations, mobile switching centers, etc. Since there is this lack of centralized infrastructure, mobile nodes are required to cooperate with one another, in order to perform network related operations such as routing and packet forwarding. In a MANET, nodes have freedom to enter into, leave from, and move around within the network. Therefore, changes in network topology can occur at any time. Due to such characteristics, MANETs are more difficult to protect against intrusions [2, 3].

An IDS is used to detect, analyze and report intrusions, and has become an indispensable component of a defense-in-depth security approach for MANETs [4]. More recently, there has been interest in intrusion response in MANETs. But due to their peculiar characteristics, mentioned above, designing either an IDS or an IRS is considerably more complex for MANETs than for traditional networks. For MANETs, it is

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 69–80, 2018.
https://doi.org/10.1007/978-3-319-74439-1_7

impossible to choose a specific set of nodes in the internetwork to perform the intrusion detection or intrusion response functions, since the ability of effectively perform such functions depends upon their placement in the internetwork, which, in a MANET, is subject to change, and potentially frequent change.

Much has been written about methods of detecting intrusions in MANETS, and differences between these methods depend to a great degree on the type of IDS architecture, which can be usefully categorized into three types. The simplest type is the Stand-alone IDS architecture, where each node of the network has a detection engine/IDS agent installed on it. Nodes are responsible themselves for detecting intrusions [5]. A more sophisticated architecture is the Distributed and Cooperative architecture, proposed by Zhang and Lee [6], which is designed so that neighboring IDS agents will cooperatively participate in MANET wide IDS. Both of these IDS architectures are suitable for a flat network infrastructure, but not for a multi-layered network divided into groups of clusters and organized into hierarchies [5, 7]. For a multi-layered infrastructure, a Hierarchical IDS architecture has been proposed, which is organized into levels, with a so-called root node present at the top of the hierarchy. The network is subdivided into groups called clusters. Each cluster has its ‘cluster head’ which acts as a control point and has more responsibility than other nodes for providing communication to other cluster heads and zones. In this architecture, local detection is carried out by a cluster, whereas global detection is carried out by cluster heads and inter-zone nodes [7].

It is often desirable to have mechanisms by which the network can respond against detected intrusions. Badie et al. [8] noted that, unlike a traditional fixed topology, in a MANET the node best suited to execute a response will often need to be determined at run time. Moreover, an attack detecting node might or might not be in a suitable position to execute the response instruction, in which case two problems arise: how is it determined which node is most suitable for executing a response, and how is a response instruction sent to that node? They argued that a Hierarchical IDS architecture would form the most suitable base for a MANET specific IRS, because of the presence of the root node at the top of the hierarchy. Being at the top of the hierarchy, and receiving aggregated information from below, the root node has, in theory, the most comprehensive picture of the attack and therefore is in the best position to make decisions about intrusion response. But to make informed choices about response, the root node needs a current and updating conception of the topology of the network, in order to know the current position of the attacker, the victim, and the identity of that node best placed to execute the response. They hinted that the MANET specific optimized link state routing protocol (OLSR) could be used to acquire the current network topology, and to quickly detect changes to it [8].

Kaur et al. [2] expanded on this these ideas, and developed an algorithm that could be implemented on the node responsible for determining intrusion response (the root node of the Hierarchical IDS architecture, in their example). This algorithm would rely upon the OLSR protocol for network topology related information. Their proposed algorithm works in a query-response mode, where the IRS function of the IDS root node queries the implemented algorithm to answer network topology related questions, which then returns a response, relying on the OLSR for its database of topology related information. By querying the implemented algorithm, a node determines the positions of the attacker and victim in the overall MANET, and most importantly, determines other network topology related information that it needs to judge which node is best suited to execute the response.

We selected an existing IDS architecture, proposed by Sterne et al. [1]. Because of various features of their proposed IDS architecture, described in Sect. 2 of this paper, it forms a very suitable starting point on which to build the IRS related solutions proposed by Badie et al. and Kaur et al. However, we observed that Sterne et al.'s IDS architecture has also some limitations, discussed later. Therefore, we enhanced it with two revisions. First, we increase its resiliency, by introducing backup cluster heads and a backup root node, and additionally, we demote somewhat the authority of the root node, by proposing cluster heads as the executers of response instructions. Since the cluster heads will have the responsibility to respond to attacks, we imagine they will run the algorithm proposed by Kaur et al. [2] to facilitate the intrusion response functionality. The root node's primary responsibility will be to maintain a comprehensive IDS information database, allowing cluster heads to consult that database while making decisions about intrusion response.

Our concern in this paper is not to identify the structure and nature of the response instruction that a cluster head would send, nor methods to determine which nodes should respond, and how. Rather, our purpose is to describe an IDS architecture conducive to such a MANET specific intrusion response system, and one that will make use of Kaur et al.'s algorithm. The rest of this paper is organized as follows: Sect. 2 reviews types of MANET IDS architectures, OLSR, the related research on IRS in MANETs, and a brief explanation of the algorithm developed by Kaur et al. [2]. Section 3 is our justification for selecting a particular type of Hierarchical IDS architecture. The following section describes imperfections in the selected IDS architecture, proposes enhancements to that architecture to make it more resilient, and proposes revisions to address the danger of a single node, the root node, having too much authority over intrusion response. Finally, Sect. 5 concludes.

2 Review of Related Research

Mobile devices in MANETs, often simply referred to as nodes, are free to leave from, enter into, and move around within the network. Due to such characteristics, intrusion detection systems designed for traditional wired and even wireless networks cannot be applied to MANETs directly. We dedicate the first part of this section to an overview of various types of MANET specific IDS architectures. Existing IDS architectures for MANETs fall under three basic categories, and we discuss them below, along with their problems.

2.1 Types of MANET Specific IDS Architectures

Existing IDS architectures for MANETs fall under three basic categories; these are discussed below along with their problems [9]:

1. Stand-alone IDS Architecture
   In the Stand-alone IDS architecture, an IDS agent is installed on each node and runs independently to detect intrusions locally. This architecture is very limited, and amounts primarily to a mere host intrusion detection system.
   These types of IDS are inherently limited in their ability to detect attacks, because decisions about intrusion or attack must be based only on information available to individual nodes. That is to say, a stand-alone IDS architecture does not engage in cooperative detection. Chadli et al. [9] presented a study and analysis of the different proposed IDS architectures for MANETs, where they identified strengths and weaknesses of the different proposed IDS architectures. They describe various different stand-alone IDS architectures, including but not limited to: Battery-Based IDS, Threshold-based IDS, and Two-stage IDS. But according to their analysis, these architectures are prone to false-positives and negatives, and also introduce new security weaknesses [9].
2. Cooperative and Distributed IDS Architecture
   In this type, an IDS agent is again installed on each node of the MANET. However, the IDS agent's responsibilities involve not only collecting and analyzing evidence obtained locally, but furthermore, nodes share data with neighboring agents in an effort to detect attacks based on a wider range of information. Zhang et al. [6] proposed this type of architecture by considering the salient features of the MANET. IDS architectures should, they argued, be distributed and cooperative, because MANETs are distributed and network nodes cooperate with each other.
   But this architecture and various other Cooperative IDS architectures, such as the Friend assisted IDS architecture, the Cooperative IDS architecture based on social network analysis, etc., have limitations, studied and analyzed by Chadli et al. [9]. They argued that, for the entire set of architectures of this type that they studied, the rate of false positives and the detection accuracy is negatively affected by the high mobility of nodes. Moreover, the majority of them are vulnerable to various attacks, such as man in the middle, session hijacking, blackmailing, etc. Another weakness they found was that almost all cooperative IDS architectures impose extra processing and communication overhead [9].
3. Hierarchical IDS Architecture
   The Hierarchical IDS is an advanced version of the distributed and cooperative IDS architecture. This IDS architecture is organized into levels. This architecture divides the network into groups called clusters, where 'clustering' refers to the virtual partitioning of the network, and the arrangement of nodes into clusters, with each cluster having a 'cluster head', and with the other nodes of a cluster referred to as cluster members. There is a root node present at the top of the hierarchy that periodically gathers the aggregated intrusion related information from the lower level cluster heads. Cluster heads detect any malicious activity in their own cluster. Cluster heads report intrusions to higher level cluster heads, and this process continues until the information reaches the root node [1].
   If the IDS root node or any cluster head that performs IDS functionality becomes unavailable or is compromised, then the IDS may be compromised. For example, if an attacker takes control of a root node, then the detection system will not be able

to detect attacks. If any cluster head is down, then the root node will be unable to receive attack information from that particular cluster, and in such cases, the root node will be in a poorer position to detect or fully understand an intrusion, for its conclusions may be based on less complete information. Thus, this type of IDS architecture has one major drawback, i.e. a single point of failure, i.e. the root node, and also various ‘single points of weakness’, namely the cluster heads. This situation is even more serious if the IDS root node is also performing an IRS function. For if the root node is compromised, then an attacker may initiate or execute a fake intrusion response against the network. Therefore, it is important to have resiliency in this type of IDS architecture, and to mitigate the consequences of IRS compromise. Sterne et al. [1] proposed a specialized hierarchical IDS architecture for MANETs, which they termed the *cooperative intrusion detection* architecture. It is designed to facilitate accurate detection of MANET-specific attacks. The architecture is described as a dynamic hierarchy that can be structured into more than two levels, and organized into clusters. Each cluster has a cluster head that performs almost the same functions as discussed earlier in the hierarchical IDS architecture. Additionally, cluster heads also perform (1) data fusion/integration and data filtering, (2) computations of intrusion, and (3) security management. Every member node of a cluster monitors, logs, analyzes, responds, and alerts or reports to cluster heads [1]. The authors also explain the process by which intrusion detection information is propagated up the hierarchy. To maintain the hierarchical structure of the IDS, a clustering algorithm is run, and the process of clustering continues until all nodes in the network are part of the hierarchy.

Though Sterne et al.’s proposed IDS architecture is suitable for detecting a wide range of MANET-specific and conventional attacks, the question we are interested in is whether it is suitable as an IDS foundation for a MANET IRS. Our answer to this question is provided later in this paper, but before that, it is important to understand how intrusion response works in MANETs, and we dedicate the next subsection to this topic.

2.2 Intrusion Response System in MANETs

There are an indefinite number of different possible response actions, such as node isolation, session disruption, tarpitting, packet filtering, disabling portions of the network, blocking ports, etc. Regardless of the specific response, it is important to note that, because MANETs have no fixed network topology, the node (e.g. the root node of the Hierarchical IDS architecture) that detects the intrusion might not be in the best, or even in an appropriate position to execute the response. Therefore, it becomes necessary for the root node to send a response instruction to a more suitable node, and for that more suitable node to instead execute the response. In MANETs, according to Badie et al., “An effective IRS often depends on an accurate conception of network topology, and it requires some method for the IRS to discover which particular node is in the best position to execute the response” [8]. For MANETs, there are complexities involved in making that determination, because a MANET does not have a fixed network topology.

What is clearly required is a current conception of the network topology at the time the response instruction is to be sent.

Badie et al. further explained that an efficient manner in which to determine the current network topology is to rely on the topology database used in link state routing protocols, and they remarked in passing that the MANET specific optimized link state routing protocol (OLSR) is a link state protocol, and therefore appears to be a good choice. They imagined that “The root of the hierarchy, in the Hierarchical IDS architecture, receives aggregated information from the cluster heads. Then, the root node, having a complete picture of attack and the topology of the network, can choose the most appropriate node to execute the response” [8].

Kaur et al. expanded on this, and proposed an algorithm that relies on the OLSR database to compute answers to relevant network topology related questions.

The proposed algorithm operates in a query-response mode, and would be employed by the root node of the IDS/IRS. The root node sends a query as input to the implemented algorithm, and it outputs the response, formatted as an unordered list of nodes satisfying the query. In the hierarchical IDS architecture, the root is placed at the top of the hierarchy and receives consolidated intrusion detection information. After examining the received information and detecting the attack, the root node may decide to perform intrusion response functions based on that consolidated information. At that time, the IDS root node becomes an IRS root node. When the IRS root node decides to send one or more response instructions, it must have current information about the topology of the network. To determine the relative position, in the network, of the attacker, the victim, and of the most suitable node(s) to respond, the IRS root node asks certain kind of questions of the implemented algorithm. According to Kaur et al., these are the three general questions that IRS root node may ask:

- Which nodes are currently in the network?
- Which nodes are n hops away from X?
- Which nodes are in-line between X and Y?

(where ‘n’ ranges over positive integers, and ‘X’ and ‘Y’ range over nodes in the MANET).

After receiving the input from the IRS root node, the implemented algorithm returns output which helps the IRS root node to determine the most suitable node to respond to the attack or intrusion. To generate results and give output, the algorithm relies on the OLSR protocol. The next subsection briefly explains how the OLSR protocol operates in MANET.

2.3 Optimized Link State Routing (OLSR) Protocol

OLSR, defined in RFC 3626 [10], is a proactive routing protocol developed specifically for MANETs, and optimized for networks with rapidly changing topologies. OLSR uses two types of messages, ‘HELLO’ and ‘TC’ (Topology Control), to maintain current topological information about the network at every node. HELLO messages are used to discover the immediate neighbors (one-hop neighbors), and then two hop neighbors are discovered from the answers given by one-hop neighbors. The information about a one hop neighbor's link status, and information about its two hop neighbors, is contained in a neighbor table that is maintained by each node. Topology control messages are broadcast by MPRs (Multipoint Relays) to advertise links or topological information throughout the network. Based on this topological information, every node calculates its topology table. This topology table contains information about the topology of the network obtained from the TC message, and information about the MPRs of the other nodes. In this way, OLSR maintains the topology database for the network at each node by periodically exchanging HELLO and TC messages with neighboring nodes. Topology and neighbor tables are maintained by OLSR at every node, and it is this which makes them able to calculate the best route to a destination node.

Kaur et al.'s [2] algorithm uses the OLSR protocol to determine facts about the topology of the network, such as the total number of nodes and positions of specific nodes in the network, e.g., the attacker and victim.

3 Selecting a Suitable IDS Architecture

One of the main concerns in choosing an IDS that will smoothly interoperate with an IRS is whether the IDS facilitates an IRS's need to determine which action to perform, and against which node and from which node that response action will be performed. The details surrounding this question are not in the scope of this paper, for to determine which particular response related actions need to be taken depends upon details such as the type of the attack detected by an underlying IDS, and the placement of the attacking and attacked nodes at the time of the attack.

However, these concerns do indicate, in a general way, a need to have mechanisms in place for effective decision making about the response instruction to be given, and about which node should execute that response.

We believe that the answer to this general question is that such decision making should be done by making use of the most attack- and network activity- related information available to the IDS, e.g., by selecting a node that has more consolidated/aggregated intrusion detection information as compared to the other nodes of the network, and for the following reasons. First, the more information that is known about the attack, the better the intrusion response will be. That is to say, the less complete the information about the attack, the greater the possibility of incorrect decisions, not only decreasing the effectiveness of the IRS, but significantly, increasing the likelihood of harmful responses, e.g. responses against innocent nodes or the network as a whole. This is especially true of attacks designed to cause the IRS to respond against innocent nodes, or that affect the functioning of routing protocols. Second, the rate of false positives would be significantly lower, that is, the IDS would be less likely to raise alarms in response to normal network behavior. This again is extremely important for any IDS underlying an IRS, since false positives are not just time wasting, as in the case of an IDS, but can in an IRS be very harmful, since, as noted, they may generate responses against innocent nodes.

Of course, these concerns are valid for IRSs generally speaking, but they are especially important in the case of a MANET specific IRS, because in MANETs it is particularly difficult to avoid misleading attack or network related information, due to the fact that a MANET does not have a fixed network topology. This can be made more clear with the following example. Consider the topology depicted in the Fig. 1, and consider also the following, very simple attack scenario.

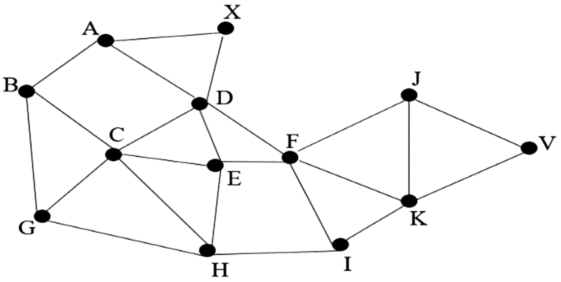


Fig. 1. MANET topology

Suppose node A attacks node V, but in doing so spoofs its IP address to make it appear that the attack has originated from node B. Suppose further that the victim node, V, detects this attack, but (incorrectly, and understandably) judges that it is coming from node B. With a stand-alone or distributed cooperative IDS architecture, this error in detection is a serious risk, and if such an architecture formed the base of an IRS, it is quite easy to imagine that a response instruction is sent to the neighboring nodes of B, e.g. nodes A, C and G. And that response instruction, depending on its details, could in effect perform a denial of service or other attack on the innocent node.

A. For example, such would be the effect if the response instruction was either to perform packet filtering for all the packets sourced from node B, or, e.g., to stop forwarding any traffic coming from and going to node B. Thus, node A would have exploited the IRS system itself, and node A might even continue the attack, as no intrusion or attack response was initiated against it. In this case, the IRS is not only ineffective but harmful.

Let us now discuss the same scenario by imagining the hierarchical IDS architecture underlying an IRS. In this case, there will be a root node at the top of the IDS hierarchy, and the network will be organized into levels in some way. Suppose, in Fig. 1 above, that node X is the root node and that the other nodes are organized into hierarchies. With this type of IDS architecture, the dissemination of intrusion or attack information is done in such a way that nodes at the lower level of the hierarchy send intrusion or attack information to the nodes stationed at their immediate upper level of the hierarchy, and this dissemination of information continues until the information, or an aggregate of it, reaches the root node (node X in our example).

In our example, referring to Fig. 1, it is the root node X that is receiving consolidated/aggregated intrusion and attack information. It is very easy to imagine an IDS so configured that information about spoofed IP packets is contained in the aggregate, detected, e.g., because of mismatches or suspicious pairings between MAC and IP addresses were noticed by adjacent nodes (e.g. nodes adjacent to the node A!). The root node would, therefore, be in a much better position to judge that the attack, apparently sourced from B, is in fact sourced from node A. The root node would, of course, therefore be in a much better position to cause an intrusion response instruction that actually targets the guilty node, e.g. by determining all possible routes from node A, and sending a response instruction to all neighboring nodes of A, to stop forwarding A's traffic. In this case, the IRS is clearly more effective, and less dangerous, when compared with the previous case. In conclusion, the hierarchical IDS architecture is, other things being equal, a superior form of MANET specific IDS architecture on which to base an IRS. There are, however, many different sub-types of Hierarchical IDS architecture described in the literature. The best sub-type for our purposes is, as discussed earlier, that proposed by Sterne et al. They proposed a dynamic hierarchy architecture that acts as a foundation for all intrusion activities in mobile ad hoc networks.

We selected Sterne et al.'s IDS architecture for various reasons: It is a general IDS architecture, and thus not restricted to MANET-specific attacks. It uses a dynamic hierarchy designed to accommodate nodes and links which might appear and disappear rapidly, even under normal network activity. It explicitly identifies particular nodes as having consolidated/aggregated intrusion detection related information, and intrusion detection and correlation occurs at the lowest level in the hierarchy at which the aggregated data is sufficient to enable an accurate detection or correlation decision. Finally, the responsibilities of nodes and cluster heads are clearly defined, and the operation of the architecture is also depicted with scenarios that carefully illustrate its intended operation and features.

4 Enhancing the Proposed Enhanced Dynamic Intrusion Detection Hierarchy Architecture

Though Sterne et al.'s architecture has commendable features, there are serious problems of resilience. There is a single point of failure at the root node. The root node, so important because it stores the greatest amount of consolidated/aggregated intrusion detection information, can itself be under attack, or can otherwise fail. There is a similar problem with the cluster heads. The cluster head of each cluster is responsible for monitoring traffic and detecting intrusions local to its cluster. If a cluster head is compromised or otherwise fails, and if there is no node at a higher level that is in a position to perform IDS functionality in that specific circumstance, we again have a single point of failure, this time at the level of the cluster as opposed to the hierarchy as a whole. Thirdly, if there is a denial of service attack or network congestion, then the root node may be unable to receive intrusion detection data from nodes at lower levels of the hierarchy. For example, suppose a cluster head is under a denial of service attack and cannot forward intrusion detection information, or at least is delayed in forwarding such information to the cluster head one level above in the hierarchy. Then the root node will be unable to get information from one cluster, i.e., from one portion of the network. The root node will therefore be working with incomplete information. Finally, the architecture depends upon the reliability of upper level cluster heads to forward upward their aggregated information toward the root node, and if any fail to do so, this too constitutes a group of single points of failure in the system, a group that grows commensurate with the size of the network.

To enhance resiliency in the dynamic intrusion detection hierarchy architecture, we propose two root nodes, and two cluster heads in each cluster, as suggested by Ishaq [11]. One of each will act as a primary, and the other as a backup. Consider the illustration shown in Fig. 2. The network is divided into five clusters. Out of five, three clusters (C1.A, C2.B, and C1.C) are at the first level of the hierarchy, and the remaining two clusters, namely C2.A and C3.A respectively, are the second and third level representatives of the hierarchical architecture. The first, second and third level cluster heads are annotated by ‘1’, ‘2’, and ‘3’ respectively. The root node is at the top of the hierarchy. Each cluster has a primary (denoted by 1, 2 and 3) and a backup (green) cluster head. The arrows originating from the member nodes and backup cluster heads, and pointing to the first level cluster heads, represent the propagation of intrusion detection information. Further, the arrows pointing from the first level cluster heads, to the second level, depict the propagation of aggregated intrusion-related information. This process of consolidating intrusion detection data from the lower- to upper-level hierarchies continues until it reaches the root node, as suggested by Sterne et al. [1].

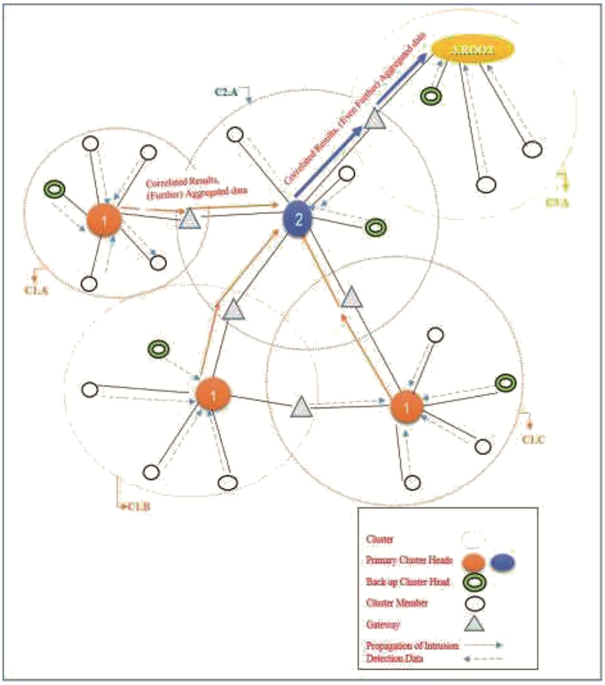

3 ROOT
C2.A
C3.A
Correlated Results, (Even Further) Aggregated data
Correlated Results, (Further) Aggregated data
1
2
1
1
C1.A
C1.B
C1.C
Cluster
Primary Cluster Heads
Back up Cluster Head
Cluster Member
Gateway
Propagation of Intrusion Detection Data

Fig. 2. Enhanced dynamic intrusion detection hierarchy architecture for MANETs (Color figure online)

To organize the architecture hierarchically, we prefer to use a clustering scheme that applies the Weight-Based Hierarchical clustering algorithm. Details of this clustering algorithm can be found in [12] but the general approach is that all nodes use some set of factors (such as mobility, transmission, battery power, bandwidth, etc.) to compute weight. Then, the selection of the root and the cluster heads is made fairly from the calculated weight. Two cluster heads would then be elected from each cluster. The criterion to choose the backup cluster head is large transmission range (LTR). A primary cluster head would select as its backup the node within its cluster that is in its best transmission range. Then, if a primary cluster head fails in a detectable way, the backup cluster head will take over its responsibilities, and thus the cluster head would not constitute a single point of failure.

Recall that the root node is at the top of the IDS hierarchy and receives aggregated/consolidated intrusion detection information from the entire network, and therefore in theory has the most complete information. It may seem natural, for this reason, to assign to it the function of determining and communicating response instructions. But assigning both IDS and IRS functionalities to a single entity is a major risk, as that node might misuse its power. For this reason; we propose that the root node does not perform IRS functions, but rather, that authority over IRS functions is distributed to the cluster heads collectively. This not to say that there would be a change in the mechanism of disseminating intrusion detection information from the lower level hierarchies to the root node. The root node will continue to receive consolidated/aggregated information but will not detect or respond to attacks. Rather, the root node will act as an attack information database, consulted by cluster heads in the process of detecting intrusions.

5 Conclusion

In MANETs as elsewhere, an IRS presupposes an underlying IDS. In MANETs, intrusion detection is complicated by the fact that the network topology is dynamic, and for this and other reasons, various MANET specific IDS architectures have been proposed in the literature. Intrusion response is further complicated by the fact that the best node to execute a response must often be determined at run time, again because MANET topologies are dynamic. We chose Sterne et al.'s DSt05 IDS architecture to form a base IDS architecture for the purpose of intrusion response in MANETs. However, their proposed IDS architecture lacks resiliency, and therefore this paper proposes an enhanced dynamic intrusion detection hierarchy architecture that we argue forms a better basis for an IRS. The proposed enhancements include removal of single points of failure, and distribution of power over intrusion response instruction execution.

References

1. Sterne, D., Balasubramanyam, P., Carman, D.: A general cooperative intrusion detection architecture for MANETs. In: Proceedings of the Third IEEE International Workshop on Information Assurance (2005)

2. Kaur, J., Lindskog, D., Zavarsky, P.: An algorithm to facilitate intrusion response in mobile ad hoc networks. In: Proceedings of the 9th International Conference on Security of Information and Networks (2016)
3. Gupta, P.: A literature survey of MANET. Int. Res. J. Eng. Technol. (IRJET) **03**(02), 95–99 (2016)
4. Hicham, Z., Ahmed, T., Rachid, L., Noureddin, I.: Evaluating and comparison of intrusion in mobile ad hoc network. Int. J. Distrib. Parallel Syst. (IJDPS) **3**(2), 243–259 (2012)
5. Anantvalee, T.: A survey on intrusion detection in mobile ad hoc network. In: Wireless/Mobile Network Security, pp. 170–196 (2006)
6. Zhang, Y., Lee, W., Huang, Y.-A.: Intrusion detection techniques for mobile wireless networks. Wirel. Netw. **9**(5), 545–556 (2003)
7. Alattar, M.: Security supervision of mobile ad hoc network: a lightweight, robust and reliable intrusion detection system, Université de Franche-Comté (2013)
8. Badie, A.M., Lindskog, D., Zavarsky, P.: Responding to intrusions in mobile ad hoc networks. In: World Congress on Internet Security (WorldCIS-2013), pp. 30–34 (2013)
9. Chadli, S., Emharraf, M., Saber, M., Ziyyat, A.: Combination of hierarchical and cooperative models of an IDS for MANETs. In: Tenth International Conference on Signal-Image Technology & Internet-Based Systems (2014)
10. Clausen, T., Jacquet, P.: Optimized link state routing protocol (OLSR). October 2003. http://www.ietf.org/rfc/rfc3626.txt. Accessed 26 June 2016
11. Ishaq, Z.: Secure MANET using two head cluster in hierarchical cooperative IDS. Int. J. Comput. Appl. (0975–8887), **57**(3), 10–13 (2012)
12. Sahana, S., Saha, S., Das Gupta, S.: Weight based hierarchical clustering algorithm for mobile ad hoc networks. Procedia Eng. **38**, 1084–1093 (2012)

Vehicular Networks

Source Mobility in Vehicular Named-Data Networking: An Overview

Joao M. Duarte[1,2](✉), Torsten Braun[1], and Leandro A. Villas[2]

[1] University of Bern, Bern, Switzerland
{duarte,braun}@inf.unibe.ch
[2] IC, University of Campinas, Campinas, Brazil
leandro@ic.unicamp.br

Abstract. This work investigates the problem of content source mobility in Vehicular Named Data Networking (VNDN). We evaluate through experiments the effects of source mobility on VNDN application performance, we analyze the main approaches already proposed in the literature to address its negative impacts and propose a new solution. Our proposed solution combines the concepts of Floating Content (FC) and Home Repository (HR). FC is an infrastructure-less mechanism that relies on in-network caching and content replication to support content sharing within a specific geographic region. An HR is a fixed node that can provide requested content objects on behalf of mobile content sources. Our main goal is to propose an efficient solution for source mobility in VNDN, able to provide high application performance and eliminate the weaknesses of existing approaches such as single points of failure and high overhead in the wireless communication medium.

Keywords: Source mobility · Vehicular Ad-hoc Networks
Named-Data Networking · Vehicular Named-Data Networking

1 Introduction

Named-Data Networking (NDN) [1] is currently seen as a viable alternative to overcome some of the main challenges associated with the traditional TCP/IP protocol suite. The TCP/IP model was conceived with the focus on static network topologies, formed by a small number of powerful and reliable computers, where resource sharing was the fundamental mission of the network. Nowadays, as opposite to resource sharing, content distribution has become the core use of computer networks [2]. The global Internet content traffic is expected to reach 1.4 zettabytes per year during 2017 [3]. Besides, network nodes have evolved from fixed to mobile and more recently to highly mobile, such as in the cases of Vehicular Ad-Hoc Networks (VANETs).

The performance of the IP point-to-point and host-based model is significantly affected by inherent VANET characteristics such as highly dynamic

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 83–93, 2018.
https://doi.org/10.1007/978-3-319-74439-1_8

topologies, and short and intermittent connectivity among vehicles [4]. Furthermore, the Internet Protocol (IP) assigns network addresses (i.e. IDs) to hosts based on their topological network location. Consequently, as mobile hosts move to new physical locations, their IDs also change.

The current paradigm shift in computer networks and the high mobility of nodes bring a set of new and challenging tasks to overcome for the future Internet. At the data link layer, new wireless communication technologies such as the IEEE802.11p [5] have been developed, enabling vehicular communications. At the network and transport layers, new solutions are still being proposed as alternatives to problems that are not easy to solve under the assumptions of TCP/IP.

Considering the above stated, to efficiently support content distribution in the network layer, new network architectures have emerged. Named-Data Networking (NDN) [1], which evolved from Content-Centric Networking (CCN), is one of the most prominent among the recently proposed network architectures.

The NDN architecture presents the Interest/Data messages exchange model. Interest messages are used by content requesters to request content, and Data messages are used by content sources to deliver requested content objects. Besides, NDN maintains three types of data structures: (i) *Content Store* (CS), for caching incoming content; (ii) *Pending Interest Table* (PIT), to keep track of forwarded Interest messages; and (iii) *Forwarding Information Base* (FIB), to store outgoing interfaces to forward Interest messages.

NDN eliminates the use of node addresses and retrieves content through content names directly. NDN also does not require some of the specific TCP/IP requirements, such as maintaining neighbor lists, domain name services, network addresses, connection-oriented sessions, etc., which generate significant overhead in dynamic wireless scenarios, negatively impacting application performance.

In NDN, content objects are replicated and cached within the network, and content requesters can obtain the requested content from the closest available content source. The closest available content source can be either the original content producer or a node caching a valid copy of the requested content. Therefore, NDN reduces content delivery delay. In this way, also the number of content sources increase within the network, contributing to increasing content delivery probability and making the content available to requesters even after the original producer has disconnected from the network.

Despite its significant advantages, deploying NDN over VANETs (i.e. Vehicular Named-Data Networking (VNDN)) presents own challenges that shall be addressed to support content distribution efficiently. Among these challenges we can mention mobility factors such as source and receiver mobility, and low vehicle densities as well as wireless communication problems such as broadcast storms and message redundancy.

The remainder of this paper describes in Sect. 2 existing solutions for mobility support in VNDN. Section 3 describes the source mobility problem, which is in the main scope of this work, and analyzes the advantages and drawbacks of already proposed solutions. In Sect. 5 we sketch a possible solution to address the drawbacks of existing solutions and efficiently mitigate the effects of source mobility in VNDN. Section 6 concludes this work.

2 Mobility Support in Vehicular Named-Data Networking

Due to Receiver mobility often partitions in communication links between Data message forwarders (i.e. reverse path partitionings (RPPs)) occur, preventing content objects from being delivered to content requesters. This problem can be addressed applying the Auxiliary Forwarding Set (AFS) mechanism [6]. AFS takes various mobility factors as inputs, including average and limit road speeds, maximum expected content delivery delay, the maximum transmission range of vehicles and distances between consecutive Interest message forwarders, to determine RPP occurrence probability. When the probability of RPP is high AFS chooses a set of eligible vehicles to forward Data messages in addition to the original forwarders. AFS increases content delivery probability and significantly improves VNDN application performance.

The problems caused by low vehicle densities can be mitigated as shown in [7] by applying the concepts of NDN agent delegation and NDN store-carry-forward (NDN-SCF). The agent delegation approach takes advantage of existing Road Side Units (i.e. RSU agents) and allows content requesters to delegate content requests to close RSU agents. In this way, when the density of intermediate vehicles to forward Interest/Data messages between content requesters and content sources is not enough, which increases the number of unsatisfied content requests, RSU agents take the role of forwarding received messages. When infrastructure support is not available, VNDN-SCF can be applied. In VNDN-SCF whenever intermediate vehicles are not able to deliver messages to subsequent vehicles, messages are buffered and periodically re-forwarded until successful delivery is achieved.

The broadcast storm problem generates congestions in the wireless communication channel. The message redundancy problem cause erroneous stop of message propagation. Both problems lead to large decrease in content request satisfaction ratios and can be addressed in different ways. For instance, [6] presents a multi-hop, receiver/based, beacon-less, and distance-based VNDN routing protocol that mitigates both problems simultaneously.

Source mobility is another challenging problem that highly impacts VNDN application performance. Nevertheless, efficient solutions for the source mobility problem are still needed.

3 Source Mobility in Vehicular Named-Data Networking

In this section we describe the source mobility problem, we evaluate its effects on VNDN through simulations, we analyze the effectiveness of different solutions that have been proposed in the literature, and finally describe a new solution that intends to solve the weaknesses of existing approaches.

3.1 Source Mobility Overview

To understand the problem of source mobility in VNDN, let us consider the following example. A vehicle *A* advertises a content object *C1* as available in its current location. Vehicles *B, C*, and *D* receive the content advertisement and save this information. After a time interval *T1*, vehicle *B* decides to request the content object *C1*. To do so, vehicle *B* sends an Interest message towards the location, where vehicle *A* advertised *C1*. However, when the Interest message arrives at the destination, vehicle *A* might already have moved to a new location. Therefore, vehicle *A* is not able to receive the Interest message, and the content request is not satisfied.

The VNDN decentralized in-network caching property provides an alternative for the source mobility problem in cases of popular content objects. In such a case the content request might be satisfied by other neighbor vehicles that have a copy of the requested content object in their Content Stores (CSs). However, in the case of less popular content, which might be requested by a single vehicle, the probability of finding the requested content in the CSs of neighbor vehicles is low.

In the following Sub-section we analyze the effects of Source Mobility in VNDN application performance.

3.2 Impact of Source Mobility on the Performance of Vehicular Named-Data Networking

To evaluate the effects of the source mobility problem in VNDN applications we performed a set of simulations. For this, we evaluated the VNDN approach proposed in [6] in the Omnet++ simulator [8]. We used SUMO [9] and VEINS [10] for road traffic and inter-vehicular communications respectively. We applied the Erlagen SUMO traces [10] and simulate a flow of four hundred VNDN enabled vehicles driving along a 10 km portion of the E45 Route in the city of Erlangen, Germany. The E45 Route is a two-way highway with four lanes.

We evaluated three different cases according to the maximum vehicle speeds. In the first case, vehicles drive with speeds between 0 and 20 km/h. In the second and third cases, we increase the maximum vehicle speeds to 50 km/h and 100 km/h respectively. In each case, we also vary the density of vehicles. We analyze the results regarding Interest Satisfaction Ratio (i.e. percentage of content objects received with respect to content objects requested) and Content Delivery Delay. The results are shown in Figs. 1 and 2.

In Fig. 1, we can observe that for the case of low mobility (20 km/h), high Interest Satisfaction Ratio (ISR) is achieved and that ISR decreases as the mobility increases. To understand why this happens, let us consider the case of two different vehicles *VA* and *VB*, advertising two different content objects, *CA* and *CB*, from the same place, at the center of a region *R*. For simplicity, let us assume that the region *R* has a circular shape and a radius $r = 200$ m. Both vehicles are following the same trajectory. However, *VA* is driving at 20 km/h (i.e. 5.6 m/s) whereas *VB* is driving at 100 km/h (27.8 m/s). Due to higher speed, *VB* only stays within the region *R* around 7.2 s after advertising the content object while

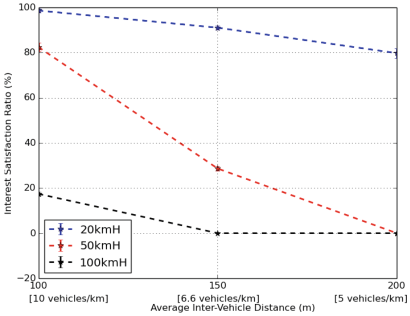


Fig. 1. Interest satisfaction ratio

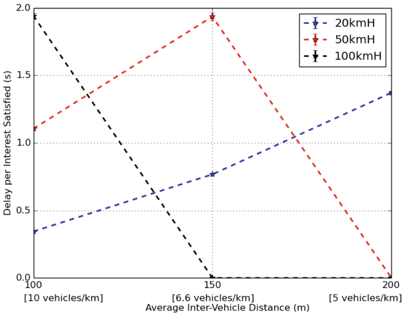


Fig. 2. Content delivery delay

VA stays within the region *R* around 35.7 s. Considering this, it is obvious that Interest messages sent by other vehicles towards the region *R* reach with higher probability *VA* than *VB*. Therefore, the Interest Satisfaction Ratio for content objects requested from *VA* is higher compared to *VB*. Furthermore, the effects of source mobility are exacerbated by the decrease in the density of vehicles.

In Fig. 2, we can observe that the Content Delivery Delay also increases as the mobility increases. We can also observe in Fig. 2 that in the case of vehicles driving with maximum speeds of 100 km/h and 50 km/h, Content Delivery Delay drops to zero when average inter-vehicle distances reach 150 m and 200 m, respectively. This happens due to complete disruptions on the communication links between vehicles that prevent content objects from being delivered to requester vehicles, as shown in Fig. 1.

Considering the above stated, solutions to efficiently address the source mobility problem in VNDN are required.

4 Existing Solutions for Source Mobility in Vehicular Named-Data Networking

In this Section we survey various approaches that have been proposed in the literature as solutions for the source mobility problem. For each solution we analyze its feasibility for VNDN scenarios.

In [11], the authors propose a proxy-based mobility support approach for mobile NDNs (PMNDN). They divide the entire network into multiple autonomous systems (ASs) and for each AS they deploy two new static functional entities. These new functional entities are called NDN access router (NAR) and proxy, respectively. NARs are used for tracking the mobility status of content sources and initiating mobility related signaling to the proxies. Proxies are used for maintaining reachability information about content sources. The authors also add two new types of data structures in NARs and proxies: the Source Location Table (SLT) and the Interest Packet Store (IPS). SLTs keep track of the content sources that are currently or previously resided in the management domain of the NAR (or the Proxy), while IPSs cache Interest messages that the NAR (or the Proxy) receives during the disconnection period of content sources.

The core idea is that content sources only exchange signaling information with the proxies. Therefore, it avoids the overhead for tracking positions of content sources by other nodes. This intends to save the resources of the wireless communication medium as well as saving the amount of energy consumed by content sources. When a content source is not reachable due to mobility, the proxy caches Interest messages forwarded towards the old location of the content source, and when it reaches another AS and connects to a new proxy, the Interest messages are forwarded by the previous proxy to the new proxy and delivered to the content source. The content source then responds with the corresponding Data messages. Therefore, the information stored in the proxies, allows NDN nodes to recover communication links that are disrupted due to content source mobility.

The solution proposed in [11] targets NDN scenarios with low mobility. In VNDN scenarios with high mobility, both content sources and content requesters join and leave ASs frequently. In dense scenarios particularly, where large number of content sources might coexist in the same AS, this approach might generate significant overhead due to signaling between content sources and proxies simultaneously with content requests and delivery. Furthermore, [11] requires significant modifications to the plain NDN structure to include the proxy and the NAR entities as well as the new data structures. This raises compatibility concerns regarding the deployment in real world scenarios.

The work in [12] proposes the idea of Indirection Points (IP) to handle source mobility in NDN. An IP is a static node that is connected to a particular Internet Service Provider (ISP). This approach uses two different types of content names. A persistent name identifies a specific content object permanently, whereas a temporary name changes as the attachment points of the content source to the network also changes. Temporary names include a prefix under which the content source can temporarily receive Interest messages. Each IP maintains a new table, called Binding Table, that relates temporal content names with persistent content names. When an IP receives an Interest message with a specific persistent name, it first tries to satisfy it from its cache. If the content is not available in its cache, the IP performs longest-prefix matching on its Binding Table for the persistent name of the requested content object. If it finds a match, it encapsulates the Interest message with the current temporary name of the content source and forwards it. When a mobile content source receives the Interest message, it decapsulates the original Interest message and sends back the corresponding Data message. After receiving the Data message, the IP decapsulates the Data Message and sends it to the initial content requester. When a mobile content source connects to the NDN network the first time, it sends a binding request to the IP specifying the content objects that it can serve, as well as the prefix under which it currently can receive Interest messages. As mobile content sources move and change their attachment points, they request IPs to delete obsolete bindings. On their sides, IPs periodically check the reachability of temporary names. If a prefix is unreachable, it is removed from the Binding Table.

This approach generates less overhead compared to [11]. Apart from the content requests and content delivery, only one extra message is exchanged within each content source and the IP. However, content requests received by an IP for content objects served by a vehicle that has already disconnected from that IP are not satisfied. Furthermore, all content requests and delivery in a given region are performed through the IP, which might generate congestions in IPs in dense scenarios. This is critical as IPs represent single points of failure in this approach. Besides, only vehicle-to-infrastructure communication is used, ignoring the potentials of vehicle-to-vehicle communications.

In [13], the authors extend the idea proposed in [12]. This approach relies on a mobility agent called Home Repository (HR), which is similar to the IP presented in [12]. The HR is a static node that can receive Interest messages on behalf of the mobile nodes. When a mobile content source moves into a new domain, the corresponding HP assigns it a prefix related to its current location, which it can use to receive Interest messages. When a content requester sends an Interest message, the Interest message is routed towards the corresponding HR. In response, the HR sends a new Interest message that is identical to the received one, except that it specifies the prefix of the corresponding content source. The content source then responds with a Data message, which includes the requested content object as well as its location prefix. The Data message propagates back to the content requester via the HR. When the content requester receives the Data message, it extracts the location prefix of the content source. With this information, next content objects can be retrieved directly from the content source without using the HR.

This work in [13] solves some of the problems of [12]. Only the first segment of a content object is retrieved directly from the HR since after knowing the prefix of the content source, content requesters send subsequent Interest messages directly to the content source. With this, the single point of failure problem and congestion on the HR are avoided. However, the problem of retrieving content objects produced by vehicles that have already disconnected from the HR still occurs. In scenarios with high mobility, where content sources might disconnect from an HR and connect to another one frequently, this can lead to considerable degradation of application performance.

The work in [14] proposes a greedy routing protocol (MobiCCN), which works in parallel with the CCN routing protocol. The goal is to support source mobility in CCN. To distinguish between the two routing protocols (i.e. MobiCCN and CCN), two different prefixes are used. Whenever a node receives a message with the prefix *greedy* the MobiCCN protocol is used whereas when the message prefix is *ccnx*, the CCN routing protocol is used. MobiCCN differentiates routers from other nodes. Each router is assigned a virtual coordinate (VC), which is embedded into message content names, while the other nodes are identified by an ID. Each router maintains a table of neighbor VCs. To forward a message, a router first extracts the destination VC from the message, then it calculates the distance between the destination and each of its neighbors. The message is forwarded to the neighbor that is closer to the destination. The router then updates its FIB, so next messages to the same node are forwarded without having to calculate the distance again. In this approach, each node also has a dedicated router that is the one closest to it and acts as its host router in the network. The host router serves as a rendez-vous point and forwards traffic to the node. Whenever a content source moves to a new attachment point, it sends an update message to its host router. Each router that receives the update message updates the corresponding entries for that content source in its FIB. Then Interest messages towards the content source can be forwarded to the new domain. This work in [14] differentiates consumer nodes from routers. However, in VANETs any vehicle might perform either as a router when forwarding messages, or as a content requester/source. Furthermore, all communications in a given region are performed through the VC. The V2V communication approach is not considered. For instance, a vehicle requesting a content object that could be satisfied by another vehicle in its one-hop neighborhood has to request the content object from the VC instead of retrieving the content object from its neighbor. This increases the delay and also creates single points of failure in VCs.

In [15], the authors present the idea of Data Spots (DS). In a DS, every requested content object is associated with a geographical region and can be generated on demand by any vehicle currently in the region. For instance, in road traffic applications, a vehicle might send an Interest message requesting information about the road conditions at a given point ahead on its trajectory. The Interest message might be geographically routed towards the specified location and delivered to the vehicles currently at that location. These vehicles might then collect the requested information and send it back towards the content requester.

This approach targets the specific case where the requested content objects are produced when requested. However, for the cases where vehicles are requesting a content object that was previously advertised by a specific content source this approach is not useful. Furthermore, it assumes that there will always be vehicles in the region where content objects are requested to collect requested content objects. With the dynamics of vehicle movement in VANETs, especially when vehicle density is low, the case of no vehicle available in the region where the content object is requested when the Interest message arrives there, might often occur, leading to unsatisfied content requests.

5 Efficiently Addressing Source Mobility in Vehicular Named-Data Networking

In this Section, we describe a possible approach to addressing the problem of source mobility in VNDN. Our main goal is to simultaneously provide high application performance and avoid the weaknesses of the solutions presented in the previous Section. Particularly we aim to provide a general purpose mechanism for efficient content delivery in VNDN and at the same time avoid single points of failure and signaling overhead in the wireless communication medium. We propose a hybrid mechanism that combines the concepts of Floating Content [16]

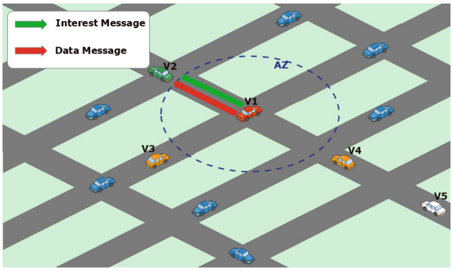


Fig. 3. Floating content

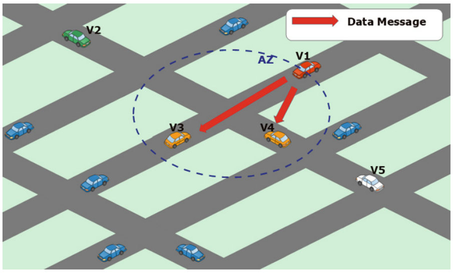


Fig. 4. Content replication 2

and the Home Repository [13] as discussed in the previous Section to provide seamless communication for VNDN content requests and delivery.

Floating Content (FC) [16] is a communication scheme for distributed content sharing within a certain geographic region (i.e. the Anchor Zone (AZ)), without the need for infrastructure support. In VNDN, FC might use decentralized in-network caching and content replication to make content available to vehicles within the AZ.

Whenever a vehicle possessing a content object leaves the AZ, it might replicate the content object to other vehicles currently within the AZ to maintain the content stored in the region where it was generated. In this way, after the vehicle that produced the content object disconnects from the network, the content can still be requested from the same region and provided by the other vehicles that received the content through replications from other vehicles that previously left the AZ.

As illustrated in Fig. 3, vehicle *V2* requests a content object *C1*. The Interest message is received by vehicle *V1* that possesses the content and sends the corresponding Data message. Later on, as shown in Fig. 4, when vehicle *V1* leaves the AZ, it replicates the content object to vehicles *V3* and *V4* that are within the AZ. When the same content object is requested by vehicle *V5*, as shown in Fig. 5, the content is provided by vehicle *V4*, which is the content source closer to vehicle *V5*.

In FC, a single vehicle caching a content object within the AZ might be enough to satisfy all content requests for that specific content object, from different vehicles. Therefore, to prevent extra overhead in the wireless communication channel, the number of vehicles that is required to replicate a specific content object can be tuned according to the density of vehicles within the AZ and the time interval between consecutive received content replicas.

In scenarios with extremely low vehicle densities, the case where no vehicle is available within the AZ at a given time, to receive content replicas from another vehicle leaving the AZ might happen. In such a case, the HR idea can be combined with FC to increase content availability probability within the AZ.

When applying HR (i.e. a fixed storage point), a vehicle leaving a sparse AZ might ask an existing HR in the AZ to also store the content object replicated.

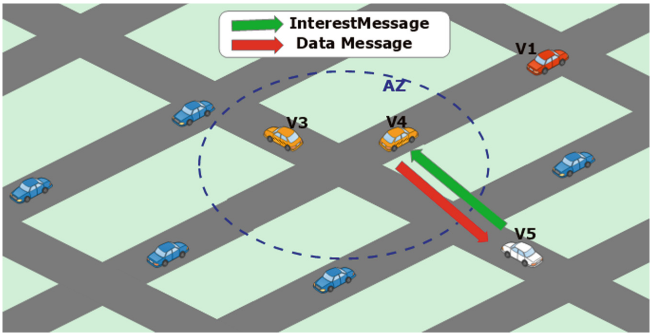


Fig. 5. Content delivery

In this way, the content object can be retrieved from the HR by other content requester vehicles. When applying this idea, content availability probability within AZ is not affected by the density of vehicles. The main reason for only storing content objects in HRs when the density of vehicles is low is to efficiently manage the storage capacity of HRs and avoid scalability issues.

6 Conclusions

In this work, we investigated the problem of content source mobility in Vehicular Named Data Networking (VNDN). We evaluated through experiments the effects of source mobility on VNDN application performance and proposed a new solution to increase content delivery probability based on the concepts of Floating Content and Home Repository. Our solution intends to eliminate the weaknesses presented by existing solutions and provide high VNDN application performance.

Acknowledgments. This work was undertaken under the CONTACT project, CORE/SWISS/15/IS/10487418, funded by the National Research Fund Luxembourg (FNR) and the Swiss National Science Foundation (SNSF) project no. 164205. The authors also would like to thank The Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior (CAPES) and the grant 2015/07538-1, Sao Paulo Research Foundation (FAPESP) for the support.

References

1. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: Networking named content. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, pp. 1–12. ACM (2009)
2. Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., Jahanian, F.: Internet inter-domain traffic. In: ACM SIGCOMM Computer Communication Review, vol. 40, pp. 75–86. ACM (2010)

3. VNI CISCO: Cisco visual networking index: Forecast and methodology, 2013–2018: Visual networking index (VNI) (2014)
4. Cunha, F., Villas, L., Boukerche, A., Maia, G., Viana, A., Mini, R.A.F., Loureiro, A.A.F.: Data communication in VANETs: protocols, applications and challenges. Ad Hoc Netw. **44**, 90–103 (2016)
5. Donato, E., Maia, G., Duarte, J.M., Loureiro, A.A.F., Madeira, E., Villas, L.: PResync: a method for preventing resynchronization in the IEEE 802.11p standard. In: 2015 IEEE Symposium on Computers and Communication (ISCC), pp. 457–462. IEEE (2015)
6. Duarte, J.M., Braun, T., Villas, L.A.: Receiver mobility in vehicular named data networking. In: ACM SIGCOMM 2017 Workshop on Mobility in the Evolving Internet Architecture (MobiArch2017). ACM (2017, accepted)
7. Duarte, J.M., Braun, T., Villas, L.A.: Addressing the effects of low vehicle densities in vehicular named-data networking. In: Submitted to MSWIM (2017). http://boris.unibe.ch
8. Varga, A., et al.: The omnet++ discrete event simulation system. In: Proceedings of the European Simulation Multiconference (ESM2001), vol. 9, p. 65. sn (2001)
9. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: Sumo-simulation of urban mobility: an overview. In: Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation. ThinkMind (2011)
10. Sommer, C., German, R., Dressler, F.: Bidirectionally coupled network and road traffic simulation for improved IVC analysis. IEEE Trans. Mob. Comput. **10**(1), 3–15 (2011)
11. Gao, D., Rao, Y., Foh, C.H., Zhang, H., Vasilakos, A.V.: PMNDN: proxy based mobility support approach in mobile NDN environment. IEEE Trans. Netw. Serv. Manage. **14**(1), 191–203 (2017)
12. Hermans, F., Ngai, E., Gunningberg, P.: Mobile sources in an information-centric network with hierarchical names: an indirection approach. In: 7th SNCNW (2011)
13. Hermans, F., Ngai, E., Gunningberg, P.: Global source mobility in the content-centric networking architecture. In: Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications, pp. 13–18. ACM (2012)
14. Wang, L., Waltari, O., Kangasharju, J.: MobiCCN: mobility support with greedy routing in content-centric networks. In: 2013 IEEE Global Communications Conference (GLOBECOM), pp. 2069–2075. IEEE (2013)
15. Zhang, Y., Afanasyev, A., Burke, J., Zhang, L.: A survey of mobility support in named data networking. In: 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 83–88. IEEE (2016)
16. Soua, R., Kalogeiton, E., Manzo, G., Duarte, J.M., Palattella, M.R., Di Maio, A., Braun, T., Engel, T., Villas, L.A., Rizzo, G.A.: SDN coordination for CCN and FC content dissemination in VANETs. In: Zhou, Y., Kunz, T. (eds.) Ad Hoc Networks. LNICST, vol. 184, pp. 221–233. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-51204-4_18

Flow-Level Simulation for Adaptive Routing Protocols in Vehicular Ad-Hoc Networks

Kais Elmurtadi Suleiman(✉) and Otman Basir

Electrical and Computer Engineering, University of Waterloo,
Waterloo, ON, Canada
{kelmurta,obasir}@uwaterloo.ca

Abstract. Adaptive routing reacts to a varying connected vehicle density by switching between different routing techniques (e.g. from Instant routing under high densities to delay-tolerant routing under low densities). These adaptations take place over large scales of time and space which makes their simulation challenging computationally. Flow-level simulators can address such challenges by using the right level of abstraction. In this paper, we present a flexible and extendable flow-level simulation environment for adaptive routing protocols in Vehicular Ad-hoc NETworks (VANETs). We discuss in details the different networking and mobility modules involved using rigorous mathematical modeling. We use MATLAB as the language of choice which allows researchers to utilize our environment while harnessing MATLAB's statistics and machine learning libraries. Such sophisticated libraries are a critical toolbox for adaptive routing protocol researchers. To the best of our knowledge, no such simulator has been publicly accessible so far.

Keywords: Flow-level · Simulation · Adaptation · Routing · VANET

1 Introduction

1.1 Adaptive Routing

Throughout different spaces and times of deployment, VANETs can be at one of three network states with a varying connected vehicles density as shown in Fig. 1. In State-1, vehicles can only communicate through the infrastructure. In State-2, vehicles can communicate through the infrastructure and other scattered vehicles. In State-3, vehicles can use both V2I and V2V communication efficiently given the small voids between vehicles. Depending on the state, different routing techniques would be suitable ranging from Multi-tier routing using V2I communication to Delay-tolerant routing, Instant routing and others using V2V communication. Adaptive routing enables switching between these different routing techniques in response to variations in the connected vehicles density.

Commonly used adaptive routing techniques include: Multi-tier routing, Instant routing, Delay-tolerant routing, Cluster-based routing, Cross-layer Optimized routing, Terminated routing, Expedited routing, Splitted routing and

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 94–105, 2018.
https://doi.org/10.1007/978-3-319-74439-1_9

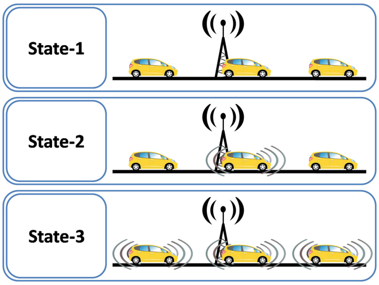


Fig. 1. VANET states

Redundant routing. **Multi-tier routing** takes place between different network tiers with different radio access technologies. **Delay-tolerant routing** overcomes big voids by storing, carrying and forwarding packets. **Instant routing** routes data instantly via optimal paths given a high connected-vehicle density. **Cluster-based routing** groups vehicle members around vehicle cluster heads. **Cross-layer optimized routing** takes into consideration information coming from other network layers. **Terminated routing** terminates packet forwarding after their Time-To-Live (TTL) period expires or after a certain number of hops is passed. **Expedited routing** forwards popular content packets before they are requested. **Splitted routing** splits packets between different paths and **Redundant routing** sends redundant packet copies using the same path or different paths.

Different adaptive routing protocols implement different sets of routing techniques and different adaptation mechanisms. A famous example is the Spray and Wait routing protocol proposed in [1]. This protocol adapts between Instant, Redundant and Delay-tolerant routing techniques. This is done by sending L message copies to L neighbors in the Spraying phase while simply transmitting messages directly to the destination in the Waiting phase if the destination has not yet been reached. Another well-known example is the Mobility-centric Data Dissemination algorithm for Vehicular networks (MDDV) proposed in [2]. This protocol adapts between Instant, Delay-tolerant and Terminated routing techniques by allowing vehicles to decide: what to send, when to send and whether to store or drop messages.

1.2 Flow-Level Simulation

As we can see, adaptive routing protocols allow for successful VANET deployment throughout different network states by switching between different routing techniques. However, this wide operational span in terms of both space and time makes their simulation challenging computationally. Flow-level simulators can address such challenges by using the right level of abstraction. Contrary to packet-level simulators, flow-level simulators deal with the data traffic as a flow inside the network. This approach makes them an efficient tool that gives a quick estimation of the network status while acknowledging the fact that it is impractical to simulate all details. Figure 2 depicts the flow-level simulation model for buffers where the current status depends on both the incoming and outgoing rates. We shall discuss this more rigorously using mathematical modeling next in the paper.

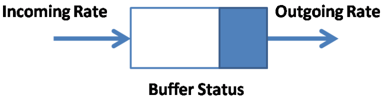


Fig. 2. Flow-level buffer model

1.3 Paper Organization

In Sect. 2, we go through some related work in order to highlight the contributions made afterwards in Sect. 3. Our simulation environment is explained in details in Sect. 4 including the simulation scenario and modules. Finally, we present in Sect. 5 some sample results in order to validate this environment.

2 Related Work

Few flow-level simulators have been proposed for high speed networks. In [3], Venkataramanan et al. propose a flow-model simulator for the internet while Yan and Gong in [4] propose a time-driven flow-level simulator for high speed networks in which traffic is treated as fluids inside the network. To the best of our knowledge, no flow-level simulator has ever been proposed for VANETs in general and for adaptive routing protocols in specific. Boban and Vinhoza in [5] focus on modeling vehicle signal obstacles while Kaisser et al. in [6] propose an enhancement for the well-known packet-level simulator "OPNET" that allows integration with the mobility simulator "SUMO". Liu et al. in [7] propose "VGSim"; an integrated mobility and networking simulation environment for VANETs also based on packet-level simulation. They claim that their simulator is the first of its kind in terms of integrating both networking and mobility aspects of VANET simulation. In both [8,9], enhancements are proposed for packet-level VANET simulators; in [8], an extension for SUMO's TafficModeller program has been presented which allows for easy traffic simulation on the network simulator "Veins" using "OpenStreetMaps". In [9], Naoumov and Gross propose an enhancement for the packet-level simulator "NS2" based on exploiting signal propagation properties. Conventional simulation platforms such as NS, OMNET++ or OPNET provide packet-level simulations which, despite the high accuracy, may struggle with a large number of nodes. Moreover, these platforms do not provide an integrated mobility and networking platform for VANET simulation. They are also not as rich as MATLAB in terms of providing efficient statistics and machine learning toolboxes which we claim to have a high potential for adaptive routing protocol research.

3 Contributions

Given the above discussion and as it is shown in the next sections, we believe that our contributions can be summarized as follows:

- Presenting an efficient and scalable flow-level simulation environment for adaptive routing protocols in VANETs. This environment is flexible and extendable in terms of adopting new scenarios and modules. It allows for fair and valid comparisons between different VANET adaptive routing protocols,
- Providing a rigorous mathematical modeling for the simulation modules,
- Covering both mobility and networking aspects of simulation which makes our simulation environment self-reliant, and
- Using MATLAB as the language of choice which allows researchers to use our environment while utilizing MATLAB's sophisticated statistics and machine learning libraries. We believe that such libraries are of high importance for adaptive routing protocol research.

4 Simulation Environment

Although other scenarios and simulator modules can easily be implemented, we restrict ourselves in this paper, due to space constraints, to the highway simulation scenario shown in Fig. 3 and the overall simulator structure shown in Fig. 4.

4.1 Scenario

Our simulation runs for a duration S_T in steps of $\triangle t$. A total of NV vehicles in the set $\mathbf{V}$ are distributed uniformly between N_L highway lanes. Each lane starts at X_{min}, ends at X_{max} and has a width of L_W. Each vehicle $V_i \in \mathbf{V}$ travels continuously while generating a constant bit rate CBR traffic with a broadcasting range of V_{BR}. The Road Side Units (RSUs) are IRD apart and located midway. The cellular access point is also located midway at (AP_X, AP_Y). Given advances in today's location services, we assume that each vehicle knows about the positions of: the access point, the RSUs and all neighboring vehicles.

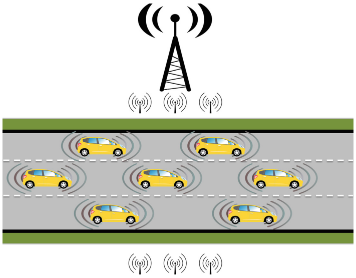


Fig. 3. Simulation scenario

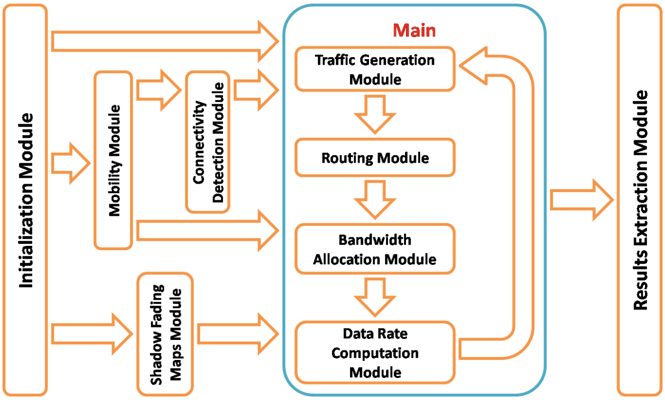

Initialization Module
Mobility Module
Connectivity Detection Module
Shadow Fading Maps Module
Main
Traffic Generation Module
Routing Module
Bandwidth Allocation Module
Data Rate Computation Module
Results Extraction Module

Fig. 4. Simulator structure

4.2 Simulator Structure

Initialization Module. Initial parameter assumptions are made here including those related to the overall simulation scenario and other simulator modules as summarized in Table 1.

Mobility Module. This modules generates vehicle mobility traces given the initial assumptions of the corresponding parameters mentioned in Table 1. Initially, $\{V_{X_1},...,V_{X_{NV}}\}$ are set uniformly randomly between X_{min} and X_{max} and $\{V_{Y_1},...,V_{Y_{NV}}\}$ fall in the middle of a lane chosen at random. Speeds $\{V_{S_1},...,V_{S_{NV}}\}$ are set uniformly randomly between V_{Smin} and V_{Smax} and accelerations $\{V_{A_1},...,V_{A_{NV}}\}$ are set uniformly randomly between 0 and $MaxVA$. Driver behavior parameters $\{V_{B_1},...,V_{B_{NV}}\}$ are set uniformly randomly between 0 and 1/2. After each $\triangle t$, V_{S_i} for all vehicles is updated as follows:

- $V_{S_i}(t+\triangle t) = min(V_{S_i}(t) + V_{A_i}, V_{Smax})$ if:
  - V_i is at least Dis_{max} length units behind the same-lane front vehicle, or
  - V_i is at least Dis_{max} length units behind a neighboring-lane front vehicle. In which case, V_i changes to this lane before accelerating.
- $V_{S_i}(t+\triangle t) = max(V_{S_i}(t) - V_{A_i}, V_{Smin})$ if:
  - V_i is at most Dis_{min} length units behind the same-lane front vehicle and less than Dis_{max} length units behind both neighboring-lane front vehicles.
- Otherwise:

$$V_{S_i}(t+\triangle t) = \begin{cases} V_{S_i}(t), & w/prob: 1 - 2 \times V_{B_i} \\ min(V_{S_i}(t) + V_{A_i}, V_{Smax}), & w/prob: V_{B_i} \\ max(V_{S_i}(t) - V_{A_i}, V_{Smin}), & w/prob: V_{B_i} \end{cases}$$

Table 1. Simulation parameters

	Item	Symbol		Item	Symbol
	Scenario parameters				
	Simulation time	S_T		Lane width	L_W
	Time step	$\triangle t$		Vehicle broadcasting range	V_{BR}
	Number of lanes	N_L		Inter-RSU distance	IRD
	Minimum X-position	X_{min}		Access point X-position	AP_X
	Maximum X-position	X_{max}		Access point Y-position	AP_Y
	Mobility Module parameters				
	Minimum vehicle speed	V_{Smin}		Minimum inter-vehicle distance	Dis_{min}
	Maximum vehicle speed	V_{Smax}		Maximum inter-vehicle distance	Dis_{max}
	Maximum vehicle acceleration	$MaxVA$			
	Shadow Fading Maps Module parameters				
Cellular	decorrelation distance	C_{Dcorr}	Vehicular	decorrelation distance	V_{Dcorr}
	shadow fading mean	SF_{CM}		shadow fading mean	SF_{VM}
	shadow fading standard deviation	SF_{CSD}		shadow fading standard deviation	SF_{VSD}
	Data Rate Computation Module parameters				
Access point	carrier frequency	AP_{Freq}	Vehicle	carrier frequency	$V2V_{Freq}$
	bandwidth	AP_{BW}		bandwidth	$V2V_{BW}$
	Physical resource block bandwidth	PRB_{BW}		transmission power level	V_P
				antenna-azimuth pattern	$V_{A(\theta)}$
	transmission power level	AP_P		antenna gain	V_{AG}
	antenna-azimuth pattern	$AP_{A(\theta)}$		antenna height	V_{AH}
	antenna gain + cable loss	AP_{AG}		cable loss	V_{CLoss}
				cable length	V_{CL}
	Noise figure	NF		Thermal noise density	TND

After updating V_{S_i}, V_{X_i} is updated as follows:
$V_{X_i}(t+\triangle t) = V_{X_i}(t) + \triangle t \times V_{S_i}(t+\triangle t)$
where: $V_{X_i}(t+\triangle t) \leftarrow V_{X_i}(t+\triangle t) - X_{max}\, if\, V_{X_i}(t+\triangle t) > X_{max}$

Connectivity Detection Module. Given the generated mobility traces, this module produces traces of all vehicle reachable neighbors including other vehicles and RSUs. These reachable neighbors set traces are needed by the Routing Module and the Bandwidth Allocation Module. Given $V_i \in \mathbf{V}$ located at (V_{X_i}, V_{Y_i}), this module constructs V_i neighbors set $\mathbf{V}_{N_i}$ by first excluding V_i and any neighboring V_j located at (V_{X_j}, V_{Y_j}) if V_j is located more than V_{BR} length units away from V_i as follows:

$$\mathbf{V}_{N_i} = \mathbf{V} \cup \mathbf{R} - \{V_i\} - \{V_j \in \mathbf{V} | ((V_{X_i} - V_{X_j})^2 + (V_{Y_i} - V_{Y_j})^2)^{\frac{1}{2}} > V_{BR}\}$$

where $\mathbf{R}$ is the set of all RSUs. After these exclusions, this module excludes neighboring V_j from $\mathbf{V}_{N_i}$ if there is another V_k obstructing the line of sight communication between V_j and V_i as follows:

$$\begin{aligned}&\mathbf{V}_{N_i} \leftarrow \mathbf{V}_{N_i} - \{V_k \in \mathbf{V}|\\ &(((V_{X_i} - V_{X_k})^2 + (V_{Y_i} - V_{Y_k})^2)^{\frac{1}{2}} < ((V_{X_i} - V_{X_j})^2 + (V_{Y_i} - V_{Y_j})^2)^{\frac{1}{2}})\\ &\wedge (|(V_{X_i} - V_{X_k})|/|(V_{Y_i} - V_{Y_k})| = |(V_{X_i} - V_{X_j})|/|(V_{Y_i} - V_{Y_j})|)\\ &\wedge sgn(V_{X_i} - V_{X_k}) = sgn(V_{X_i} - V_{X_j}) \wedge sgn(V_{Y_i} - V_{Y_k}) = sgn(V_{Y_i} - V_{Y_j})\}\end{aligned}$$

The same type of exclusions applies between V_i and any $R_w \in \mathbf{R}$.

Shadow Fading Maps Module. Using the same method adopted by [10], this module computes the correlated shadow fading map values SF_C given the initial parameter assumptions mentioned in Table 1. It starts by computing the uncorrelated values SF_U for both cellular and vehicular networks as follows:

$$10 \times log_{10}(SF_U) = SF_M + SF_{SD} \times randn$$

where SF_M and SF_{SD} are the corresponding shadow fading mean and standard deviation in dBs, respectively and "$randn$" represents a normally-distributed random number. The map sizes are given by: $[\frac{X_{max}-X_{min}}{C_{Dcorr}}] \times [\frac{AP_Y}{C_{Dcorr}}]$ for the cellular network and $[\frac{X_{max}-X_{min}}{V_{Dcorr}}] \times [\frac{AP_Y}{V_{Dcorr}}]$ for the vehicular network.

Given the distances $Xpos$ and $Ypos$ and the four SF_U values shown in Fig. 5, this module computes SF_C at (X, Y) as follows:

$$\begin{aligned}SF_C(X,Y) = (1 - \frac{Xpos}{Dcorr})^{\frac{1}{2}}(SF_{U,0}(\frac{Ypos}{Dcorr})^{\frac{1}{2}} + SF_{U,3}(1 - \frac{Ypos}{Dcorr})^{\frac{1}{2}})\\ + (\frac{Xpos}{Dcorr})^{\frac{1}{2}}(SF_{U,1}(\frac{Ypos}{Dcorr})^{\frac{1}{2}} + SF_{U,2}(1 - \frac{Ypos}{Dcorr})^{\frac{1}{2}})\end{aligned}$$

this applies for both cellular and vehicular networks where $Dcorr$ represents the corresponding decorrelation distance (i.e. C_{Dcorr} or V_{Dcorr}).

Traffic Generation Module. Given the vehicle buffers $V_{Buff_{i,i}}(t = 0) = 0\,\forall V_i \in \mathbf{V}$, this module generates the incoming flow rate represented by CBR for each vehicle V_i as follows, where $V_{Buff_{i,i}}$ is the V_i data stored at V_i buffer: $V_{Buff_{i,i}}(t + \triangle t) = V_{Buff_{i,i}}(t) + CBR \times \triangle t$.

Routing Module. Given $V_{Buff_{i,i}}$ and $\mathbf{V}_{N_i}$ of each vehicle, this module forwards data according to the implemented routing protocol.

Bandwidth Allocation Module. Given the routing decisions made, this module divides bandwidth between the contending vehicles. This is shown throughout the next Data Rate Computation Module.

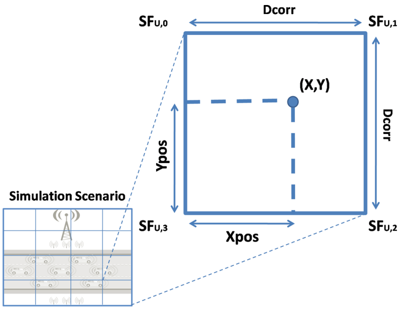


Fig. 5. Shadow fading map computation (reproduced from [4])

Data Rate Computation Module. Given the bandwidth allocations, all vehicles and access point locations, SF_C values of both cellular and vehicular networks and the initial assumptions of the corresponding parameters mentioned in Table 1, this module computes data rates and then updates vehicle buffers accordingly. For the cellular network, we adopt the OFDM access scheme, the LTE path loss model proposed by [11] and Shannon's capacity formula. For the vehicular network, we adopt the same assumptions except for using the path loss model proposed by [12] instead. Starting with the cellular network, this module allocates the bandwidth V_{BW_i} to each V_i and computes the thermal noise V_{N_i} as follows:

$$V_{BW_i} = \lfloor AP_{BW}/(NV \times PRB_{BW}) \rfloor \times PRB_{BW}$$
$$10 \times log_{10}(V_{N_i}) = TND + 10 \times log_{10}(V_{BW_i}) + NF$$

The distance $Dis_{i,AP}$ between V_i and the access point is computed followed by computing the signal path loss V_{PL_i} as follows:

$$Dis_{i,AP} = ((V_{X_i} - AP_X)^2 + (V_{Y_i} - AP_Y)^2)^{\frac{1}{2}}$$
$$V_{PL_i} = 128.1 + 37.6 \times log_{10}(Dis_{i,AP}) - 37.6 \times log_{10}(1000) + SF_C(AP_X, AP_Y)$$

The signal strength V_{S_i} is computed afterwards in order to compute the cellular network data rate V_{CDR_i} representing the outgoing flow rate and then update vehicle buffer statuses as follows:

$$10 \times log_{10}(V_{S_i}) = V_P + V_{AG} + V_{A(\theta)} + AP_{AG} - V_{CL} \times V_{CLoss} + AP_{A(\theta)} - V_{PL_i}$$
$$V_{CDR_i} = V_{BW_i} \times log_2(1 + V_{S_i}/V_{N_i})$$
$$V_{Buff_{i,i}}(t + \triangle t) = max(V_{Buff_{i,i}}(t) - V_{CDR_i}, 0)$$

For the vehicular network, the thermal noise for each V_i and the distance $Dis_{i,j}$ between V_i and V_j are computed as follows:

$$10 \times log_{10}(V_{N_i}) = TND + 10 \times log_{10}(V2V_{BW}/|\mathbf{V}_{PP_i}|) + NF$$
$$Dis_{i,j} = ((V_{X_i} - V_{X_j})^2 + (V_{Y_i} - V_{Y_j})^2)^{\frac{1}{2}}$$

where $\mathbf{V}_{PP_i}$ is the set of vehicle data paths passing through V_i. Before computing the signal path loss $V_{PL_{i,j}}$between V_i and V_j, the breaking point distance BPD is computed using the parameters $V2V_{Freq}, V_{AH}$ and V_λ as follows:

$$
\begin{aligned}
&V_\lambda = (3 \times 10^8)/V2V_{Freq}\\
&BPD = (4 \times V_{AH}^2 - V_\lambda^2/4)/V_\lambda\\
&V_{PL_{i,j}} = \begin{cases} 58.7 + 1.66 \times 10 \times log_{10}(Dis_{i,j}) + SF_C(V_{X_j}, V_{Y_j}),\\ if\, Dis_{i,j} < BPD\\ 58.7 + 1.66 \times 10 \times log_{10}(BPD/10) + SF_C(V_{X_j}, V_{Y_j})\\ +2.88 \times 10 \times log_{10}(Dis_{i,j}/BPD),\\ if\, Dis_{i,j} \geq BPD \end{cases}
\end{aligned}
$$

Using $V_{PL_{i,j}}$ and knowing $\mathbf{V}_{PP_i}$, the signal strength $V_{S_{i,j}}$ at V_j received from V_i is computed using the formula:

$$
\begin{aligned}
&10 \times log_{10}(V_{S_{i,j}}) =\\
&V_P - 10 \times log_{10}(|\mathbf{V}_{PP_i}|) - V_{PL_{i,j}} + 2 \times (V_{AG} + V_{A(\theta)} - V_{CL} \times V_{CLoss})
\end{aligned}
$$

The outgoing flow rate represented by the vehicular network data rate V_{VDR_i} is computed afterwards and before updating V_i data at both V_i and V_j buffer statuses as follows:

$$
\begin{aligned}
&V_{VDR_i} = V2V_{BW}/|\mathbf{V}_{PP_i}| \times log_2(1 + V_{S_{i,j}}/V_{N_i})\\
&V_{Buff_{j,i}}(t + \triangle t) = V_{Buff_{j,i}}(t) + min(V_{VDR_i}, V_{Buff_{i,i}}(t))\\
&V_{Buff_{i,i}}(t + \triangle t) = max(V_{Buff_{i,i}}(t) - V_{VDR_i}, 0)
\end{aligned}
$$

With an intermediate vehicle V_k in the data path, we have:

$$
\begin{aligned}
&V_{VDR_i} = min(V_{VDR_i}(E_{V_i,V_k}) + V_{VDR_i}(E_{V_k,V_j}))\\
&V_{Buff_{j,i}}(t + \triangle t) = V_{Buff_{j,i}}(t) + min(V_{VDR_i}, V_{Buff_{k,i}}(t))\\
&V_{Buff_{k,i}}(t + \triangle t) = max(V_{Buff_{k,i}}(t) - V_{VDR_i}, 0)
\end{aligned}
$$

where E_{V_i,V_k} is the data path edge between V_i and V_k.

Results Extraction Module. This module extracts final results in the form of traces and plots. These traces and plots can be accustomed to collect specific information throughout simulation.

5 Sample Results

As a proof-of-concept and in order to validate our flow-level simulation environment, we simulate a representative adaptive routing protocol and compare its performance to that of its individual routing techniques, namely: Multi-tier,

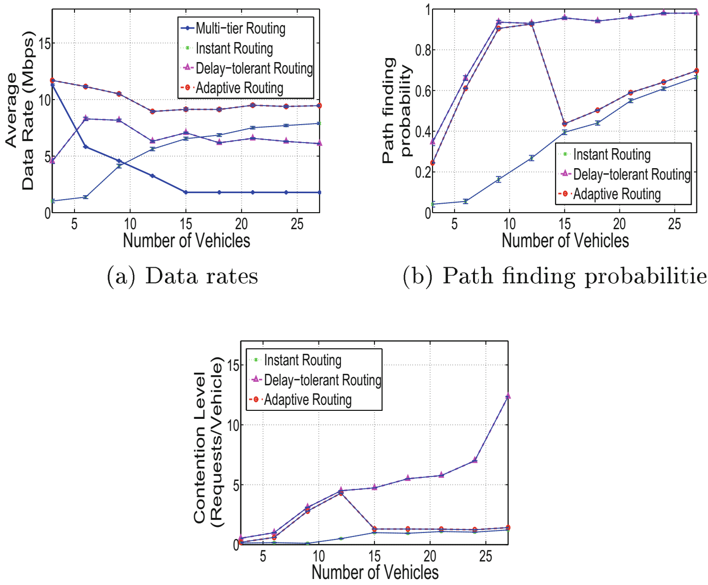
Algorithm 1. Representative adaptive routing protocol

```
while $V_i$ is ON do
  Route $V_{Buff_{i,i}}$ using *Multi-tier routing*;
  if $(V_{Buff_{i,i}} > V_{CDR_i})$ then
    if $(\mathbf{V}_{Path_i} \neq \emptyset)$ then
      Route $(V_{Buff_{i,i}} - V_{CDR_i})$ using *Instant routing*;
    else
      if $Time \leq Monitoring\,Period$ then
        Compute $P(\mathbf{V}_{Path_i} \neq \emptyset)$;
        Route $(V_{Buff_{i,i}} - V_{CDR_i})$ using *Delay-tolerant routing*;
      else
        if $P(\mathbf{V}_{Path_i} \neq \emptyset) < \alpha$ then
          Route $(V_{Buff_{i,i}} - V_{CDR_i})$ using *Delay-tolerant routing*;
        end if
      end if
    end if
  end if
  Route $(V_{Buff_{i,j}} > 0 \forall V_j)$ using *Delay-tolerant routing*;
end while
```

Instant and Delay-tolerant routing. Algorithm 1 shows the pseudocode of this protocol while leaving the technical details of its individual routing techniques for future work due to space constraints.

Using our adaptive routing protocol, each vehicle forwards its data using Multi-tier routing. If there is still data remaining $(V_{Buff_{i.i}} > V_{CDR_i})$, then the protocol routes this data using Instant routing if there is an instant path $(\mathbf{V}_{Path_i} \neq \emptyset)$ or using Delay-tolerant routing if there isn't and the probability of finding one is less than a threshold $(\alpha = \frac{1}{3})$ once the "$Monitoring\,Period$" has passed. The "$Monitoring\ Period$" is the period during which $P(\mathbf{V}_{Path_i} \neq \emptyset)$ is monitored/computed while relying on Delay-tolerant routing if: $(V_{Buff_{i.i}} > V_{CDR_i}) \wedge (\mathbf{V}_{Path_i} = \emptyset)$. This condition is implemented in order to avoid a more congested delay-tolerant path while waiting for a less congested instant path estimated to come shortly with $(P(\mathbf{V}_{Path_i} \neq \emptyset) \geq \alpha)$. In all cases, any $(V_{Buff i,j} > 0)$ is always routed using Delay-tolerant routing. Results in Fig. 6a to c validate our simulation environment by meeting our intuitions as follows:

- Multi-tier routing provides a declining data rate due to the increasing congestion occurring at the access point,
- Instant routing provides a growing data rate due to the higher probability of finding instant paths as the number of vehicles increases. The slight contention level increase is due to the growing path competition,
- Delay-tolerant routing starts with a fast growing data rate that quickly struggles under the pressure of more vehicles. This is due to the fact that the delay-tolerance exhibited allows for communications under low vehicle density while incurring the cost of a quickly rising contention level that jeopardizes eventually the data rate,



(a) Data rates

(b) Path finding probabilities

(c) Contention levels

Fig. 6. Adaptive routing performance against its individual routing techniques

- Our adaptive routing protocol utilizes initially Multi-tier routing. Then, it finds delay-tolerant paths by switching to Delay-tolerant routing. However, it switches afterwards to Instant routing as the contention level rises after exceeding 12 vehicles. After 12 vehicles, the probability of finding an instant path surpasses the threshold ($\alpha = \frac{1}{3}$) which we set as acceptable given the benefit of avoiding high contention levels under Delay-tolerant routing.

6 Conclusions and Future Work

Adaptive routing protocols allow for successful VANET deployment by switching between different routing techniques. However, their operational span makes their simulation computationally demanding. Flow-level simulators offer the right level of abstraction in order to overcome such computational challenges.

In this paper, we have presented a flow-level simulation environment for adaptive routing in VANETs with its different modules explained using rigorous mathematical modeling. These modules include both VANET networking and mobility aspects which makes our environment self-reliant. MATLAB has been chosen in order to allow researchers in the field to utilize our environment while harnessing the rich MATLAB statistics and machine learning libraries needed for adaptive routing research. In order to validate our environment, we have evaluated the performance of a representative adaptive routing protocol against its individual routing techniques. Results confirm our intuitions and show the adaptations made. In the future, we plan to extend our environment in order to simulate and develop adaptive routing techniques which utilize MATLAB's statistics and machine learning libraries. Finally, plans are underway to make our environment publicly accessible online.

References

1. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN 2005), pp. 252–259. ACM, New York (2005)
2. Wu, H., Fujimoto, R., Guensler, R., Hunter, M.: MDDV: a mobility-centric data dissemination algorithm for vehicular networks. In: Proceedings of the 1st ACM International Workshop on Vehicular Ad hoc Networks (VANET 2004), pp. 47–56. ACM, New York (2004)
3. Venkataramanan, R., Jeong, M.-W., Prabhakar, B.: A Flow-and Packet-level Model of the Internet
4. Yan, A., Gong, W.-B.: Time-driven fluid simulation for high-speed networks. IEEE Trans. Inf. Theory **45**(5), 1588–1599 (1999)
5. Boban, M., Vinhoza, T.T.V.: Modeling and simulation of vehicular networks: Towards realistic and efficient models. INTECH Open Access Publisher (2011)
6. Kaisser, F., Gransart, C., Berbineau, M.: Simulations of VANET scenarios with OPNET and SUMO. In: Vinel, A., Mehmood, R., Berbineau, M., Garcia, C.R., Huang, C.-M., Chilamkurti, N. (eds.) Nets4Cars/Nets4Trains 2012. LNCS, vol. 7266, pp. 103–112. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29667-3_9
7. Liu, B., Khorashadi, B., Du, H., Ghosal, D., Chuah, C., Zhang, M.: VGSim: An integrated networking and microscopic vehicular mobility simulation platform. IEEE Commun. Mag. **47**(5), 134–141 (2009)
8. Arellano, W., Mahgoub, I.: TrafficModeler extensions: A case for rapid VANET simulation using, OMNET++, SUMO, and VEINS. In: High Capacity Optical Networks and Emerging/Enabling Technologies. Magosa 2013, pp. 09–115 (2013)
9. Naoumov, V., Gross, T.: Simulation of large ad hoc networks. In: Proceedings of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2003), pp. 50–57. ACM, New York (2003)
10. IEEE 802.16: EEE 802.16m Evaluation Methodology Document (EMD). IEEE 802.16 Broadband Wireless Access Working Group, January 2009
11. Astely, D., Dahlman, E., Furuskar, A., Jading, Y., Lindstrom, M., Parkvall, S.: LTE: the evolution of mobile broadband. IEEE Commun. Mag. **47**(4), 44–51 (2009)
12. Abbas, T., Kåredal, J., Tufvesson, F.: Shadow fading model for vehicle-to-vehicle network simulators. In: COST IC1004 5th Management Committee and Scientific Meeting, COST IC1004 (2012)

Control Overhead Reduction in Cluster-Based VANET Routing Protocol

Ahmad Abuashour(✉) and Michel Kadoch

Department of Electrical Engineering, École de Technologie Suprieure,
1100 Notre-Dame Street West, Montreal, QC H3C 1K3, Canada
ahmad.abuashour.1@ens.etsmtl.ca, michel.kadoch@etsmtl.ca

Abstract. Vehicular Ad-Hoc NETworks (VANETs) are unique form of Mobile Ad-Hoc NETworks (MANETs), where the nodes act as vehicles moving with relatively high mobility, and moving in a predefined routes. The mobility in VANETs causes high topology changes and in turn leads to excessive control overhead and frequent link communication failures. Traditionally, clustering techniques have been used as the main solution to reduce the control overhead messages in VANET, in which the network is divided into multiple clusters and selecting one of the Cluster Members (CMs) as a Cluster Head (CH). Still, a problem occurs when the control overhead messages increase due to periodically forwarding of CM HELLO (CMHELLO) messages between the CMs and the CH, and when the CH periodically broadcasts an CH advertisement (CHADS) messages to declare itself to the CMs. In this paper, we propose a Control Overhead Reduction Algorithm (CORA) which aims to reduce the control overhead messages in a clustered topology. Therefore, we develop a new mechanism for calculating the optimal period for updating or forwarding the CMHELLO messages between the CMs and the CH. Finally, we evaluate the performance of our proposed work by comparing with other recent researches that published in this field. Based on the simulation results, the CORA algorithm significantly reduces the CMHELLO messages, where it generates the minimum percentage of CMHELLO messages compared with other techniques proposed on this field.

Keywords: CH · CM · CMHELLO · CHADS · VANET
MANET · CORA

1 Introduction

Vehicular Ad Hoc NETwork (VANET) is a derived form of self-organized Mobile Ad Hoc NETwork (MANET). In VANET, vehicles are equipped with an On-Board Units (OBUs) that can communicate with each other (V2V communications), or/and with stationary road infrastructure units (V2I) that are installed along the roads. VANETs have several characteristics that makes it different from MANETs; such as high node mobility, predictable and restricted mobility patterns, rapid network topology change, and long battery life.

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 106–115, 2018.
https://doi.org/10.1007/978-3-319-74439-1_10

The Cluster-Based Routing (CBR) protocol combines the features of both proactive [1] and reactive [2] routing protocol. The nodes in the clustered network are grouped together in a particular area called clusters. CBR protocols are widely used to improve the scalability of VANET environment and to reduce the control overhead message. Although the clustering techniques are minimizing the routing control overhead, clustering management and frequent CH elections increase the clustered control overhead. The clustered control overhead messages are produced by forwarding or broadcasting of control messages between the CMs and the CH, and these massages are classified into CMHELLO messages and CHADS messages, respectively. When the generated control overhead messages are increasing in a cluster topology, then the available bandwidth resources are decreasing. The main objectives of this paper is minimizing the number of generated clustered control overhead messages in a cluster-based VANET topology. Therefore, we propose a Control Overhead Reduction Algorithm (CORA) that optimizes the updating or forwarding period of CMHELLO messages in a clustered-based topology.

This paper is outlined as follows; in Sect. 2, we present a literature review that related to control overhead reduction techniques in clustered topology. Section 3 presents the CORA algorithm in a clustered highway scenario. Section 4 shows the simulation, results and analysis. Finally, Sect. 5 concludes this article, respectively.

2 State of Arts

Control overhead reduction techniques are an important and interesting subject in many of recent researches. The main objective of minimizing the control overhead messages is improving the network efficiency by producing more bandwidth resources for data transmission.

VANETs are an autonomous systems formed by connected vehicles without the need for any infrastructure. Routing in VANET is a significant challenge due to the nature of fast topology changes. The high mobility in VANET forces the vehicles to periodically exchanging of control overhead message. Therefore, the excessive amount of control overhead messages yield to consume high amount of available bandwidth resources.

The main solution to reduce the control overhead messages is to use the clustering technique, the concept of clustering means to transform the big network into small grouped networks called clusters. In each cluster, one of cluster members (CMs) should be elected to be responsible for all local cluster communication, and its called Cluster Head (CH). This process will significantly reduce the control overhead because restricts the communication between each CM and CH instead of exchanging the control overhead messages between all the CMs in the cluster. Many researches proposed several algorithms of selecting the CH in each cluster based on specific parameters, such as: vehicle ID, vehicle location, vehicle speed, vehicle direction, and vehicle LT. The process of electing CH is out of scope in this paper. In general, dividing the network into multiple clusters reduces the communication overhead and improves the network efficiency.

In the cluster, CMs and CH should periodically exchange the control overhead messages. The CMHELLO message is one of important control overhead messages that used to define the vehicle identity and location in VANET network. The number of control overhead messages in the cluster is in proportion to the number of CMs. Many techniques are proposed in the literature to reduce the number of CMHELLO messages as follows:

Tao et al. [3] proposed a Cluster-Based Directional Routing Protocol (CBDRP) for highway scenario. The CMs exchange the control overhead packets that contain the location of the cluster, location of vehicle, and the velocity of the vehicle. A CH distributed algorithm is used to select one CH among CMs, the selected CH has full information about its CMs. CBDRP concentrates to reduce the routing overhead packet from source to destination, without considering the control overhead packets that produced by the CMs in each cluster.

Pedro et al. [4] proposed a Beacon-less Routing Algorithm for Vehicular Environment (BRAVE), the proposed protocol objects to reduce the control overhead messages in a broadcast approaches. In BRAVE, the next forwarder vehicle is reactively selected among those neighbors that have successfully received the messages. The drawback of BRAVE protocol that each vehicle participates in the routing protocol still required to exchange a beacon message among them. In the simulation setting, BRAVE sets the exchanging time of the beacon message to 2 s to keep monitoring the vehicles location. In general, reactive routing protocol reduce the control overhead messages compared with proactive routing protocol, however it still suffering of high control overhead compared with CBR protocols.

Dan et al. [5] proposed a MOving-ZOne-based (MoZo) architecture, MoZo consist of multiple moving zones that group vehicles based on the movement similarity. The selected CH is responsible for managing information about CMs as well as the forwarding packets. The control overhead updating period for the CMs in MoZo architecture is varied between the changing of moving function of 5 m/s or 4 s. In [6], a periodically live message is broadcasted by every node for announcement of it's existence in the cluster. Also, this paper does not consider the live message size and the period of updating these messages in its evaluation.

In the literature, the authors do not provide any guidelines to exploit the cluster resources. Though the main properties of any clustering algorithm are high CMHELLO message and CHADS messages. In addition, most of the literature ignoring the control overhead message size. To the best of our knowledge there are no researches that investigated to reduce the control overhead message by optimizing the control overhead exchanging period time for the CMs and CH. Therefore, in this paper we propose a Control Overhead Reduction Algorithm (CORA) that optimizes the updating period of CMHELLO messages in each cluster.

3 Control Overhead Reduction Algorithm

In VANET, the CBR protocols do not require that every vehicle knows the entire topology information. Only the selected CH vehicles require to know the topology information and other CMs only require to periodically exchange their information with the CH via CMHELLO messages. CMHELLO message is one kind of the control overhead messages that we discuss in this paper. The CMHELLO messages inform the CH about CM identity and it could combine other parameters; such as current location, direction, velocity, and life time. The increasing size of CMHELLO messages consider an important issue that degrade the performance of any mobile and limited resources networks. Furthermore, the frequently exchanging of CMHELLO message negatively impact the network performance. Therefore, in this section we propose a new algorithm that reduces the number of control overhead messages, which called CORA algorithm. CORA is based on the assumption that each vehicle in the VANET environment can know its current location and cluster ID by using a digital map and Global Positioning System (GPS). Also we used Cluster-Based Life-Time Routing (CBLTR) protocol [7], which outperforms many other cluster protocols in terms of increasing the average throughput and stability in clustered network. The CBLTR [7] protocol selects the CH based on maximum LT among the CMs and the CH maintains it's status as CH until arrives to a predefined threshold point. Therefore, the CBLTR protocol significantly reduced the CH election processes.

In general, each vehicle must be defined as CM or CH at any time. Algorithm 1 explains the CORA algorithm as follows; initially, each vehicle enters any cluster coordination zone sets its status as CM by default (lines 2 and 3). Then, it waits for τ second (line 4), if it does not receive any message, it changes its status to CH and starts periodically (every τ second) forwarding CHADS message (lines 10 and 11), this message consists of CH identification information and the remaining LT that the CH predict to spend in the cluster zone. Otherwise, it stays as CM and replies with only one CMHELLO message which consists of the CM identification and the remaining LT that the CM predict to spend in the cluster zone (lines 5–8). The remaining LT is varied among vehicle due to the velocity variation. The objective of periodically exchanging CHADS message is to inform newly-arrived CMs that an active CH exist. When the CH receives all replies from the CMs within its

Algorithm 1. CORA PROTOCOL

```
 1: for $t = 1$ $to\ end\ of\ simulation\ time$ do
 2:     if $any\ vehicle\ enters\ the\ cluster\ Zone$ then
 3:         $vehiclestaus = CM$
 4:         $wait\ \tau\ sec$
 5:         if $CM\ receives\ CH\ ads\ message$ then
 6:             $reply\ to\ CH\ by\ CMHELLO\ message$
 7:             $Containes < CMID, CMLT >$
 8:         else
 9:             $vehicle\ staus = CH$
10:             $every\ \tau\ sec\ send\ CH\ ads\ message$
11:         end if
12:     end if
13: end for
```

associated LT, the CH is capable to calculate the candidate CH (CCH) before leaving the cluster. Therefore, the CMs do not require to periodically update their information with the CH while the CHLT is not expired. In other word, the CMHELLO messages that produced by the CMs are proportional to the number of CH changes instead of specific period of time. Thus, that yields to significantly minimize the control overhead messages in each cluster.

To calculate the number of CHADS message within the simulation time, first we divide the elected CH remaining LT time by the period of exchanging time τ (τ is a constant value), as in Eq. 1:

$$AdsCH_{ijk} = \frac{CHLT_{ijk}}{\tau} \tag{1}$$

where:

$AdsCH_{ijk}$: Total number of CHADS messages produced from CH with ID i in cluster j in segment ID k.

$CHLT_{ijk}$: The remaining LT for CH with ID i in cluster j in segment ID k.

τ: The periodic exchanging time for CHADS message. Next, we calculate the overall CHADS messages for all elected CHs in the same cluster within the simulation time, as in Eq. 2:

$$TotalAdsclus_{jk} = \sum_{i=1}^{x} AdsCH_{ijk}, 0 < TotalAdsclus_{jk} < simulationtime \tag{2}$$

where:

$TotalAdsclus_{jk}$: The number of CHADS message produced from CHs in cluster ID j in segment ID k.

To calculate the total CHADS messages that generated in a segment with multi-cluster, we do the summation for the number of CHADS messages for each cluster, as follow:

$$TotalCHAds_k = \sum_{j=1}^{y} TotalAdsclus_{jk} = \sum_{j=1}^{y} \sum_{i=1}^{x} \frac{CHLT_{ijk}}{\tau} \tag{3}$$

where:

$TotalCHAds_k$: The number of CHADS message produced by CHs in segment ID k.

y: Total number of clusters within the segment.

Since τ is constant value, then the number of CHADS messages that produced by the CH are proportional to the CHLT value in each cluster.

In Fig. 1, the CH forwards CHADS messages every τ seconds to all of its CMs until its LT expires. Each selected CH should periodically forward an CHADS messages to announce itself in the cluster zone. The vehicles A, B, C, and D are CMs that receive CHADS from the CH while its LT time does not expire.

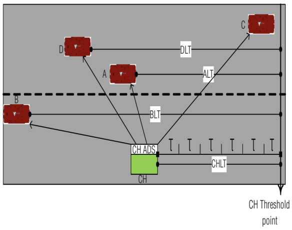


Fig. 1. CHADS message

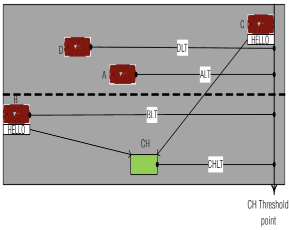


Fig. 2. CMHELLO enters and leaves the cluster

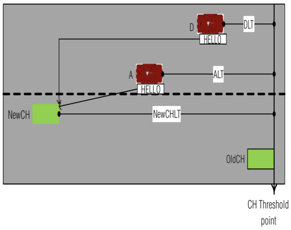


Fig. 3. CMHELLO when new CH selected

On the other hand, when any vehicle enters the cluster zone, its default status is CM. It should exchange the CMHELLO message with the CH. So, in this paper the main contribution is to minimize the number of CMHELLO messages by taking into consideration CHLT. When any vehicle enters the cluster zone, it sends a CMHELLO message to the CH (if it receives the CHADS after τ second). In this case we have two scenarios; if the CMLT is greater than CHLT, then the number of CMHELLO message equals to the number of CH changes within the CMLT plus two (the mandatory two CMHELLO messages when the CM enters the cluster and before leaves the cluster), otherwise; the CM generates the CMHELLO message only two times; that is when it enters the cluster and before leaves the cluster. Figure 2 explains a scenario of CM CMHELLO message; first, when vehicles enter the cluster zone (as vehicle B), then it should send CMHELLO message, and when the vehicle leaves the cluster zone, then it sends another CMHELLO message (as vehicle C), whereas the vehicles (vehicle A and D) that already in the cluster zone and within the CHLT do not require to send any CMHELLO message. Figure 3 explains another scenario when the CH (Old CH) arrives to the threshold point (the point that the current CH should select another CH), the old CH sends an CHADS message informing the CMs for the new CH, in the meantime; all the CMs (vehicle A and D) should send the CMHELLO message to the new CH.

The following Equation describes mathematically the two scenarios in Figs. 2 and 3:

$$NumCM_{ijk} = \begin{cases} numCH_{ijk} + 2, & \text{if } CMLT_{ijk} > CHLT_{jk} \\ 2, & \text{if } CMLT_{ijk} \leq CHLT_{jk} \end{cases} \tag{4}$$

where:

$NumCM_{ijk}$: The number of CMHELLO message produced by CM with ID i in cluster ID j in segment ID k.

$numCH_{ijk}$: The number of CH changes within $CMLT_{ijk}$.

$CMLT_{ijk}$: The remaining LT for CM i in cluster ID j in segment k.

We can mathematically formulate the total of CMHELLO messages for a specific cluster by following expression:

$$TotalHELLO_k = \sum_{j=1}^{y} NumCM_{ijk} \tag{5}$$

where:

$TotalHELLO_k$: The total number of CMHELLO messages produced from CMs in cluster k.

y: Total number of CM in the cluster ID k.

Also, we can mathematically formulate the total of CMHELLO messages for specific pre-divided cluster segment as the following Equation:

$$TotalCMHELLO_m = \sum_{j=1}^{p} TotalHELLO_j \tag{6}$$

where:

$TotalCMHELLO_m$: The total number of CMHELLO message produced from CMs in segment ID m.

p: Total number of clusters in the segment ID m.

Finally, the total control overhead messages within the simulation time equal the summation of CMHELLO messages that produced from the CMs and the periodically CHADS messages that produced by the CHs. As the following Equation:

$$TotalAdsmessage_k = TotalCMHELLO_k + TotalCHAds_k \tag{7}$$

4 Simulation, Results, and Analysis

By using the SUMO version 0.28.0 traffic generator and Matlab version R2016b, we implement and evaluate our proposed protocol. In Table 1, we present the simulation parameter we used to evaluate the performance of our proposed work.

We first implemented a bidirectional highway scenario with length 10000 m, then we divided the highway to fixed sizes of clusters of length 250 m each. The vehicles enters the highway scenario in fixed rate which equals 1 vehicle/sec, when any vehicle arrives any end of the highway, it makes a U turn and drives back in the opposite direction. The SUMO traffic generator keeps safety distance between the vehicles, and the distance distribution between the vehicle follow an exponential distribution. All the vehicles remain in the highway until the end of the simulation. The simulation starts to gather the results after all vehicle entering the Highway scenario.

Based on Eq. 6, we calculate the number of CMHELLO messages in each cluster, we assumed here that the vehicles use the same architecture of CMHELLO message in terms of size. In Fig. 4, we compare our results with three other

Table 1. Simulation parameters

Parameter	Value
Simulation time	500 s
Topology type	Highway
Number of cluster	40
Number of vehicles in each direction	200
Vehicles arrival rate	1 vehicle/sec
Communication range	250
Vehicle range speed	(10–60) kmph
CH protocol used	CBLTR

Table 2. The mean and percentage of HELLO messages generated by the CMs

Protocol Name	Mean	Percentage of HELLO messages
CORA	215.25	2.5%
MoZo	1215.8	13.9%
BRAVE	2431.6	27.8%
CBDRP	4863.3	55.8%

protocols that mentioned in the literature; CBDRP, BRAVE, and MoZo protocols. In CBDRP protocol, the CMs in each cluster are updated vary quickly, and this yields to produce high CMHELLO messages. In BRAVE protocol, the CMHELLO interval is 2 s. In MoZo protocol, the authors assume that the vehicles need to send CMHELLO updates messages when they deviate from their defined original moving function more than 5 m/s or the time from the last update which equals to 4 s. CORA outperforms all previous protocols in terms of the number of CMHELLO messages that generated in each cluster within a period of time. The CORA protocol minimizes the number of CMHELLO messages due to avoid periodically exchanging of CMHELLO message, CORA propagate the CMHELLO messages in three scenarios; which are: when the CM enters the cluster zine, second; when the CM leave the cluster zone, and when new CH announces about itself. In general, CORA calculate the optimal number of CMHELLO messages in each cluster.

In Table 2, we present a numerical results to validate the performance of the CORA protocol. Column 2 calculates the average number of CMHELLO messages that generated within the simulation time by CORA, MoZo, BRAVE, and CBDRP protocols. Column 3 calculates the percentage number of HELLO messages that generated by the CMs, the percentage is calculated by dividing the average number of CMHELLO messages that generated by any algorithm to the overall CMHELLO messages that generated by all algorithms. The CORA algorithm significantly reduces the CMHELLO messages, where it generates the minimum percentage of CMHELLO messages, which is equal to 2.5%, and the main reason of that returns to forward the CMHELLO messages only in three scenarios that we explained in the previous section. In contrast, MoZo, BRAVE, and CBDRP algorithms, show high number of CMHELLO messages, and the reason of that because all of these protocols forward periodically the CMHELLO messages.

We evaluate the performance of CORA algorithm in terms of the total number of control overhead messages. Based in Eq. 7, the total of number of control

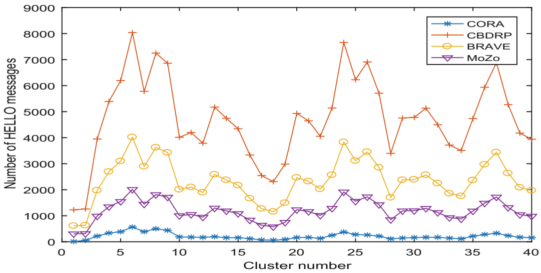
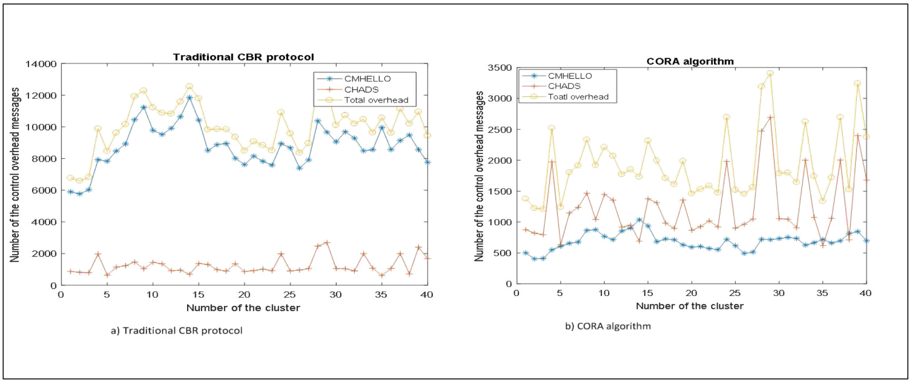


Fig. 4. Number of CMHELLO message in highway scenario



a) Traditional CBR protocol

b) CORA algorithm

Fig. 5. CORA vs traditional CBR protocol

messages are the summation of all messages that forwarded by the CMs and broadcasted by the CHs in a specific period of time. In Fig. 5, we present the total number of control overhead for the CORA algorithm and another traditional CBR protocol (such as CBDRP). As shown in Fig. 5a, in traditional CBR protocol all the vehicles in the clusters should forwards or broadcasts the control overhead messages periodically and depending mainly on time. Therefore, an excessive amount of generated control overhead messages are produced in a traditional CBR protocols. In contrast, Fig. 5b shows that CORA algorithm achieves a significant reducing of CMHELLO messages, and the reason of that because the CM only forward CMHELLO messages only in three cases; when the CMs enters or leaves the cluster zone or CH election process notification received. In other words, the CMs mainly depend on the location to forwards the CMHELLO message rather than the times.

5 Conclusion

In this paper, we propose a Control Overhead Reduction Algorithm (CORA), which aims to reduce the number of CMHELLO messages that generated by the CMs in the clusters, a new mechanism for calculating the optimal period for updating or exchanging CMHELLO messages is proposed. CORA propagate the CMHELLO messages in three scenarios: when the CM enters the cluster zone, second; when the CM leave the cluster zone, and when new CH elected. Based in the simulation results, CORA significantly minimized the number of CMHELLO message in each cluster and in any segment in general.

References

1. Spaho, E., Ikeda, M., Barolli, L., Xhafa, F., Younas, M., Takizawa, M.: Performance of OLSR and DSDV protocols in a VANET scenario: evaluation using CAVENET and NS3. In: 2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications, pp. 108–113, November 2012
2. Ding, B., Chen, Z., Wang, Y., Yu, H.: An improved AODV routing protocol for VANETS. In: 2011 International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1–5, November 2011
3. Song, T., Xia, W., Song, T., Shen, L.: A cluster-based directional routing protocol in VANET. In: 2010 IEEE 12th International Conference on Communication Technology, pp. 1172–1175, November 2010
4. Ruiz, P.M., Cabrera, V., Martinez, J.A., Ros, F.J.: Brave: Beacon-less routing algorithm for vehicular environments. In: The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010), pp. 709–714, November 2010
5. Lin, D., Kang, J., Squicciarini, A., Wu, Y., Gurung, S., Tonguz, O.: MoZo: a moving zone based routing protocol using pure V2V communication in VANETs. IEEE Trans. Mob. Comput. **16**(5), 1357–1370 (2017)
6. Singh, S., Rajpal, N., Sharma, A.: Address allocation for MANET merge and partition using cluster based routing. SpringerPlus **3**(1), 605 (2014). http://dx.doi.org/10.1186/2193-1801-3-605
7. Abuashour, A., Kadoch, M.: A cluster-based life-time routing protocol in VANET. In: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 213–219, August 2016

Ariel Networks and Routing

A Hierarchical Framework for Estimating the Performance of an Aerial Network

Kamesh Namuduri[1(✉)], Amjad Soomro[2], and Srinivasa Kiran Gottapu[1]

[1] University of North Texas, Denton, USA
kamesh.namuduri@unt.edu
[2] Air Force Research Laboratory, Rome, NY, USA

Abstract. Dynamic networks such as airborne networks are characterized by fast changing topologies. Such networks require efficient strategies for estimating performance measures towards mission-specific objectives. Performance measures defined over a network will help choose optimal routes for information sharing between a pair of nodes.

This article presents a model and approach to estimate the performance of a dynamic network. First, it introduces *goodness* measures at three levels of hierarchy - link, path, and network, in terms of primitive metrics such as reliability, throughput, and latency. Second, it presents a strategy to estimate these goodness measures. The strategy is illustrated by applying it to find an optimal path between a pair of nodes in a network. Results presented on five benchmark networks illustrate the value of the proposed model.

Keywords: Aerial network · Adhoc network · Reliability Throughput · Latency · Goodness measures · Configuration graph

1 Introduction

Often times, there is a need to find a reliable route between a pair of nodes in a network. At other times, it may be necessary to find a suitable node in a network to host a service, such as a controller in a software-defined network (SDN). Many solutions exist for finding optimal routes in wired networks as well as for wireless networks. However, finding optimal routes in extreme dynamic networks such as airborne networks is challenging and requires new strategies and solutions. Furthermore, in many practical scenarios, there may be multiple criteria for choosing an optimal route between a pair of nodes such as based on the shortest distance, reliability or security. At other times, it may be a combination of several metrics which define a suitable route for sharing information between a pair of nodes. Thus, there is a need for developing a framework and strategy for finding the best routes given performance metrics of interest in mission-critical dynamic networks.

This paper presents the results of our investigation into a hierarchical framework for evaluating *goodness* measures at various levels in a dynamic network. We define a goodness metric as a function of several primitive parameters in a

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 119–130, 2018.
https://doi.org/10.1007/978-3-319-74439-1_11

hierarchical manner. This hierarchical model allows one to answer questions such as the following: which network topology makes the network the most efficient in terms of a set of given measures? Which routes are critical? Which routes are more reliable? Which mobility patterns make the network maintain connectivity and performance? Which topology maximizes the longest surviving path between two nodes? In which topology one link failure has minimal effect on the network performance? Which nodes have maximum impact on the network performance? How can the criticality of routes and their impact be minimized? The model is independent of any specific metrics and it is suitable for many types of networks. However, in this paper, we consider an aerial network as a specific scenario where the model could be applied to illustrate its application.

The rest of this paper is organized as follows. Section 2 provides a brief survey of related literature. Section 3 presents a general description of an aerial network. Section 4 defines metrics to measure the performance of an aerial network in terms of reliability, throughput, and latency. Section 5 defines goodness measures at three levels of hierarchy - link, path, and network. An algorithm based on goodness measures for generating optimal paths from source node to all other nodes in the network is presented. The implementation of the proposed strategies on benchmark graphs is discussed in Sect. 5.2. Section 6 concludes the paper with a summary.

2 Literature Survey

There is significant literature in the area of modeling and analysis of networks including wireless networks, wireless sensor networks, ad hoc networks, and vehicular networks among others. However, in the domain of extremely dynamic networks such as airborne networks, performance modeling is mostly done through simulations and to a certain extent through experimental analysis. Our work shares some commonalities with disruption-tolerant networks, wireless sensor networks and airborne networks. Concepts gathered from the literature in these three domains are briefly discussed below.

Evaluation of network resilience, survivability, and disruption tolerance in networks is analyzed in [1,2]. The authors describe a comprehensive methodology to evaluate network resilience using a combination of topology generation, mathematical analysis, simulations and experimental evaluation techniques with the goal of improving the resilience and survivability of a network. In [3], multilevel resilience of networks is investigated in terms of redundancy for fault-tolerance, diversity for survivability and connectivity for disruption tolerance.

A shortest path tree-based algorithm for relay placement in a wireless sensor network and its performance analysis is presented in [4]. The authors investigate the problem of designing a multihop wireless network for interconnecting sensors to a base station by deploying a minimum number of relay nodes at a subset of given potential locations while meeting the hop-count constraints.

A position-aware, secure and efficient routing strategy for airborne mesh networks is investigated in [5]. Cognitive radio technology is investigated as a means for communication and networking among unmanned aerial vehicles is investigated in [6]. The authors discuss the challenges associated with the integration of unmanned aerial vehicles and cognitive radio technology. An analysis of topology algorithms for commercial airborne networks is presented in [7]. The authors present an airborne network architecture based on free-space optical communications links that form a high-bandwidth mesh network. Evaluation of a multihop airborne IP network with heterogeneous radio technologies is presented in [8]. The authors discuss performance of link, radio-to-router interface, and multihop network they simulated using open source platforms.

There is also significant amount of research present in the literature on the trade-offs among the primitive measures such as throughput, reliability, and latency that are discussed in our work. A model to balance the relationship between throughput and latency for a multihop communication link is presented in [9]. Throughput, delay and reliability trade-offs are investigated in [10]. Their results suggest that single hop transmissions are optimal for maximizing the lower bound on the transmission capacity in the sparse network regime. A quality-of-service profile based on throughput, delay and reliability trade-offs in body area sensor networks is presented in [11]. This analysis is intended for time critical bio-medical applications. Throughput, delay and reliability trade-offs in multihop networks with random access are investigated in [12]. The authors characterized the trade-offs between the achievable throughput, end-to-end delay and reliability in wireless networks with random access.

The research discussed in our paper provides a generic framework for performance modeling and analysis of extremely dynamic networks. We present a novel algorithm to discover an optimal route between any two nodes in a network given a performance criteria represented by an arbitrary function. The algorithm is demonstrated to converge rapidly. This algorithm could be used to generate network routing tables in a centralized controller such as it would be in *Software Defined Networking* (SDN) paradigm.

3 Aerial Network

An aerial network is formed by aircraft deployed for a specific mission. The network may include manned and unmanned aircraft systems (UASs), ground vehicles, control stations and services, as illustrated in Fig. 1. In an aerial network nodes may travel at high speeds with range extending to hundreds of miles and with network topology constantly changing.

Successful deployment of aerial networks requires comprehensive modeling and simulation beforehand. Modeling and simulation of airborne networks, in turn, requires models of airborne vehicles, antenna propagation patterns, mobility models, terrain models, and weather patterns. Deployment of successful aerial networks also requires the implementation of information assurance strategies and their integration with network management and planning tools. An aerial network with its dynamic topology can best be represented as a random graph.

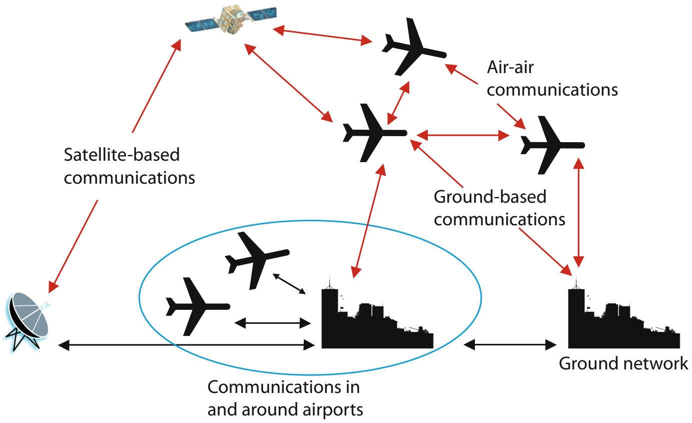


Fig. 1. An illustration of airborne network consisting of terrestrial, satellite, and RF links to connect to control stations on the ground and to other airplanes.

4 Aerial Network as a Random Graph

This section outlines the mathematical preliminaries of random graphs. We consider networks with two terminals: a source (s) node and a destination (t) node and follow the notation used in [13]. Let $G = (V, E, P)$ represent a probabilistic graph with a set of nodes $v_i \in V$, a set of edges $e_{ij} \in E$, and a link failure probability matrix $p_{ij} \in P$. Let $G_{st}(\mathrm{V}, E_{st}, P_{st})$ represent an overlay graph containing a path from s to t with its associated set of edges and probabilities (E_{st}, P_{st}). An overlay graph is created during the route discovery process (RDP); a process followed by a source node to find its destination node. Although either nodes or links may fail in a network, the scope of this analysis is limited to networks with link failures only, i.e., nodes are assumed to be failure-free. An edge e_{ij} represents a link connecting two adjacent nodes v_i and v_j. A path between two nodes v_i and v_j which is not adjacent to each other in G is defined as a sequence of distinct links connecting the two nodes. Information flows from one node to another as long as there is a path connecting the two nodes. A '$(s - t)$' cut divides the set of vertices V in the graph $G_{st}(\mathrm{V}, E_{st}, P_{st})$ into two disjoint subsets S and T such that $s \in S$ and $t \in T$. $C_{st}(i)$ represents a cut-set indexed by i in the overlay graph connecting the two nodes s and t (Table 1).

4.1 Reliability Analysis

This section outlines the concept of reliability in the context of an aerial network. It provides an approach to estimate the reliability of a link between a pair of nodes, a path between any source and destination nodes and the reliability of the entire network.

A link failure in a dynamic network could be due to link attributes such as mobility and orientation of a node. For example, a link failure may be temporary

Table 1. Terminology and notation used to represent a graph [13].

G(V, E, P)	A network with the set of nodes V, set of links E and link failure probability matrix P
s	A source node in the network G
t	A destination node in the network G
S	The set of all source nodes
T	The set of all destination nodes
NS	A set of network states
NS_i	Network state i
$G_{st}(\mathrm{V}, E_{st}, P_{st})$	An overlay network that contains a path from s to t with its associated $(\mathrm{V}, E_{st}, P_{st})$
n	Number of nodes, $\|V\|$
m	Number of edges, $\|E_{st}\|$
nc	Number of cut-sets in a graph
Fp	Probability that a network is disconnected
$s - t$	A cut in $G_{st}(\mathrm{V}, E_{st}, P_{st})$ where $s \in S$ and $t \in T$
$C_{st}(i)$	i^{th} cut-set of $G_{st}(\mathrm{V}, E_{st}, P_{st})$
c_{ij}	Capacity of the link e_{ij}
$c(S, T)$	Capacity of an $s - t$ cut in a static network
$c_p(S, T)$	Capacity of an $s - t$ cut in a probabilistic network
$R_{st}(G_{st})$	Reliability of route between s and t
R	Reliability of an entire network
$\Re_{st}$	Data flow between s and t
z_{st}	Probability that a data flow occurs between s and t
$l(i, j)$	Link latency between two nodes i and j
$L(s, t)$	Path latency between the source node s and its destination node t
F_p	Probability that the network gets disconnected

as the link may become active again when the node comes back within the range of another node which is connected to the network. On the other hand, if a node fails, it will be removed from the aerial network. The topology of the network might change when a node is disconnected from one node and is connected back again possibly to a different node. Hence, it is reasonable to assign a probability of failure to every link in the network.

An overlay network is created while a node s is discovering a path to its destination t. Although the graphs, in general, may be directed, we consider undirected graphs for simplicity of analysis. The model can easily be extended to directed graphs as well. While the probability of failures may be different from

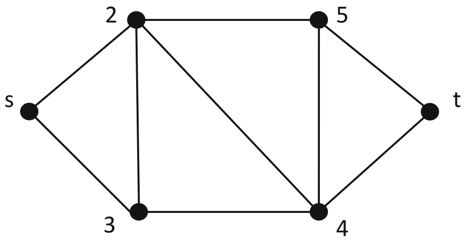


Fig. 2. An illustration of an overlay network [14]

one link to another, for simplification, it is assumed that the failure probabilities are the same for all the links, i.e., $p_{ij} = p$ and that they are independent of one another. For illustration purpose, let us consider a benchmark graph from among those given in [14] shown in Fig. 2. It represents a typical overlay network created during a route discovery process (RDP).

Reliability is a performance measure for the overlay network created during RDP between two nodes. Network reliability can be computed as a function link failure probabilities and cut-sets in the corresponding graph. The problem of enumerating all cut-sets in a graph is an NP-hard problem, for which a solution is proposed in [14]. The graph shown in Fig. 2 has the following cut-sets:

$$\begin{aligned} C_{st}(2) &= \{\{e_{s2}, e_{13}\}, \{e_{5t}, e_{4t}\}\} \\ C_{st}(3) &= \{\{e_{s2}, e_{23}, e_{34}\}, \{e_{25}, e_{24}, e_{34}\}, \{e_{25}, e_{54}, e_{4t}\}\} \\ C_{st}(4) &= \{\{e_{5t}, e_{54}, e_{24}, e_{34}\}, \{e_{s3}, e_{23}, e_{24}, e_{25}\}\} \\ C_{st}(5) &= \{\{e_{s2}, e_{23}, e_{24}, e_{45}, e_{4t}\}, \{e_{s3}, e_{23}, e_{24}, e_{54}, e_{5t}\}\} \end{aligned} \tag{1}$$

Each $C_{st}(i)$ above lists all the cut-sets in the graph that contain exactly i physical links. Assume that there are m physical links in a network between s and t, i.e., $(|E_{st}| = m)$. Let p represent the probability of link failure. The failure probability of a network state NS_i with exactly i physical link failures, is $p^i(1-p)^{m-i}$. Let N_i be the number of disconnected states in NS_i with $|NS_i| = N_i$. Then, the probability that the network gets disconnected (F_p) is the sum of the probabilities over all disconnected states, i.e.,

$$F_p = \sum_{i=0}^{m} N_i p^i (1-p)^{m-i} \tag{2}$$

Reliability of a two-terminal network is defined as the probability of having atleast one path between the two nodes [15]. When viewed as the complement of the network failure probability, it can be expressed as follows:

$$R_{st}(G_{st}) = 1 - \sum_{i=0}^{m} N_i p^i (1-p)^{m-i}. \tag{3}$$

Network failure states (NS) can be completely characterized by cut-sets. With the use of cut-sets, reliability $R_{st}(G_{st})$ of the network $G_{st}(V, E_{st}, P_{st})$ [13] can be expressed in the following closed form:

$$R_{st}(G_{st}) = 1 - \sum_{nc} \sum_{i=0}^{m} |C_{st}(i)| p^i (1-p)^{m-i} \tag{4}$$

where $m = |E_{st}|$ is the cardinality of the edge set E_{st}, nc is the number of cut-sets, and $|C_{st}(i)|$ is the cardinality of cut-set with exactly i edges. The reliability of the entire overlay network can be defined as [13]

$$R = \frac{\sum_{s \in V} \sum_{t \in V, t \neq s} z_{st} R_{st}(G_{st})}{n(n-1)} \tag{5}$$

where $n = |V|$, and z_{st} is the probability that a data flow occurs between the two nodes s and t.

4.2 Throughput Analysis

Throughput of a network can be estimated using cut-sets of a graph that represents the network. The concept of max-flow min-cut strategy to estimate the throughput of a network was first introduced in [16]. This section extends this concept to probabilistic networks.

Definition 1. *Cut-set: An $s-t$ cut-set is a partition of V such that $s \in S$, $t \in T$, and S and T are disjoint subsets of V.*

Definition 2. *Capacity of a Cut-set: The capacity of a $s-t$ cut-set is defined as follows:*

$$c(S,T) = \sum_{(u,v) \in (S \times T), (i,j) \in E} c_{ij} d_{ij} \tag{6}$$

where c_{ij} is the capacity of the link e_{ij} and $d_{ij} = 1$ if $i \in S$ and $j \in T$, 0 otherwise. Minimum $s-t$ cut is obtained by minimizing $c(S,T)$.

Definition 3. *Max-flow min-cut: The max-flow min-cut theorem suggests that the maximum amount of data passing from the source (s) to the destination (t) in a network is equal to the amount of flow corresponding to the minimum $s-t$ cut [16].*

Throughput for a probabilistic network needs to take the reliability of the links into account. In its simplistic form, throughput of an unreliable link can be obtained by multiplying the amount of flow on the link with its reliability. Thus, the capacity of an $s-t$ cut in a probabilistic network can be expressed as follows:

$$c_p(S,T) = \sum_{(u,v) \in (S \times T), (i,j) \in E} c_{ij}(1 - p_{ij}) d_{ij} \tag{7}$$

where p_{ij} represents the failure probability of the link e_{ij}. The minimum $s-t$ cut for a probabilistic network will be different from a static network. Hence, the throughput of a probabilistic network could be different from the throughput of a static network.

4.3 Latency Analysis

Link latency ($l_{i,j}$) is a parameter that characterizes an aerial communication link (i, j) between two nodes i and j. If a path consisting of n number of nodes exists between a source s node and its destination (t) node, then, the path latency, $L(s, t)$, is the sum of the latencies corresponding to the sequence of links $\{(\mathrm{s}, 2), (\mathrm{s}, 3), \ldots, (\mathrm{i}, \mathrm{i}+1), \ldots, (\mathrm{n}-1, \mathrm{t})\}$ that constitute the path $(\mathrm{s} - \mathrm{t})$.

$$L(s,t) = l_{s,2} + \sum_{i=2}^{n-2} l_{i,i+1} + l_{n-1,t} \tag{8}$$

Path latency can be viewed as the end-to-end delay between the source and destination nodes assuming that there is no queuing delay. Latency is a deterministic parameter for a given path unlike throughput which is a function of the reliability of the communication links between the source and destination nodes.

4.4 Security Analysis

Security of a link ($S_{i,j}$) is a probabilistic parameter that characterizes an aerial communication link (i, j) between two nodes i and j. If a path consisting of n number of nodes exists between a source (s) node and its destination (t) node, then, the security of the path $(s - t)$ can be represented by $S(s, t)$. Detailed security analysis is beyond the scope of this paper.

5 Goodness Measures

This "goodness" determination is based on three levels of analysis. At the first level, goodness of a link can be estimated in terms of basic measures such as reliability, throughout, and latency of communication. At the second level, goodness of a path can be measured between a pair of nodes that need to share data and network control information. At the third level, goodness is measured for the entire network. Below, we develop a general hierarchical framework that includes these three levels of analysis.

5.1 First Level Analysis: Link Metrics

A link represents one hop communication channel between a pair of nodes. Metrics defined over a link represent the basic measures that characterize the quality of communication over the link. One can define a goodness measure for a communication link as a mapping (g) from the set of basic measures (M) defined on the link to a goodness function. This function assigns weights (W) to basic measures and combines the weighted measures in some form to estimate the goodness of a communication link.

$$g \colon M \mapsto f(M, W) \tag{9}$$

Link metrics may be deterministic (M_D) or probabilistic (M_P). There may be more than one link between a pair of communicating nodes. In this case, the mapping needs to take into account the metrics corresponding to all available links.

Goodness of a link ($g(i-j)$) can be formulated as a function of basic measures such as reliability ($R_{ij}(G_{ij})$), throughput, latency ($l_{i,j}$), and security ($S_{i,j}$) of that specific communication link. Out of these measures, throughput and latency are deterministic measures and form a subset M_D. Reliability and security are probabilistic measures and form a subset M_P.

In general, the subset M_P represents a collection of probabilistic measures $\{m_{p1}, m_{p2}, \ldots, m_{pn}\}$ and the subset M_D represents a collection of deterministic measures $\{m_{d1}, m_{d2}, \ldots, m_{dn}\}$ as described below.

$$\begin{aligned} M &= M_D \cup M_P,\ \mathit{where}\ \ M_D \cap M_P = \emptyset \\ M_D &= \{m_{d1}, m_{d2}, \ldots, m_{dn}\} \\ M_P &= \{m_{p1}, m_{p2}, \ldots, m_{pn}\} \end{aligned} \tag{10}$$

Thus, the goodness of a communication link can be estimated as a weighted function of M_D and M_P.

$$g(\mathit{link}) = f(M_D, M_P, W) \tag{11}$$

5.2 Second Level Analysis: Path Metrics

The link analysis discussed above can be extended to path level. At this level, each link is seen as a potential connection that facilitates information flow between a pair of nodes that it connects. Goodness of a path is an estimation of the connectivity between a pair of nodes that may be one or more hops away from each other. As a path represents a sequence of links, the mapping needs to take into account the metrics corresponding to the sequence of links that form the path. If a path consisting of n number of links exists between a source (s) node and its destination (t) node then the goodness of that path, represented by $g(\mathit{path})$, is described as follows:

$$g(\mathit{path}) = f(g(\mathit{link}(1)), g(\mathit{link}(2)), \ldots, g(\mathit{link}(n))) \tag{12}$$

where ($\mathit{link}(1)$, $\mathit{link}(2)$, ..., $\mathit{link}(n)$) constitutes the path. For illustration purposes, let us consider the benchmark graph shown in Fig. 2 as an example. The goodness function given in Eq. 12 may be used to compare goodness values of two different paths between a pair of nodes. For example, there are two paths from node s to node 2: $s-2$ and $s-3-2$ in the graph shown in Fig. 3. While $s-2$ is a direct path from node s to node 2 with just one link, $s-3-2$ includes two links. Even though $s-2$ is the one hop link, it may be possible that $g(s-2)$ may be worse than $g(s-3-2)$.

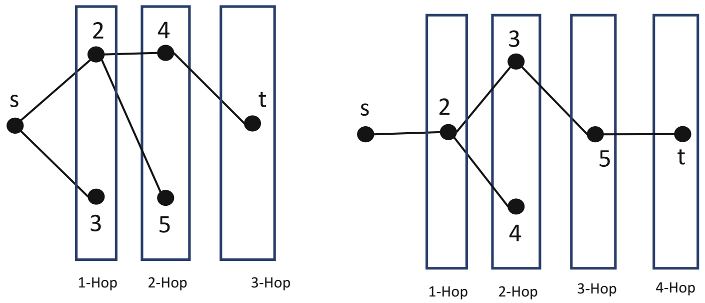


Fig. 3. Two possible configuration graphs q_1 and q_2 showing the best possible paths from the s to the rest of the nodes in the network

5.3 Third Level Analysis: Network Metrics

The network considered here is an overlay network, a partial network found by node s while it is trying to find possible routes to a destination node t. The third level analysis requires us to create a network configuration graph (q_1), a multi-layer graph starting from the source node to the destination node. In the configuration graph n^{th} layer represents the set of all nodes that can be reached from the source node in n hops. This process of generating configuration q_1 is described in Algorithm 1 below.

Algorithm 1. Algorithm for Generating the Configuration Graph

```
*Initialization*
Let $S = \{s, 2, 3, \ldots, t\}$ be the set of all nodes in the network and Let $S^{'}$={};
Move the source node ($s$) from S to $S^{'}$;
**while** there are nodes in the set $S$ **do**
    **for** each node $i$ in set $S$ **do**
        identify the set of neighbors $N_i$ of $i$
        **for** each node $j$ in $N_i$ **do**
            **if** the optimal path from $s$ to $j$ includes $i$, and $i \notin S^{'}$ **then**
                update the path from node $s$ to $j$;
            **end if**
        **end for**
        Move the node $i$ to $S^{'}$;
    **end for**
**end while**
```

Algorithm 1 outlines a systematic way to find optimal paths from a source (s) node to a destination (t) node. Starting from s, the algorithm enumerates the neighbors of s and finds the best routes to reach these neighbors in the first iteration. This process continues iteratively until all nodes are included in the configuration graph. In each iteration, the routes are refined taking into account the new neighbors added in each iteration. Figure 3 shows two possible configuration graphs that were generated using Algorithm 1. Configuration q_1 is generated when $g(s-2) < g(s-3-2)$ and $g(2-4) < g(2-3-4)$. Configuration q_2 is generated when $g(s-3) > g(s-2-3)$ and $g(2-4) > g(2-3-4)$.

6 Summary

This paper presented a hierarchical framework for computing goodness measures for links and paths in dynamic networks. A strategy for finding an optimal path between a pair of nodes is presented as an application for this hierarchical framework.

Acknowledgments. This research work was initially carried out at the Air Force Research Laboratory, Rome, NY, USA, as part of the visiting faculty research program during the summer 2016. It was continued at the University of North Texas with the support from the National Science Foundation through the smart and connected communities program, grant No. 1622978.

References

1. Sterbenz, J.P.G., Çetinkaya, E.K., Hameed, M.A., Jabbar, A., Qian, S., Rohrer, J.P.: Evaluation of network resilience, survivability, and disruption tolerance: analysis, topology generation, simulation, and experimentation. Telecommun. Syst. **52**(2), 705–736 (2013)
2. Çetinkaya, E.K., Broyles, D., Dandekar, A., Srinivasan, S., Sterbenz, J.P.G.: Modelling communication network challenges for future internet resilience, survivability, and disruption tolerance: a simulation-based approach. Telecommun. Syst. **52**(2), 751–766 (2013)
3. Sterbenz, J.P.G., Hutchison, D., Çetinkaya, E.K., Jabbar, A., Rohrer, J.P., Schller, M., Smith, P.: Redundancy, diversity, and connectivity to achieve multilevel network resilience, survivability, and disruption tolerance invited paper. Telecommun. Syst. **56**(1), 17–31 (2014)
4. Bhattacharya, A., Kumar, A.: A shortest path tree based algorithm for relay placement in a wireless sensor network and its performance analysis. Comput. Netw. **71**, 48–62 (2014)
5. Sbeiti, M., Goddemeier, N., Behnke, D., Wietfeld, C.: PASER: secure and efficient routing approach for airborne mesh networks. IEEE Trans. Wirel. Commun. **15**(3), 1950–1964 (2016)
6. Saleem, Y., Rehmani, M.H., Zeadally, S.: Integration of cognitive radio technology with unmanned aerial vehicles: issues, opportunities, and future research challenges. J. Netw. Comput. Appl. **50**, 15–31 (2015)
7. Newton, B., Aikat, J., Jeffay, K.: Analysis of topology algorithms for commercial airborne networks. In: 2014 IEEE 22nd International Conference on Network Protocols (ICNP), pp. 368–373. IEEE (2014)
8. Cheng, B.-N., Charland, R., Christensen, P., Veytser, L., Wheeler, J.: Evaluation of a multihop airborne IP backbone with heterogeneous radio technologies. IEEE Trans. Mobile Comput. **13**(2), 299–310 (2014)

9. Sparrow, R.D., Adekunle, A.A., Berry, R.J., Farnish, R.J.: Balancing throughput and latency for an aerial robot over a wireless secure communication link. In: 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF), pp. 184–189. IEEE (2015)
10. Vaze, R.: Throughput-delay-reliability tradeoff in ad hoc networks. In: 2010 Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), pp. 459–464. IEEE (2010)
11. Akbar, M.S., Yu, H., Cang, S.: Delay, reliability, and throughput based QOS profile: a MAC layer performance optimization mechanism for biomedical applications in wireless body area sensor networks. J. Sens. **2016**, 17 (2016)
12. Srinivasa, S., Haenggi, M.: Throughput-delay-reliability tradeoffs in multihop networks with random access. In: 2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 1117–1124. IEEE (2010)
13. Caleffi, M., Ferraiuolo, G., Paura, L.: A reliability-based framework for multi-path routing analysis in mobile ad-hoc networks. Int. J. Commun. Netw. Distrib. Syst. **1**(4–6), 507–523 (2008)
14. Benaddy, M., Wakrim, M.: Cutset enumerating and network reliability computing by a new recursive algorithm and inclusion exclusion principle. Int. J. Comput. Appl. **45**, 22–25 (2012)
15. Lee, K., Lee, H.-W., Modiano, E.: Reliability in layered networks with random link failures. IEEE/ACM Trans. Netw. (TON) **19**(6), 1835–1848 (2011)
16. Elias, P., Feinstein, A., Shannon, C.: A note on the maximum flow through a network. IRE Trans. Inf. Theory **2**(4), 117–119 (1956)

An Efficient Routing and Interface Assignment Algorithm for Multi-Channel Multi-Interface (MCMI) *Ad Hoc* Networks

Yifeng Zhou(✉)

Communications Research Centre Canada, Ottawa, ON, Canada
yifeng.zhou@canada.ca

Abstract. In this paper, a routing and interface assignment algorithm is proposed for MCMI wireless *ad hoc* networks. The algorithm consists of two steps: route selection and interface assignment. The process of route selection is to find the path with the minimum lower bound while the interface assignment is to assign the interfaces on the nodes along the path based on the application of the Viterbi algorithm. The proposed algorithm is computationally efficient due to the decoupling of the route selection and interface assignment processes. Computer simulation and examples are used to demonstrate the effectiveness and performance of the proposed technique. Comparisons are made to other existing routing techniques in the area of dynamical spectrum access.

Keywords: *Ad hoc* networks · MANET · Routing
Multi-Channel Multi-Interface (MCMI) · Viterbi algorithm

1 Introduction

A MANET (Mobile *Ad hoc* Network) is a type of *ad hoc* networks that can be deployed quickly without any prior planning or construction of expensive network infrastructure [1]. In the last two decades, MANET has attracted lots of interest from academics as well as industries. In a traditional MANET, all nodes use a single common channel for communications, which eliminates the need for coordination between adjacent nodes. In addition, the use of single channel links can greatly reduce the cost of a wireless network since each node only needs one wireless interface. However, the throughput capacity of a single channel network is significantly limited due to simultaneous transmission on a same channel [2,3]. A popular approach for improving the network capacity performance is to use orthogonal transmissions among adjacent hops to minimize collision and channel interference. More recently, a trend is to use multiple channels and multiple interfaces (also referred to as radios) on each node as a means for multiple simultaneous orthogonal transmissions (see [4] and references thereafter). The use of multiple interfaces has been accelerated by the recent rapid advancement in communications technologies and hardware systems that are becoming more

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 131–142, 2018.
https://doi.org/10.1007/978-3-319-74439-1_12

powerful, more compact and less expensive, and more energy efficient. It becomes feasible to fit multiple interfaces on a node to support the use of multiple channels for MANET applications. The widely used technology IEEE 802.11a [5] is already known to support multiple channels by switching from one channel to another. The development of spectrum agile software-defined radio (SDR) [6,7] is another major driving force behind the adoption of MCMI networks. SDRs can be programmed to tune to a wide spectrum range and operate on any frequency bands in the range.

Recently, many routing and medium access control (MAC) protocols have been developed for MCMI networks [4,8–17]. In [19], by assuming that the number of available interfaces on each node is less than the number of the available channels, the authors proposed an interface assignment strategy, in which one interface is fixed for coordination while the others can be switched. Routing heuristics were then discussed. Wu *et al.* [20] proposed a MAC protocol that requires two interfaces on each node: one interface is assigned to a common channel for control messages, and the second one is switched between the other channels for data communications. In [21], a similar 2-interface solution was discussed, in which a channel (or interface) is selected for data communications based on the load of the channel. In [22], a multi-channel MAC protocol is proposed for IEEE 802.11, which requires only one interface on each host and solves the multi-channel hidden terminal problem using temporal synchronization. The development of routing techniques for MCMI networks has some new challenges due to channel diversity and the use of multiple interfaces on each node. Traditional *ad hoc* routing algorithms cannot handle multi-channels efficiently since they are designed for single channel networks. In general, the steps of route selection and channel assignment can be executed either simultaneously or in a decoupled way [15–17]. In [16], a layered graph was proposed to model the discovered spectrum opportunities (SOPs), which is then used to develop efficient and routing and interface assignment algorithms to form near-optimal topologies for dynamical spectrum access (DSA) networks. The construction of the layered graph is to fully utilize the forwarding capability at each node to choose different channels on different hops of a path, and to ensure that adjacent hop interference is minimized. A main shortcoming of the layered graph routing algorithm is the heavy computational complexity involved in the construction of the layered graph and the search for the shortest path due to the increase in the number of compound subnodes in the graph. In [17], a colored multigraph based model was proposed for utilizing spectrum holes for cognitive radio networks. In the colored multigraph model, colored edges are used to represent potential neighbor nodes that share some common channels between them. The goal is to maximize the network capacity and minimize adjacent hop interference among neighboring nodes. The algorithm takes into account the effects of both adjacent hop interference and the number of interfaces available on a current node. This approach is computationally efficient with computational complexity on the order of $O(N^2)$, where N denotes the number of nodes in a network. However, the algorithm only provides locally optimized adjacent hop interference due to the fact that routing selection and channel assignment is executed simultaneously at a local level.

In this paper, we focus on the problem of routing and interface assignment for MCMI networks. More specifically, we try to find the optimum path between a source and a destination nodes given the numbers of available interfaces on each node and the sets of available channels between each pair of nodes in the network. It is assumed that the interfaces on each node can be tuned to different channels but one at a time. We assume that the number of interfaces on each node is less than the number of available channels, and the numbers of interfaces and available channels may differ for each individual node. In this work, we use a common channel approach for resource management including neighbour discovery and exchange of control messages [18,20]. On each node, a dedicated interface is assigned to the common channel. The proposed routing and interface assignment algorithm first finds the shortest path that minimizes the lower bound cost metrics among all feasible routes between the a source and a destination node. Unlike the routing algorithm in [17], channels are not assigned on each hop of the path. In the second step, interface assignment based on the Viterbi algorithm is applied to assign the interfaces on each node along the shortest path to achieve a minimized adjacent hop interference. The Viterbi algorithm is a dynamic programming algorithm for computing the most probable sequence of states in a hidden Markov model given a sequence of observations. In order to apply the Viterbi algorithm, we use a trellis to model the nodes and all available channel along the shortest path. In the trellis, each state represents a channel between a pair of consecutive node on the route. The contribution of the paper is three-fold. First, an effective cost metric is developed, which takes into account the effects of adjacent hop interference and the availability of interfaces on the nodes. Secondly, the idea of decoupling the processes of finding the shortest path and optimal channel assignment is new. The decoupling as well as the use of lower bound metric helps to achieve the globally optimality in routing selection and interface assignment. Thirdly, the Viterbi algorithm is successfully applied in the context of interface assignment for achieving the global optimal adjacent hop interference.

The paper is organized as follows. In Sect. 2, the problem of routing and interface assignment is formulated. Assumptions about MCMI networks are also made in this section. Section 3 is devoted to the development of the proposed routing and interface assignment techniques. In this section, a cost metric is defined, and the algorithm for routing and interface assignment is discussed in detail. Finally, in Sect. 4, computer simulations and examples are used to demonstrate the effectiveness of performance of the proposed routing algorithm. Comparisons are made with other existing routing algorithms for MCMI networks.

2 System Model and Assumptions

Assume that a network consists of N nodes, and that each node has I configurable half-duplex interfaces that can be tuned to one channel at a time.

A half-duplex radio cannot transmit and receive simultaneously, *i.e.*, it can only transmit or receive at a time. It is also assumed that, for each node, there are a maximum M channels available for data communications.

Routing in an MCMI *ad hoc* network can be formulated as the following problem: given a source node s and a destination node t, find the optimum path and assign a channel to an interface on each node along the path so that the resulting path for data transport between the source and destination nodes is optimum. The optimality of the route is measured in terms of a cost metric that accounts for both the number of hops the route traverses and the effects of adjacent hop interference involved.

For the proposed routing technique, it is required that all nodes in the network have the global view of the network. First, each node must detect the neighbor nodes with which it has a direct link, and obtain information about the available number of interfaces and channels on each neighbor node. This process is also referred to as the neighbor discovery. In a single-channel network, since all nodes operate on a same channel, neighbor discovery can be achieved by, for example, all nodes periodically exchanging beacons [27]. In an MCMI network, since all nodes do not stay not on a same channel, they may not always hear each other on all channels. Many neighbor discovery approaches have been developed in the literature for both signal-channel and multiple-channel ad hoc networks in the literature (*cf.* [26] and thereafter). In this work, we use the common channel approach. A common control channel is assumed for all nodes for neighbor discovery and resource management purpose [18,20]. On each node, a dedicated interface is assigned to the control channel for exchange of control messages. The process of neighbor discovery is implemented on the channel similar to the one in OLSR (Optimized Link State Routing) protocol [27]. Each node periodically transmit beacons that contain the list of neighbors, and the numbers of channels and interfaces available to them. The beacons are received by all one-hop neighbors, which enable each node to discover its one-hop neighbors as well as two-hop neighbors. Based on the information from the received beacons, each node regularly floods the topology control information about its up to two-hop neighbors to the entire network. Each node maintains the topology information of the network obtained from the dissemination of the topology control information.

3 Optimum Routing for MCMI Networks

The proposed MCMI routing algorithm consists of two decoupled steps: finding the shortest path and optimally assigning a channel to an interface on each node along the path. The shortest path is defined as the one that has the minimized lower bound cost among among all available paths. In this step, no interface assignment is implemented. The interface assignment algorithm uses the Viterbi algorithm to achieve global rather than local optimality in assigning interfaces on each node along the shortest path.

3.1 Shortest Path with Minimized Lower Bound Cost

The routing is formulated as the problem of finding the shortest path between the source node and the destination node, where the path is the sum of cost metrics defined for all connections along the path between the source node and the destination node. In general, the proposed approach for finding the shortest path uses similar procedures as Dijkstra's algorithm [23]. However, Dijkstra's algorithm is not directly applicable, since it is suitable to networks, where any two nodes are connected with one edge or channel of fixed cost. In an MCMI network, any two nodes may be connected by multiple channels, and the cost of selecting one channel may be different from the selection of another channel due to channel interference. We assign each node a distance value that represents its distance from the source node. Since in an MCMI network, multiple edges (channels) exist between each pair of nodes, we define additional metrics to represent the cost that a node choose one of the available channels for routing data. The cost metric should be able to take into account both the weight of an edge and channel diversity along a path. Figure 1 shows the connection between a current and a neighbor node, where E_p denotes the edges that connect to the current node v_c, and E_c denotes the edges between the current and next neighbor node. We assume that each edge is characterized by a cost of one (hop). It should be noted that the problem of how to define the cost of an edge is not the focus of this study, and the cost assumption is only for the purpose of demonstration. In practice, the edge cost can be extended to include other factors depending on the applications. For example, the cost can be generalized to use the link state of the edge, *i.e.*, a real number that represents the effective capacity and quality of the edge (link).

Definition 1. Let v_c be the current node, and E_{pi} be a channel that connects to v_c. Let E_{ck} be a channel that connects with a neighbor node. The cost of selecting E_{ck}, denoted by c, can be computed according to the following rules:

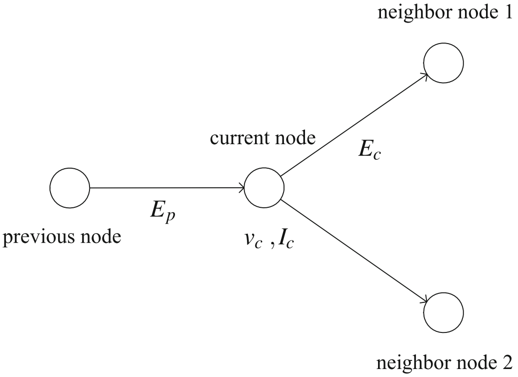


Fig. 1. Connection cost between current and neighbor node.

1 if no interface is available on v_c, $c = \infty$.
2 when v_c has one interface, if E_{pi} and E_{ck} represent a same channel, then $c = 1 + \beta$, where β denotes a penalty term;
3 when v_c has more than one interface, if E_{pi} and E_{ck} are different channels, $c = 1$; otherwise, $c = 1 + \beta$.

The above definition provides the cost metric for connecting to a neighbor node on a given channel. In the following, we define *the lower bound cost metric* for connecting a pair of nodes for the case of multiple channels between the two nodes. Note that in this metric, no specific channel is specified that connects the pair.

Definition 2. Let v_c be the current node. Denote E_p as the channels that connect to v_c, and E_c the available channels between v_c and another neighbor. The cost of selecting the path that connects v_c and v_n is computed according to the following rules.

1 If no interface is available on v_c, $c = \infty$.
2 When v_c has only one interface, if E_p and E_c share at least one common channel, then, $c = 1 + \beta$; The same rule applies when one or multiple consecutive nodes preceding the current node on a path have only one interface.
3 When v_c has more than one interface, if E_p and E_c each has at least one non-common channel, then, $c = 1$; otherwise, $c = 1 + \beta$.

This cost metric defines the minimum cost that v_c connects to v_n given the edges in E_c. In other words, the cost is a lower bound on the cost of selecting any edge from E_c under any given edge in E_p.

The route selection algorithm is used to find the shortest route between the source and the destination node with the minimized lower bound cost metric. All the nodes of the network are classified into two exclusive sets: visited and unvisited. At the beginning, all nodes are marked unvisited. The algorithm iterates over all unvisited nodes until all nodes are marked visited. At each step, a current node will be found and the distances of all its unvisited neighbor nodes will be evaluated and updated. The distance of an unvisited neighbor node is updated as the distance of the current node plus the cost of edges connecting to the neighbor node.

3.2 Interface Assignment

Although the effects of channel availability on each hop is taken into account, channels are not explicitly assigned to interfaces on the nodes. Since there may be more than one channel available between a pair of adjacent nodes on the path, and more than one interface on each node, the interfaces on each node along the path need to be assigned with an available channel in order to minimize the adjacent channel interference [16,17]. In this paper, we use a trellis to describe the interface assignment problem. Let $\mathcal{P}$ denote the shortest path from the route selection algorithm, which is assumed to consists of $l + 1$ nodes. Let $\underline{e}_i$ denote

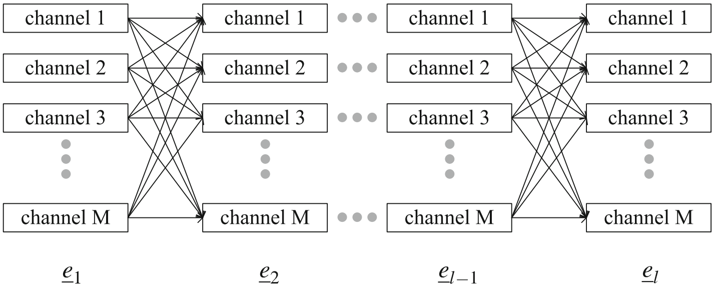


Fig. 2. Cost functions for different scenarios.

channels that are available between the ith and the $(i+1)$th adjacent node on $\mathcal{P}$. The trellis shown in Fig. 2 depicts all connections between the source and the destination when multiple channels exist between pairs of adjacent nodes along the shortest path. The problem is to find the path that has the minimized cost. The Viterbi algorithm [24,25] is perhaps the most popular technique for solving such a problem. The Viterbi algorithm is a dynamic programming algorithm for computing the most probable sequence of states in a hidden Markov model given a sequence of observations. The trellis in Fig. 2 has M states or channels denoted by $\{e_1, e_2, \ldots, e_M\}$. The path $\mathcal{P}$ consists of l hops, which are indexed by i with $1 \leq i \leq l$. The transition cost $\epsilon^{(i)}(j,k)$ is computed according to Definition 1.

The Viterbi algorithm is a recursive approach that runs from $i = 1$ to $i = l$. For each intermediate state in the trellis, the best partial path is computed as the one that has the minimum cost among all paths that end at the state. The algorithm will produce a set of optimum paths that reach the available states on the last hop in the trellis, and the sequence that has the minimum associated partial cost will be selected as the optimum channel assignment. From the state on the last hop on the shortest path, we then can use the back pointers to propagate backwards to recover the optimum path. The application of the Viterbi algorithm has the useful property of using the context of entire information of channels and interfaces on each node along the shortest path for judgement, and is able to provide the global optimum channel assignment solution. The Viterbi type algorithm is also computationally efficient due to the recursive nature of the algorithm.

4 Performance Analysis

In this section, we use two examples to show the performance and effectiveness of the proposed routing and interface assignment algorithm for MCMI *ad hoc* wireless networks. In the first example, a network is simulated, in which the nodes are randomly distributed in a square area of 10 m by 10 m. The simulated network consisted of 30 nodes. The number of interfaces on each node is randomly selected between 1 and 4. The number of available channels between a pair of connected nodes is assumed to be uniformly distributed between 1 and 6.

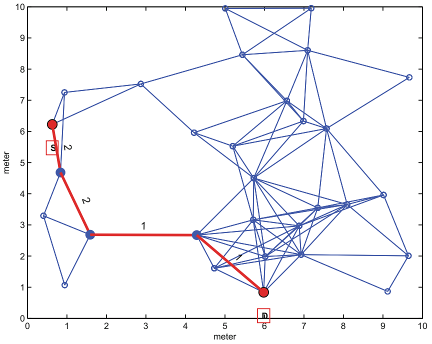


Fig. 3. The optimal route and interface assignment selected by the proposed routing algorithm.

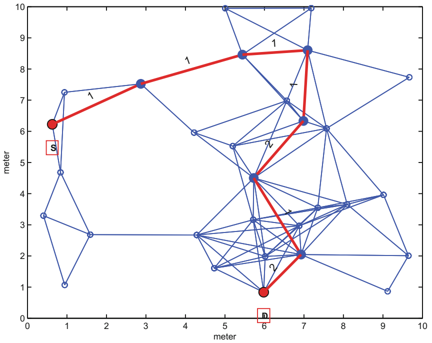


Fig. 4. The optimal route and interface assignment selected by the colored multigraph model based approach.

The penalty due to adjacent hop interference was selected as $\beta = 0.5$, which is equivalent to half of the cost for one hop. Figures 3 and 4 show the routing paths between the pair of source and destination nodes decided by the proposed approach and the colored multigraph model (CMM) based algorithm, respectively. In the figures, it can be seen that, the route selected by the proposed algorithm has 4 hops while the route computed by the CMM based algorithm has to traverse 7 hops for the source and destination nodes. In terms of adjacent hop interference, the route by the proposed algorithm contains two pairs of adjacent hops that have interfering channels while the route by the CMM based algorithm contains 3 pairs. For this example, the proposed routing algorithm outperforms the CMM based algorithm both in the number of hops that the route traverses and the adjacent hop interference. In the simulation studies, it was observed that the proposed routing algorithm had performed better in most case and was never worse than the CMM based algorithm.

Another example is used to demonstrate the flexibility of the proposed routing and interface assignment algorithm in dealing with weighting of adjacent hop interference by selecting different values for the penalty terms. As discussed before, the selection of the penalty value affects both the route selection and and the channel assignment. Increasing penalty value may force the routing algorithm to select a route that consists of more hops and less adjacent hop interference. In some applications, the prevention of adjacent channel interference is critical

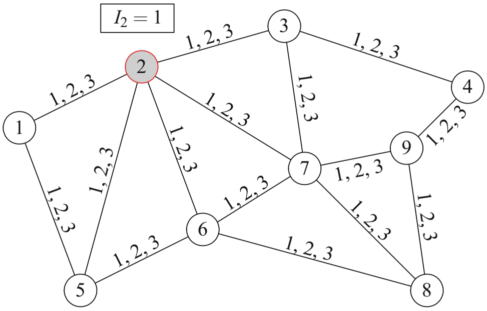


Fig. 5. An example of network for effects of penalty values.

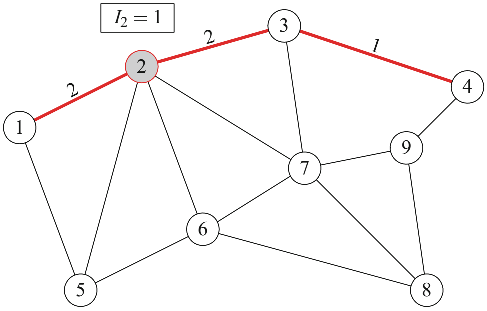


Fig. 6. Optimal route and channel assignment when $\beta = 0.5$ and 1.0.

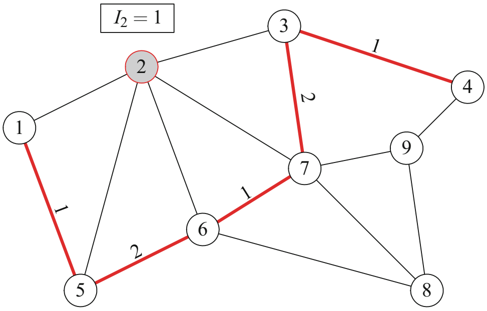


Fig. 7. Optimal route and channel assignment when $\beta = 2.5$.

to the operation of a network because channel interference may form a bottleneck for packet throughput on the selected route. Another advantage of reducing adjacent hop interference is that it can improve spectrum utilization of a network. The simulated network topology is shown in Fig. 5. The network consists of 9 nodes, where node 3 has one interface and the others are assumed to have 2 interfaces on them. Without loss of generality, all connected node pairs have channels 1, 2 and 3 available. First, we set the penalty value to 0.5 and 1, respectively, and apply the proposed routing algorithm. The routing results are shown in Fig. 6. The selected optimal route traverses 3 hops and contains one occurrence of adjacent hop interference at node 2, where the *in* and *out* routes are all on channel 2. The selected route has a minimized cost of 3.5. To demonstrate the effects of penalty value on the route selection algorithm, we increased the penalty value to 2.5. In this case, one occurrence of adjacent hop interference is equivalent to 2.5 hops in cost metric, and is considered a significant penalty for selecting a route with adjacent hop interference for data traffic. The selected route and the interface assignment selected by the proposed algorithm is shown in Fig. 7. The route has successfully avoided the route with adjacent hop interference. However, the tradeoff is an increase in the number of hops that the route traverses. The selected route has a minimized cost of 5.

5 Conclusions

In this paper, an optimized routing and interface assignment algorithm was presented for MCMI wireless *ad hoc* networks. The technique decouples the routing and interface assignment into two steps, *i.e.*, route selection and interface assignment. A cost metric was proposed that accounts for both the number of hops of a route to be traversed and the effects of adjacent hop interference. Unlike traditional path searching algorithms, a lower bound cost metric rather than the cost metric itself is used as the path searching criteria. The Viterbi algorithm is used to assign interfaces on the nodes along the shortest path to achieve the globally minimized adjacent hop interference. Computer simulation were used to demonstrate the effectiveness and performance of the proposed technique. The proposed routing algorithm is computationally efficient, and has the advantage of flexibility in dealing with weighting of adjacent hop interference by selecting different values for the penalty terms. Future studies will include other factors that will affect the routing performance such as switching latency and end-to-end delay.

References

1. Basagni, S., Conti, M., Giordano, S., Stojmenovic, I. (eds.): Mobile Ad Hoc Networking. IEEE Press, Wiley-Interscience, Piscataway (2004)
2. Gupta, P., Kumar, P.R.: The capacity of wireless networks. IEEE Trans. Inf. Theory **46**(2), 388–404 (2000)

3. Grossglauser, M., Tse, D.: Mobility increases the capacity of ad-hoc wireless networks. IEEE/ACM Trans. Networking **10**(4), 477–486 (2002)
4. Kyasanur, P., Vaidya, N.: Routing in Multi-Channel Multi-Interface Ad Hoc Wireless Networks. Technical report, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, December 2004
5. IEEE Standard for Wireless LAN-Medium Access Control and Physical Layer Specification. IEEE Computer Society, IEEE, New York, June 2007
6. Jondral, F.K.: Software-defined radio-basics and evolution to cognitive radio. EURASIP J. Wirel. Commun. Networking **3**, 275–283 (2005)
7. Fette, B.: Cognitive Radio Technology. Elsevier Science & Technology Books (2006)
8. Dang, D.N.M., Nguyen, V., Le, H.T., Hong, C.S., Choe, J.: An efficient multichannel MAC protocol for wireless ad hoc networks. Ad Hoc Netw. **44**(C), 46–57 (2016)
9. Draves, R., Padhye, J., Zill, B.: Routing in multi-radio, multi-hop wireless mesh networks. In: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom 2004), Philadelphia, PA, USA, pp. 114–128, October 2004
10. Miu, A.K., Balakrishnan, H., Koksal, C.E.: Improving loss resilience with multiradio diversity in wireless networks. In: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom 2004), Philadelphia, PA, USA, pp. 16–30, October 2004
11. Raniwala, A., Chiueh, T.C.: Architecture and algorithms for an IEEE 802.11-based multichannel wireless mesh network. In: Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), Miami, FL, USA, vol. 3, pp. 2223–2234, March 2005
12. Chereddi, C., Kyasanur, P., Vaidya, N.: Design and implementation of a multichannel multi- interface network. In: Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality, International Symposium on Mobile Ad Hoc Networking and Computing, Florence, Italy, vol. 5, pp. 23–30, May 2006
13. Maheshwari, R., Gupta, H., Das, S.: Multichannel MAC protocols for wireless networks. In: 2006 Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON 2006), Reston, VA, USA, vol. 2, pp. 393–401, September 2006
14. Sharma, A., Belding, E.: FreeMac: Framework for multi-channel MAC development on 802.11 hardware. In: Workshop on Programmable Routers for Extensible Services of Tomorrow (PRESTO), Seattle, WA, USA, August 2008
15. Wang, Q., Zheng, H.: Route and spectrum selection in dynamic spectrum networks. In: 2006 Third IEEE Consumer Communications and Networking Conference (CNCC), Las Vegas, Nevada, USA, pp. 625–629, 8–10 January 2006
16. Xin, C., Xie, B., Shen, C.C.: A novel layered graph model for topology formation and routing in dynamic spectrum access networks. In: Proceedings of the 2005 First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN), Baltimore, MA, USA, pp. 308–317, November 2005
17. Zhou, X., Lin, L., Wang, J., Zhang, X.: Cross-layer routing design in cognitive radio networks by colored multigraph model. Wirel. Pers. Commun. **49**, 123–131 (2009)
18. Jain, N., Das, S., Nasipuri, A.: A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks. In: The 10th IEEE International Conference on Computer Communications and Networks (IC3N), Scottsdale, AZ, USA, pp. 432–439, October 2001

19. Kyasanur, P., Vaidya, N.H.: Routing and interface assignment in multi-channel multi-interface wireless networks. In: IEEE Wireless Communications and Networking Conference (WCNC 2005), LA, USA, New Orleans (2005)
20. Wu, S.L., Lin, C.Y., Tseng, Y.C., Sheu, J.P.: A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In: Proceedings of the 2000 International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN 2000), Dallas, TX, USA, pp. 232–237, December 2000
21. Hung, W.C., Law, K.L.E., Leon-Garcia, A.: A dynamic multi-channel MAC for ad hoc LAN. In: Proceedings of the 21st Biennial Symposium on Communications, Kingston, Ontario, Canada, pp. 31–35, June 2002
22. So, J., Vaidya, N.H.: Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In: Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc 2004), Roppongi Hills, Tokyo, Japan, pp. 222–233, May 2004
23. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269–271 (1959)
24. Forney Jr., G.D.: The viterbi algorithm. Proc. IEEE **61**(3), 268–278 (1973)
25. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
26. Karowski, N., Viana, A.C., Wolisz, A.: Optimized asynchronous multi-channel neighbor discovery. In: Proceedings of the 30th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2011), Shanghai, China, pp. 536–540, April 2011
27. Clausen, T., Jacquet, P. (eds.), Project Hipercom (INRIA): Optimized link state routing protocol (OLSR). The Internet Engineering Task Force (IETF) Network Working Group RFC 3626, Experimental, October 2003

Exploiting Multiple Beam Antennas for End-to-End Delay Reduction in Ad Hoc Networks

Jean-Daniel Medjo Me Biomo, Thomas Kunz(✉), and Marc St-Hilaire

Department of Systems and Computer Engineering,
Carleton University, Ottawa, ON, Canada
{jemeda,tkunz}@sce.carleton.ca, marc_st_hilaire@carleton.ca

Abstract. Multi-Beam Antennas (MBAs) have two main characteristics: the Multi-Packet Transmission (MPT) capability and the Multi-Packet Reception (MPR) capability whereby a node can transmit/receive multiple packets at the same time. In this paper, we provide an analysis of how this MPT/MPR capability can be used to reduce the end-to-end delay in ad hoc networks. We formulate the delay reduction issue as an optimization problem. Simulations show that in order to exploit the full potential of MBAs for delay reduction, the scheduling of links has to promote the formation of *star nodes* and keep the formation of *bridges* to a minimum; which leads to the selection of routes that very often are not the shortest. We also show that using only the shortest routes has a negative impact on the delay.

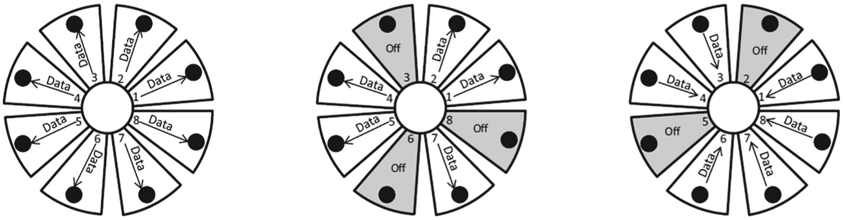
Keywords: Multi-beam antenna · Ad hoc network · Optimization Routing · Delay minimization

1 Introduction

A Multi-Beam Antenna (MBA) is defined by its Multi-Packet Transmission (MPT) and Multi-Packet Reception (MPR) capabilities that allow multiple packets to be transmitted/received at the same time. However, MBA-equipped nodes need to follow a rule called Concurrent Packet Receiving (CPR) and Concurrent Packet Transmission (CPT) due to their half-duplex operation [1]. In other words, an MBA-equipped node cannot transmit signals in some beams and receive signals in other beams at the same time. At a given time, an MBA-equipped node has all its beams operate in either transmission or reception mode (see Fig. 1).

MBAs can be implemented either in the form of Multiple Fixed-Beam directional Antennas (MFBAs) or in the form of Multi-Channel Smart Antennas (MCSAs). To form multiple fixed beams, MFBAs and multiple radios (MRs) with a directional antenna equipped in each radio (transceiver) can be exploited [2,3]. As a result, high network throughput can be achieved. In a stationary environment, the antenna patterns can be optimized to further improve network performance. However, the performance of MFBAs/MRs degrades in a time-varying

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 143–155, 2018.
https://doi.org/10.1007/978-3-319-74439-1_13



(a) Omni mode (b) Tx mode (c) Rx mode

Fig. 1. Multi-beam antenna modes [1]

multipath propagation environment, which is typically experienced in indoor and low-altitude outdoor wireless networks [4]. The alternative approach to implement MBAs is to use MCSAs [5–7]. By using smart antenna techniques, multiple beams can be adaptively and dynamically formed by a node so as to provide robust communication links with multiple users. At the expense of higher complexity, an MCSA-based approach shows the same advantages as the MFBA/MR implementation, but its performance does not degrade in time-varying multipath environments [6,7].

In the literature, the work around MBAs has focused on designing MAC and/or routing protocols that exploit spatial reuse in order to increase network performances such as throughput and packet delivery ratio. Not much has been done to use the full MPT/MPR potential for end-to-end delay reduction. Moreover, a great deal of existing work on MBAs is done for infrastructure-based networks with an applicability to ad hoc networks that is yet to be clarified. Furthermore, some link scheduling proposals for network performance improvement do not consider MBAs at all.

In this paper, we make the case that the full potential of the MPT/MPR capability of MBAs can be unlocked to drastically reduce the end-to-end delay in ad hoc networks. We define a formal optimization model for delay reduction, and we observe that the optimal end-to-end delay is attained when links are scheduled in such way that opportunities for MPT/MPR are maximized. This results in a selection of routes that, up to half of the times, are not the shortest. We actually show that using only the shortest routes, a widespread criterion in traditional routing protocols for ad hoc networks, results in higher delays. To the best of our knowledge, no such analysis of the potential of MBAs to reduce/minimize the delay has been conducted thus far for ad hoc networks.

The remainder of this paper is organized as follows. In Sect. 2, a literature review on link scheduling and the utilization of MBAs for the improvement of network performances is presented. Section 3 lays out our optimization model. Simulation results are discussed in Sect. 4, and concluding remarks are provided in Sect. 5.

2 Literature Review on MBAs and Link Scheduling

In [8], Cheng et al. show that the shortest path does not always lead to the minimum delay. End-to-end delay being a result of both the number of hops on the path and the interference level along the path, the shortest path leads to the minimum delay only if the shortest path is the least interfered path. The authors propose a linear programming-based link scheduling scheme that computes timeslot assignments in order to minimize the end-to-end delay without causing conflicting transmissions. The use of MBAs is not considered.

In [9], Wang et al. propose a CSMA/CA-based uplink MAC protocol for wireless LANs with MBA access-points. Spatial reuse is utilized by allowing as many parallel uplink data transmissions as possible in order to improve the throughput. Since all the nodes including the access point run a CSMA/CA-based MAC protocol, the authors claim that the proposed protocol is not limited to the single-hop case, but can be easily extended to multi-hop ad hoc networks. In [10], the same authors go further to present an analytical model to evaluate the performance of multi-beam wireless LANs. The beam-synchronization problem, the beam-overlapping problem, and the mobility issue are also addressed.

Jain et al. [11] present a detailed study of MBAs MAC issues. They propose a cross-layer Hybrid MAC (HMAC) protocol that can leverage the benefits of MBAs in multi-hop networks. Using extensive topological and traffic patterns, the authors demonstrate that employing MBAs and HMAC can result in significant performance improvements in terms of both aggregate throughput and average end-to-end packet delay. In most of the sample topologies, HMAC delivers near-optimal performance. From a study of random topological scenarios, the authors also conclude that both single and multi-beam antennas deliver comparable performance in ad hoc scenarios. But these claims are not based on any formal optimization model of their metrics.

In [12], Tang et al. propose a MAC protocol for WLANs with MBA-equipped access points, omnidirectional-antenna mobile nodes and a single frequency channel. The protocol addresses a series of challenging problems such as the beam-load unbalance problem, the unnecessary defer problem, the receiver blocking problem, the antenna-imperfection problem, and mobility. By addressing these issues, as many parallel transmissions between terminals and the AP as possible are successfully facilitated and the performance (throughput) of the network is improved. There is no indication of how this protocol would fare in ad hoc networks.

Wang and Garcia-Luna-Aceves [13] present an approach that takes advantage of the MPR capability of MBAs to reduce the negative effects of multiple access interference and therefore increase the capacity of an ad hoc network. The MPT capability is not considered, nor is the end-to-end delay performance metric. They formulate an optimization problem under a deterministic model and seek to maximize the aggregate network throughput. They then propose a polynomial-time heuristic algorithm aimed at approximating the optimal solution to the joint routing and channel access problem under MPR. This is a methodology that is similar to our work presented in this paper.

3 Scheduling Problem Formulation and Optimization Model

We assume the following conditions:

- there is a perfect time synchronization for all the nodes in the network,
- the nodes run a perfect TDMA-based MAC protocol for MBA antennas,
- time is divided in timeslots,
- multiple nodes can transmit in the same timeslot if their transmissions do not interfere with one another,
- nodes operate in half-duplex: a given node cannot transmit and receive at the same time.

For a given static network with multiple flows, and provided that the antennas are MPT/MPR capable, we would like to know the route selection (link scheduling) that gives us the lowest end-to-end delay (average) possible. This is an optimization problem. We represent the network with a directed connectivity graph $G(V, E)$, where V represents the set of nodes (or vertices) in the network, and E is the set of directed links (or edges). If node i is within the reception range of node j, then Links (i, j) and (j, i) are members of E. We assume that a node can receive up to M simultaneous packets (MPR capability), or perform up to M simultaneous transmissions (MPT capability) at a time, provided that each reception/transmission occurs at a different antenna beam/sector. Therefore, M is also the number of beams for each antenna.

The inputs of our problem are:

- the connectivity graph $G(V, E)$,
- the MPR/MPT capability, M,
- the number of traffic flows $|F|$ that is the cardinality of the set F containing the numbers that represent the flows.
  $F = \{1, 2, ..., |F|\}$,
- the source and destination of each flow. s_f and d_f are respectively the source and destination of Flow f, with $f \in F$,

The outputs are:

- the selected route for each flow, presented as a sequence of links,
- the delay (in timeslots) of each route. The average delay per route is then the sum of all delays divided by the number of flows.

In order to solve this optimization problem, we define a decision variable α_{ij}^{fk} that tells us whether or not Link (i, j) of Flow f is scheduled at Timeslot k. $\alpha_{ij}^{fk} \in [0, 1]$. T is the set of timeslots. $T = \{1, 2, ..., |E| \times |F|\}$, where $|E|$ is the cardinality of E. The T set is constructed assuming that each flow uses all the available links (edges) and each link from each flow in the network gets its own timeslot and there are no concurrent transmissions (worst case scenario). The problem can be formulated as follows:

minimize

$$\sum_{f \in F} \sum_{k \in T} \sum_{i \in V} \sum_{d_f} \alpha_{id_f}^{fk} \times k \tag{1}$$

subject to the following 6 constraints:

1. Flow Circulation Constraints

$$\sum_{j:(s_f,j) \in E} \quad \sum_{k \in T} \alpha_{s_f j}^{fk} = 1 \qquad \forall f \in F \tag{2}$$

$$\sum_{i:(i,d_f) \in E} \quad \sum_{k \in T} \alpha_{id_f}^{fk} = 1 \qquad \forall f \in F \tag{3}$$

$$\alpha_{ij}^{fk} - \sum_{t=1}^{k-1} \quad \sum_{l:(l,i) \in E} \alpha_{li}^{ft} \leq 0 \forall f \in F, \forall k \in T, \forall i \in V - \{s_f\}, \forall j \in V | (i,j) \in E \tag{4}$$

2. Flow Consistency Constraints

$$\sum_{l:(i,l) \in E} \alpha_{il}^{fk} \leq 1 \qquad \forall f \in F, \forall k \in T, \forall i \in V \tag{5}$$

$$\sum_{l:(l,j) \in E} \alpha_{lj}^{fk} \leq 1 \qquad \forall f \in F, \forall k \in T, \forall j \in V \tag{6}$$

$$\sum_{k \in T} \alpha_{ij}^{fk} \leq 1 \qquad \forall f \in F, \forall (i,j) \in E \tag{7}$$

3. Link Occupancy Constraint

$$\sum_{f \in F} \alpha_{ij}^{fk} \leq 1 \qquad \forall k \in T, \forall (i,j) \in E \tag{8}$$

4. Half-duplex Constraint

$$\sum_{f \in F} (\alpha_{ij}^{fk} + \alpha_{jl}^{fk}) \leq 1 \qquad \forall k \in T, \forall i,j,l \in V \tag{9}$$

5. MPT Capability Constraint

$$\sum_{l:(i,l) \in E} \quad \sum_{f \in F} \alpha_{il}^{fk} \leq M \qquad \forall k \in T, \forall i \in V \tag{10}$$

6. MPR Capability Constraint

$$\sum_{l:(l,j) \in E} \quad \sum_{f \in F} \alpha_{lj}^{fk} \leq M \qquad \forall k \in T, \forall j \in V \tag{11}$$

Concerning the objective function at Eq. 1, the goal is to minimize the average number of timeslots per flow required to complete all the flows. This is equivalent to finding routes where the last link (link to the destination node) occurs the earliest possible, such that the average end-to-end delay is minimal. To that effect, k in the function represents the cost for each timeslot. The constraints are explained as follows:

1. Flow Circulation Constraints
   - Constraint 2: a flow always starts, meaning the source node should be scheduled exactly once.
   - Constraint 3: the destination of a flow must always be reached, by exactly one link.
   - Constraint 4: Link (i, j) can be scheduled for Flow f only if i has previously received a packet before; meaning there exist a completed (l, i) link from the same flow. Except if i is the source of the flow.
2. Flow Consistency Constraints
   - Constraint 5: the same flow cannot have multiple next hops at any given intermediate node belonging to the route.
   - Constraint 6: the same flow cannot be received from multiple previous hops at any given intermediate node belonging to the route.
   - Constraint 7: a link can be scheduled only once at most, for any given flow.
3. Link Occupancy Constraint
   - Constraint 8: only one flow is permitted on a link at a time.
4. Half-duplex Constraint
   - Constraint 9: transmission and reception cannot occur at same time at any given node.
5. MPT Capability Constraint
   - Constraint 10: up to M outgoing links can be scheduled at a node at any timeslot.
6. MPR Capability Constraint
   - Constraint 11: up to M incoming links can be scheduled at a node at any timeslot.

4 Simulation and Analysis

We solved the foregoing optimization model using the Optimization Programming Language (OPL) with IBM-ILOG-CPLEX Optimization Studio 12.7.1 [14], with all the parameters set to their default values. We considered two topologies, Topology 1 and Topology 2.

4.1 Topology 1

Topology 1 (Fig. 2) is a static grid topology of 16 nodes. The grid is of size $3D \times 3D$, and the transmission range of the nodes is R, with $R = D\sqrt{2}$. We set M such that any node can simultaneously communicate with all its neighbors. Given our topology, the maximum number of neighbors that a node can have is 8; Let us consider eight traffic flows, Flow 1 through Flow 8, as follows: $1 \longrightarrow 15$, $3 \longrightarrow 13$, $5 \longrightarrow 12$, $9 \longrightarrow 8$, $15 \longrightarrow 2$, $13 \longrightarrow 4$, $12 \longrightarrow 1$, and $8 \longrightarrow 5$.

Table 1 presents the optimal link scheduling of the network as found by the solver. We can obtain from the table that the optimal average end-to-end delay is 3.25 slots per flow. 6 flows are completed in 3 slots and 2 flows are completed in 4 slots, thus the average of 3.25 slots. All the 8 routes chosen are also the

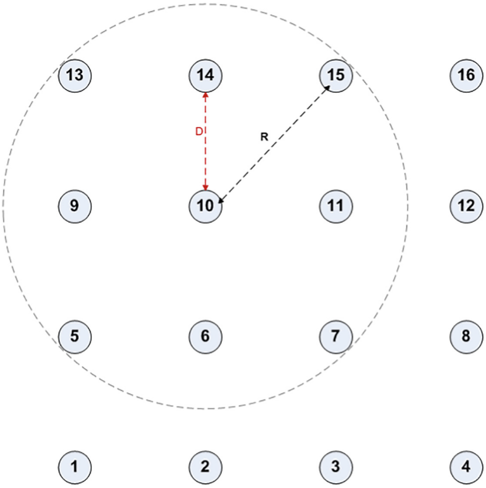


Fig. 2. Topology 1

Table 1. Topology 1, optimal link scheduling with 8 flows

Traffic flow	Slot 1	Slot 2	Slot 3	Slot 4
$1 \longrightarrow 15$	(1,6)	(6,11)	(11,15)	
$3 \longrightarrow 13$	(3,6)	(6,9)	(9,13)	
$5 \longrightarrow 12$	(5,6)	(6,7)	(7,12)	
$9 \longrightarrow 8$	(9,6)	(6,3)	(3,8)	
$15 \longrightarrow 2$	(15,12)	(12,7)	(7,2)	
$13 \longrightarrow 4$	(13,10)	(10,7)	(7,4)	
$12 \longrightarrow 1$		(12,11)	(11,6)	(6,1)
$8 \longrightarrow 5$		(8,3)	(3,6)	(6,5)

shortest (3 hops). Note that, given the regularity of our grid topology, each flow has many possibilities for its shortest route.

The optimal (delay-wise optimality) routes selected form a lot of *star nodes* in the network. We define *star nodes* as nodes that use their MPR capability at one timeslot before using their MPT capability at the following timeslot. With a star node, packets from distinct traffic flows travel two hops in two consecutive timeslots, which is obviously a desirable effect for the minimization of the delay. From Table 1, we can see that one star node, namely Node 6, is formed between Slot 1 and Slot 2. In effect, Flows 1 through 4 all send a packet to Node 6 at Slot 1 from different previous nodes (Nodes 1, 3, 5, and 9 respectively) and all 4 packets leave Node 6 for different next hops (Nodes 11, 9, 7 and 3 respectively). Similarly, three star nodes (Nodes 11, 7, and 3) are formed between Slot 2 and Slot 3. Finally, between Slot 3 and Slot 4, one star node (Node 6) is formed.

We note that we do not have any *bridges* formed in the network with the optimal scheduling. We define a *bridge* as a beam (equivalent to a link in our configuration) that routes two or more different flows that have arrived at a given node at the same timeslot. For example, as already mentioned, Node 6 receives four different flows from four different links (beams, as we assume that each neighbor lies in its own beam, no two neighbors share a beam) during Slot 1. It then forwards the four different packets using four different links (beams). For a bridge to be formed, we would have had less than four outgoing links at Node 6 at Slot 2, with the missing link being already scheduled for a different flow. It would have therefore waited for a subsequent slot. For example, at Slot 2 we have the following outgoing links from Node 6: $(6, 11)$, $(6, 9)$, $(6, 7)$, and $(6, 3)$. Had we had a bridge on the beam that covers Link $(6, 7)$ for instance, the four outgoing links from Node 6 would had been: $(6, 11)$, $(6, 9)$, $(6, 7)$, and $(6, 7)$, with Link $(6, 7)$ appearing twice for the forwarding of two different flows (Flow 3 and Flow 4) at the same timeslot. In this case, only three outgoing links would had been scheduled at Slot 2: $(6, 11)$, $(6, 9)$, and $(6, 7)$. The packet from Flow 4 that is supposed to go through Link $(6, 7)$ would have had to be scheduled at a later slot, since two packets cannot be transmitted at the same time on the same beam (corresponding to the same link in our configuration). Note that we do not have to have a duplicate link in order to have a bridge. Two links can be different (obviously still sharing the same origin, such as $(6, 7)$ and an hypothetic $(6, X)$), but as long as they are serviced by the same beam and compete for scheduling at the same time, a bridge is formed. As we can see from the table, no bridge is formed on any of the nodes that receive multiple flows at once at given timeslots (Node 6 at Slot 1, Nodes 11, 7, 3 at Slot 2, and Node 6 again at Slot 3).

When MBAs are not used ($M = 1$), the performance (Table 2) degrades as follows. Only two flows (Flow 4 and Flow 7) are completed in 3 slots, five flows are completed in 4 slots, and one flow needs 6 slots to complete. This gives an average of 4 slots per flow, hence a degradation of 0.75 slot compared to when MBAs are used. We also notice that three of the eight flows do not use their shortest path (3 hops), and use a longer (4 hops) path instead. This tells us that in some cases, using longer paths can improve the overall delay. Note that

Table 2. Topology 1, optimal link scheduling with 8 flows, no MBA

Traffic flow	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
1⟶ 15		(1,5)	(5,10)	(10,15)		
3⟶ 13	(3,6)	(6,9)		(9,13)		
5⟶ 12	(5,10)	(10,15)	(15,16)	(16,12)		
9⟶ 8	(9,14)	(14,11)	(11,8)			
15⟶ 2	(15,16)	(16,12)	(12,7)	(7,2)		
13⟶ 4			(13,14)	(14,11)	(11,7)	(7,4)
12⟶ 1	(12,7)	(7,2)	(2,1)			
8⟶ 5		(8,3)	(3,6)	(6,5)		

in any scenario, the scheduling of a given link is prohibited at a given timeslot for two reasons: the non-availability of the MPT/MPR capability and/or the half-duplex constraint.

4.2 Topology 2

Topology 2 (Fig. 3) is obtained by constraining/altering Topology 1 significantly. Nodes 5, 8, 9, and 12 are suppressed, resulting in a 12-node topology. In this

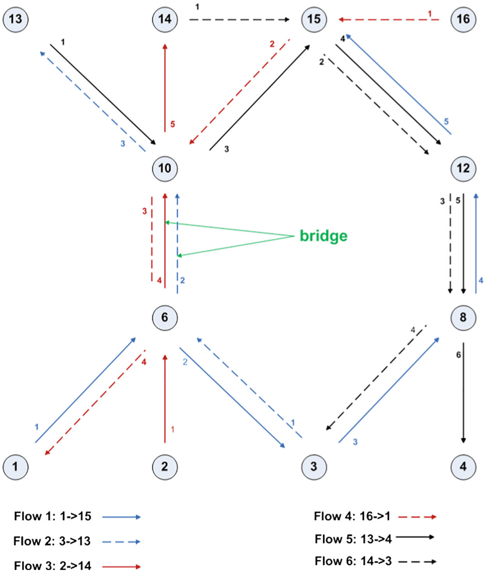


Fig. 3. Topology 2

network we have 6 traffic flows, Flow 1 through Flow 6, as follows: 1 ⟶ 15, 3 ⟶ 13, 2 ⟶ 14, 16 ⟶ 1, 13 ⟶ 4, and 14 ⟶ 3.

The optimal link scheduling for Topology 2 is presented in Table 3 and Fig. 3. It can be observed that the optimal average end-to-end delay is 4.5 slots per flow.

From Table 3, we can see that five star nodes are formed as well as one bridge. Node 6 is a star node for Flows 1 and 2 between Slots 1 and 2. However, Node 6 also forms a bridge for Flows 2 and 3 at Slot 2. In effect, packets from Flows 2 and 3 both arrive at Node 6 at Slot 1, and both packets (from two different flows) are scheduled to leave Node 6 using the same link $(6, 10)$, thus the same antenna beam: this is a bridge. Consequently, one of the flows has to wait until a subsequent slot. As we can see, Link $(6, 10)$ of Flow 3 is rescheduled for Slot 4. It cannot be rescheduled for Slot 2 because it is already scheduled for Flow 2 at that slot, and it cannot be rescheduled at Slot 3 either because of the half-duplex constraint (Node 6 that is supposed to be the transmitter is already at the receiving end of Link $(10, 6)$ of Flow 4). Table 3 gives us the optimal solution, which means that this bridge could not be avoided. Had this bridge been avoided at all cost, it would have resulted in a higher delay.

Only half the routes chosen are also the shortest (Flows 2, 3, and 4). For Flow 1, a 5-hop route is chosen, marking a 2-hop increase from the shortest route. Similarly, for Flow 5, the 5-hop route chosen is 1 hop longer than the shortest route between Node 13 and Node 4 (which is 4 hops with this new topology, unlike Topology 1). For Flow 6, the chosen route is 1 hop longer than

Table 3. Topology 2, optimal link scheduling with 6 flows

Traffic flow	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
1⟶ 15	(1,6)	(6,3)	(3,8)	(8,12)	(12,15)	
3⟶ 13	(3,6)	(6,10)	(10,13)			
2⟶ 14	(2,6)			(6,10)	(10,14)	
16⟶ 1	(16,15)	(15,10)	(10,6)	(6,1)		
13⟶ 4	(13,10)		(10,15)	(15,12)	(12,8)	(8,4)
14⟶ 3	(14,15)	(15,12)	(12,8)	(8,3)		

Table 4. Topology 2, optimal link scheduling with 6 flows, no MBA

Traffic flow	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
1⟶ 15			(1,6)	(6,10)	(10,15)		
3⟶ 13	(3,6)	(6,10)	(10,13)				
2⟶ 14					(2,6)	(6,10)	(10,14)
16⟶ 1	(16,12)	(12,8)	(8,3)	(3,2)		(2,1)	
13⟶ 4		(13,14)	(14,15)	(15,16)	(16,12)	(12,8)	(8,4)
14⟶ 3	(14,15)	(15,16)	(16,12)	(12,8)	(8,3)		

Table 5. Topology 2, link scheduling with 6 flows, shortest path

Flow	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8	Slot 9	Slot 10
1→15	(1,6)	(6,10)			(10,15)					
3→13	(3,6)		(6,10)		(10,13)					
2→14	(2,6)			(6,10)	(10,14)					
16→1	(16,15)	(15,10)					(10,6)	(6,1)		
13→4	(13,10)				(10,6)			(6,3)		(3,4)
14→3	(14,10)					(10,6)			(6,3)	

the shortest route. We therefore observe that, even with MPT/MPR enabled, longer routes than the shortest are often preferred in order to attain the minimal end-to-end delay.

The number of transmissions performed is equal to the number of links scheduled. With the optimal link scheduling, we have a total of 24 transmissions.

When MBAs are not used ($M = 1$), the performance (Table 4) degrades as follows. Two flows are completed in 7 slots, one flow is completed in 6 slots, two flows are completed in 5 slots, and one flow needs 3 slots to complete. This gives an average of 5.5 slots per flow. This is a degradation of 1 slot per route compared to when MBA are used. Here too, only half the routes chosen are also the shortest (Flows 1, 2, and 3). We see that with this topology also, using longer paths does improve the overall delay. With no MPT/MPR capability, the number of transmissions for the optimal scheduling is 25, a mere 4% increase from the case with MPT/MPR.

With Topology 2, unlike with Topology 1, there is only one possible shortest path for each flow; therefore we can also quantify the cost (in terms of delay) of using the shortest path here. The scheduling in Table 5 shows that if flows are restricted to their shortest route we have a considerable degradation in delay, even if MBAs are used. In effect, we can deduct from the table an average delay of 7 slots per flow, which is a degradation of 2.5 slots from the optimal solution described above (4.5 slots). Therefore, the latter is a 36% decrease in delay compared to the former. Moreover, using only shortest routes with MBAs also shows a 1.5 slot degradation from the optimal solution without MBAs. We therefore observe that limiting routes to the shortest paths, even while using MBAs, is detrimental to the delay to the point that having a "smarter" choice of paths (some paths being longer than the shortest) without MBAs is better. The number of transmissions here is equal to 20, a 16% decrease from the optimal scheduling presented earlier.

5 Conclusions

In this paper, we have shown that MBAs need to be used for delay reduction in ad hoc networks. Moreover, with static deterministic topologies with multi-flow scenarios, we have shown that in order to exploit the full potential of MBAs for that end-to-end delay reduction, the selection of routes needs to include other criteria such as the promotion of star nodes and the minimization of the number of bridges. We have shown that the mere use of shortest routes, as it is the case in most existing routing protocols for ad hoc networks, results in relatively high delays because such paths usually result in the formation of bridges in multi-flow scenarios; and bridges incur waiting and rescheduling delays that add to the end-to-end delay. We formulated determining the best route selection (best link scheduling) as an optimization problem that we solved using linear programming. Our scenarios showed that the optimal link scheduling forms a lot of star nodes and eliminates bridges as much as possible for a considerable 36% reduction in end-to-end delay. Our optimization model finds the lower bound of the end-to-end delay. However, this huge reduction in delay comes at the expense of a 16% increase in overhead measured by the total number of transmissions.

As future work, we will include beamforming into the model. Furthermore, we will design a protocol that establishes routes in light of what we have learnt here, and is able to deliver near-optimal results for end-to-end delay.

References

1. Bao, K., Hu, F., Bentley, E., Kumar, S.: Diamond-shaped mesh network routing with cross-layer design to explore the benefits of multi-beam smart antennas. In: 2016 25th International Conference on Computer Communication and Networks (ICCCN), pp. 1–5 (2016)
2. Bahl, P., Adya, A., Padhye, J., Walman, A.: Reconsidering wireless systems with multiple radios. SIGCOMM Comput. Commun. Rev. **34**(5), 39–46 (2004). https://doi.org/10.1145/1039111.1039122
3. Draves, R., Padhye, J., Zill, B.: Routing in multi-radio, multi-hop wireless mesh networks. In: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, MobiCom 2004, pp. 114–128. ACM, New York (2004). https://doi.org/10.1145/1023720.1023732
4. Winters, J.H.: Smart antenna techniques and their application to wireless ad hoc networks. IEEE Wirel. Commun. **13**(4), 77–83 (2006)
5. Singh, A., Ramanathan, P., Veen, B.V.: Spatial reuse through adaptive interference cancellation in multi-antenna wireless networks. In: 2005 IEEE Global Telecommunications Conference, GLOBECOM 2005, vol. 5, pp. 3092-3096, 5 pp. (2005)
6. Zhang, Y., Li, X., Amin, M.G.: Multi-channel smart antennas in wireless networks. In: 2006 Fortieth Asilomar Conference on Signals, Systems and Computers, pp. 305–309 (2006)
7. Li, X., Zhang, Y., Amin, M.G.: Performance evaluation of wireless networks exploiting multi-beam antennas in multipath environments. In: 2007 International Waveform Diversity and Design Conference, pp. 188–192 (2007)
8. Cheng, M.X., Gong, X., Xu, Y., Cai, L.: Link activity scheduling for minimum end-to-end latency in multihop wireless sensor networks. In: 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011, pp. 1–5 (2011)
9. Wang, J., Fang, Y., Wu, D.: Uplink medium access control for WLANs with multi-beam access point. In: 2005 IEEE Global Telecommunications Conference, GLOBECOM 2005, vol. 5, pp. 3012–3016, 5 pp. (2005)

10. Wang, J., Fang, Y., Wu, D.: Enhancing the performance of medium access control for wlans with multi-beam access point. IEEE Trans. Wirel. Commun. **6**(2), 556–565 (2007)
11. Jain, V., Gupta, A., Agrawal, D.P.: On-demand medium access in multihop wireless networks with multiple beam smart antennas. IEEE Trans. Parallel Distrib. Syst. **19**(4), 489–502 (2008)
12. Tang, Z., Xing, X., Jiang, F.: Providing balanced and enhanced transmission for WLANs with multi-beam access point. In: 6th Annual Communication Networks and Services Research Conference (CNSR 2008), pp. 242–248 (2008)
13. Wang, X., Garcia-Luna-Aceves, J.J.: Embracing interference in ad hoc networks using joint routing and scheduling with multiple packet reception. In: IEEE INFOCOM 2008 - The 27th Conference on Computer Communications (2008)
14. IBM ILOG CPLEX optimization studio. https://www.ibm.com/jm-en/marketplace/ibm-ilog-cplex

Trajectory and Buffer Aware Message Forwarding for Multiple Cooperating UAVs in Message Ferry Networks

Mehdi Harounabadi(✉) and Andreas Mitschele-Thiel

Integrated Communication Systems Group, Ilmenau University of Technology,
Ilmenau, Germany
{mehdi.harounabadi,mitsch}@tu-ilmenau.de

Abstract. This paper presents a Trajectory and Buffer Aware message Forwarding (TaBAF) scheme in message ferry networks with multiple mobility-controlled UAVs. UAVs are basically message ferries with an on-the-fly mobility decision maker to deliver messages between isolated nodes. Besides, UAVs forward messages opportunistically when they visit each other on-the-air. UAVs have only local observation in our network model and do a signaling with each other when they fly into the radio transmission range of each other. The signaling information are the next node that a UAV will visit and its buffer state. TaBAF applies this information in its message forwarding decision and forwards a message to a neighbor UAV if it can deliver the message earlier. The results show that the TaBAF in message ferry networks outperforms pure message ferry approaches and existing message forwarding schemes in terms of end to end message latency. We showed that the TaBAF decreases the average traveling delay of messages in UAVs and this is the reason for its performance improvement. Moreover, TaBAF decreases average flied distance of each UAV in the network by efficient message forwarding.

Keywords: Trajectory aware · Message forwarding · UAV
Message ferry

1 Introduction

In disaster scenarios or sparse sensor networks, message delivery is a challenging task due to disconnections in the network and isolation of wireless nodes. Delay Tolerant Network (DTN) routing approaches can be applied for message delivery in such scenarios. However, if nodes are stationary, messages cannot be delivered to their destinations. Unmanned Aerial Vehicles (UAVs) can be employed in such networks as message ferry nodes to make data communication possible. A UAV, as a message ferry, is responsible to travel among disconnected nodes and deliver their messages.

One of the main properties of a UAV that distinguishes it from other types of wireless nodes is its mobility that be controlled. The flight path of a UAV

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 156–165, 2018.
https://doi.org/10.1007/978-3-319-74439-1_14

can be planned in advanced (offline path planning) or the UAV can decide it on-the-fly and autonomously. Controlling the mobility of a UAV in a message ferry network improves dramatically the performance of message delivery [1]. Moreover, by employing multiple cooperating UAVs, the latency of message delivery is decreased more [2]. Some challenges are emerged in multi UAV networks such as the coordination of UAVs and collision avoidance between them [4], but UAVs can build an ad hoc network on-the-air and cooperate in message delivery with multi hop communication. In this case, a UAV is not only a message ferry but also it acts as a router to forward messages.

In this paper a Trajectory and Buffer Aware message Forwarding (TaBAF) scheme for multiple cooperating UAVs in message ferry networks is proposed where UAVs act basically as message ferries but they forward messages opportunistically to each other to accelerate message delivery in the network. UAVs decide about their mobility on-the-fly based on their local observations from the state of network [2] and are coordinated applying stigmergy [3]. To cooperate in message delivery, UAVs do a signaling on-the-air when they fly close to each other. The signaling information are the next node that a UAV is flying towards it and the state of its message buffer. The received signaling information from neighbor UAVs are applied in a UAV for its message forwarding decision. TaBAF in a UAV forwards a message to a neighbor if the neighbor UAV can deliver the message earlier.

The results show that TaBAF in message ferry networks with multiple UAVs outperforms pure message ferry approaches which UAVs do not forward messages and existing message forwarding schemes. Trajectory and buffer aware message forwarding between UAVs decreases end to end message delivery latency in a network by decreasing message traveling delay in buffer of UAVs. Moreover, TaBAF decreases the average flied distance for each UAV in the network by efficient message forwarding between UAVs. Average flied distance is the cost for each UAV. To the best of our knowledge, this is the first work that applies trajectory and buffer aware message forwarding in multi UAV based message ferry networks.

The remainder of this paper is organized as follows: we discuss existing work in Sect. 2. In Sect. 3, the network model is described. The on-the-fly next node decision maker in UAVs is presented in Sect. 4. Section 5 introduces signaling between UAVs and nodes. In Sect. 6, we present our trajectory and buffer aware message forwarding scheme. Simulation study and performance evaluations are in Sect. 7. Finally, we conclude the paper in Sect. 8.

2 Related Work

The message ferrying approach was proposed in [5] to deliver messages in disconnected networks. They controlled the mobility a ferry node by an offline path planning. The path planning was same as the solution of Traveling Salesman Problem (TSP) which the main objective is to visit all nodes in the shortest path. [7] is another single ferry approach where a ferry node has a full observation over the network. In [1], authors proposed an on-the-fly decision maker in a ferry with local observations. They showed that an on-the-fly decision making outperforms offline path planning approaches in networks with asymmetric traffic load.

The latency of message delivery increases in large and highly loaded networks with a single ferry. To overcome the limitation of single ferry networks, several architectures for multi ferry networks was proposed in [8]. Ferries mobility decision, coordination and cooperation of ferries were out of the scope of this work. In [9], authors proposed a multi ferry approach with local observations in ferries. The mobility decision in ferries was based on the mobility of nodes. An on-the-fly decision making for multi ferry networks was proposed in [2] that ferries decide the next node to visit on-the-fly after visiting a node and exchanging data and control messages. Ferries have only local observations in their assumptions. Therefore, the coordination of ferries was done applying a stigmergic communication among ferries by leaving traces in nodes. Exiting multi ferry works are pure message ferry networks where the possibility of message forwarding among ferries is neglected.

As mentioned in Introduction, UAVs are mobility controlled wireless nodes which can be considered as a realization of message ferries. In multi ferry networks, message forwarding among UAVs is possible and UAVs are not only message ferries but also routers that can forward messages.

A hybrid DTN/geographical routing approach for multi UAV networks has been proposed in [10] that the message forwarding decision is made based on an estimation about the future location of UAVs. They apply the speed and mobility direction of UAVs in their estimations. Moreover, they assume a long range communication in UAVs that provides a full observation of the network in each UAV. Their message forwarding scheme cannot be applied in networks with local observations in UAVs. The authors in [11] also proposed a geographical routing protocol that estimates a UAV link life time using the mobility direction and speed of a UAV. The mobility of UAVs are not controlled in this work and UAVs are used as opportunistic relays.

Existing work are either pure message ferry networks without any message forwarding between ferries or consider a full observation in UAVs. A message forwarding scheme for multiple cooperating UAVs with an on-the-fly mobility decision making which each UAV has only a local observation in message ferry networks is required.

3 Network Model

3.1 Assumptions

In our network model, wireless nodes (N) are of two types; regular nodes ($R \subset N$) and UAVs that can act as data ferries or routers ($F \subset N$). From now on, we call regular nodes only 'nodes'. Nodes are stationary and isolated. Therefore, no direct communication between any pair of nodes is possible. The location of nodes is given to UAVs. Nodes are producer (generator) and consumer (receiver) of messages. They generate messages with a variable rate. UAVs are wireless nodes that act basically as message ferries with controlled mobility. They can forward messages between each other, either. UAVs only carry messages or forward them and do not generate any message. UAVs always travel with a constant velocity. Moreover, we assume an unlimited buffer size in UAVs and nodes.

To model opportunistic visits of UAVs on-the-air for message forwarding, we assume a constant radio transmission range for UAVs such that they can communicate if they come into the range of each other. Moreover, the required time for a UAV to travel among nodes is much longer than the required message transmission time (T_{tx}) between wireless nodes. Thus, we neglect T_{tx}.

$$T_{travel}(i, j \in R) \gg T_{tx} \tag{1}$$

In our network model, there is no direct communication among all UAVs while their radio transmission range is limited. UAVs can only communicate when they are in the radio transmission range of each other. Therefore, A UAV has no global knowledge about the network and can only observe the network locally when it visits a node or another UAV. During a UAV visit to a node or another UAV, several steps occurs sequentially that will be described in the next section.

3.2 Steps of a UAV Visit

In our multi UAV based message ferry network, UAVs travel between nodes and when they visit a node or another UAV several steps are triggered and run as follows:

1. Exchange control information (signaling)
2. Exchange data messages
   (a) If the UAV visits a node, it collects all messages from the node's buffer and delivers all messages for which the current node is the destination
   (b) If the UAV visits another UAV, it decides to forward all or part of its buffered messages based on the received signaling information
3. Decide the next node to visit using the on-the-fly decision maker in the UAV
4. Travel to the next (decided) node

In the next sections, we describe steps in more details.

4 On-the-Fly Next Node Decision Maker in UAVs

To control the mobility of UAVs, an on-the-fly decision maker similar to [2] is applied. The main goal of the decision maker is to make on-the-fly decisions in a UAV about the next node to visit. The decision maker works only based on the local observations of a UAV and the history of nodes that a UAV keeps in its memory. Each UAV applies a $Score$ function for its mobility decision. A score is calculated in a UAV for each node r and a node with the maximum $Score(r)$ value is selected as the next node to visit. The $Score$ for each node r is calculated as follows:

$$Score(r) = \frac{mb(r) + hist(r)}{d(c, r)} \tag{2}$$

where $mb(r)$ is a function that returns a normalized value based on the number of waiting messages in the UAV buffer for the destination r. The second function is based on the history of nodes in the UAV. Each UAV keeps the history of its last visit time to all nodes and applies it in its decision maker. $hist(r)$ returns a normalized value for the node r based on the last visit time of a UAV to the node r. The function $hist(r)$ returns a bigger value for a node r which has been visited a long time ago than a node which has been visited recently. This function avoids any visit starvation in nodes and frequent visits of the node in a short time window. $d(c, r)$ in $Score$ function is the distance between the current node c that the UAV is visiting and a candidate next node r.

5 Exchange of Control Information

In the proposed multi UAV based message ferry network in this paper, a UAV signals some control information with nodes or other UAVs in the network when it visits them. Signaling information are different in a visit of a UAV to a node or to another UAV. The signaling information are applied later in the mobility decision maker of the UAV and its message forwarding decision.

5.1 Signaling Between UAV and Node for Stigmergic Coordination of UAVs

As mentioned in Sect. 4, to avoid frequent visits of UAVs to a node and visit starvation in nodes, each UAV keeps a history of its last visit time to all nodes. Same as UAVs, each node keeps a history table that contains the last visit time of all other nodes. As the nodes are stationary in our network, the history information is about the visits of UAVs to nodes. During a UAV visit to a node, the UAV and the node exchange their last visit time history table and update their tables with more up-to-date information.

As UAVs have not a long range communication in our network, they cannot signal this information directly to each other. Therefore, UAVs are coordinated using a stigmergic communication in a form of an indirect signaling. Indirect signaling among UAVs is formed when each UAV does a signaling with nodes. In other words, a UAV leaves traces in nodes (environment) and take existing traces from them. A node acts as a relay for UAVs to exchange their control information.

5.2 Signaling Between UAVs

Another type of signaling occurs in a UAV visit to another UAV on-the-air when two UAVs fly into the radio transmission range of each other. There are two types of information in this signaling that are applied in the message forwarding decision of UAVs and are as follows:

1. The mobility decision of UAV: It is the output of the on-the-fly next node decision maker in the UAV.
2. State of message buffer in UAV: It is the number of buffered messages for each destination in a UAV and reflects future nodes that the UAV will visit.

6 Trajectory and Buffer Aware Message Forwarding

In our network, multiple UAVs are employed as message ferries to deliver messages between disconnected nodes. Besides, there is a possibility for message forwarding between UAVs if they visit each other on-the-air. Therefore, UAVs are not only message ferries but also flying wireless routers. Different metrics can be applied in message forwarding decision between two visiting UAVs. In this paper, we propose a trajectory and buffer aware message forwarding scheme that UAVs exploit the signaling information that they exchange with each other (see Sect. 5.2). The message forwarding decision is done in each UAV during an on-the-air visit to a neighbor UAV in two steps as follows:

1. In the first step, each UAV uses the trajectory information of neighbor UAVs that it has received during signaling. The trajectory information is the next node that the on-the-fly decision maker in Sect. 4 has decided for a UAV. The UAV forwards all message for which the neighbor UAV is flying directly towards the destination of messages. In case which both UAVs fly to the same destination, the UAV forwards messages if the neighbor UAV is closer to the destination. After forwarding messages to a neighbor UAV, if the UAV has not any message to deliver to its next visiting node, it runs the on-the-fly next node decision maker and may change its trajectory.
2. In the second step, the UAV applies both trajectory and buffer information of the neighbor UAV to forward more messages. In the first step, the UAV forwards all messages which the neighbor UAV will fly directly to their destination. In this step, the UAV may forward more messages even if the neighbor UAV will not fly directly to the destination of messages but the trajectory and buffer state of the neighbor UAV meet two conditions as follows:
   (a) The first condition is based on the traveling direction of UAVs. The first condition is met, if a UAV is going to fly away from the destination of a message while the neighbor UAV will fly to a node that is closer to the destination of message.
   (b) A UAV forwards a message to a neighbor UAV if the first condition is met and only if the neighbor UAV has messages in its buffer to the same destination of the message. If the both conditions are met for more than one neighbor UAV, the message is forwarded to the neighbor with highest number of buffered messages with the same destination address.

As all UAVs apply the on-the-fly decision maker in Sect. 5 for the next node to visit and the decision maker applies $mb(r)$ and $distance(c, r)$ functions in its decision function, the neighbor UAV will visit the destination of a message earlier than the UAV that has buffered the message if both conditions (a and b) are met.

7 Simulation Study

In this section, we evaluate and study the performance of proposed message forwarding scheme in multi UAV based message ferry networks. To do this, we developed a Python based simulator.

We simulate message ferry networks with 10 nodes where nodes are placed randomly with a uniform distribution. The position of a node is limited to a $1000 \times 1000\,\mathrm{m}^2$ area. Message generation in nodes has a variable rate. It starts at $t = 0$ and runs for 1000 s. Then, the simulation is continued till delivery of all messages. The traffic load in the network is asymmetric. Message generation rate in nodes can be classified into four classes which are very high rate (20% nodes), high rate (20% of nodes), normal rate (50% of nodes) and no message generation (10% of nodes) with mean inter-message arrival time of 1 s, 3 s, 5 s, ∞, respectively. Moreover, UAVs start their travel from different nodes with a constant velocity of 5 m/s and their radio transmission range is assumed 10 m. We run the simulation 10 times for each algorithm. In each run the topology of the network, i.e. the placement of nodes is different. We compare the proposed trajectory and buffer aware message forwarding scheme with a pure message ferrying approach and three other message forwarding schemes in multi UAV based message ferry networks. Different message delivery approaches in our comparisons are as follows:

1. Pure Message Ferry (Pure MF) [2,3]: It is a pure message ferry approach for multi UAV networks where UAVs are only ferries and there is no message forwarding between them.
2. Greedy forwarding (GF) [6]: It is a multi UAV based message ferry network where the message forwarding between UAVs is enabled. A UAV forwards a message to its neighbor UAV if the neighbor is closer to the destination of message.
3. Mobility Direction based Forwarding (MDF) [10,11]: In MDF message forwarding between UAVs in a message ferry network is done based on the mobility direction of UAVs.
4. Trajectory Aware Forwarding (TAF): It is a message forwarding scheme that is proposed in this paper for multi UAV based message ferry networks and will be used in our comparisons. The message forwarding decision in TAF is done only based on the trajectory information of a neighbor UAV that has been received during signaling. Trajectory information is the next node that the UAV will visit. TAF has only the step 1 in message forwarding which was mentioned in Sect. 6.

5. Trajectory and Buffer Aware Forwarding (TaBAF): TaBAF is our proposed message forwarding scheme in this paper which considers trajectory and buffer information of UAVs to forward messages.

The mobility decision in UAVs for all 5 approaches are made based on the on-the-fly next node decision maker in Sect. 4. In Pure MF, there is no singling between UAVs. In GF and MDF, UAVs exchange only their position information and mobility direction. Signaling between UAVs in TAF and TaBAF is same as Sect. 5.2.

Figure 1 shows the end to end latency of messages in five different approaches. The end to end latency of a message refers to the time difference between a message generation in its source and the delivery of message at its destination. The results show that enabling message forwarding between UAVs decreases end to end delay of message delivery in a message ferry network. TaBAF outperforms all approaches. TAF has the closest results to TaBAF, but TaBAF decreases maximum message delay and dispersion of delays. TaBAF forwards a message if the neighbor UAV will deliver the message earlier even if it will not fly directly to the destination of the message. TAF forwards messages only if the neighbor UAV will fly directly to the destination of messages and looses the opportunity of earlier delivery of some messages by neighbor UAVs. MDF and GF are better than Pure MF but median for delays, maximum delay and dispersion of delays are worst than TaBAF and TAF. In GF, a message is forwarded to a neighbor UAV which is closer to the destination of the message but it is flying away from the destination. MDF considers only the mobility direction and a message may be forwarded between several UAVs without considering the final destination of UAVs. MDF applies the short term mobility information of UAVs in its decision which causes inefficient decisions.

The end to end message latency consists of two components. Message waiting delay is the waiting time of a message in a node buffer till its collection by a UAV- $delay_{wait}$ and message traveling delay is the time that a message travels in a UAV after its collection from the source node till delivery to the destination- $delay_{travel}$.

$$Delay_{e2e} = delay_{wait} + delay_{travel} \tag{3}$$

In Figs. 2 and 3, we show the impact of different approaches on the average message traveling and waiting delay employing 1 to 7 UAVs. It can be seen that different message delivery schemes have not any impact on the average message waiting delay. Message waiting delay decreases by increasing number of UAVs in all approaches. On the other hand, message forwarding impacts mostly on the average message traveling delay. Message traveling delay decreases slightly by increasing number of UAVs in Pure MF. Increasing number of UAVs impacts message traveling delay dramatically if message forwarding between UAVs is enabled like MDF, GF, TAF and TaBAF. TaBAF has the least message traveling delay because a UAV forwards messages to a neighbor if the neighbor can deliver them earlier even if the neighbor will not fly directly to the destination of messages. This strategy decreases the average message traveling delay in the

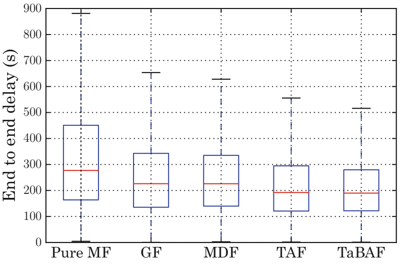


Fig. 1. Average end to end delay of messages.

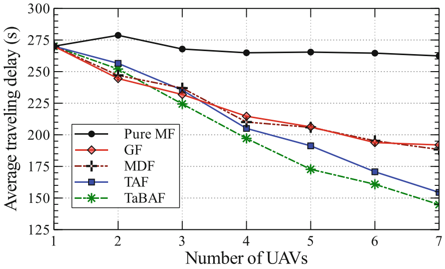


Fig. 2. Average traveling delay of messages in the buffer of a UAV.

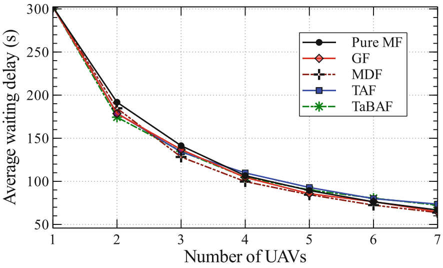


Fig. 3. Average waiting delay of messages in the buffer of a node.

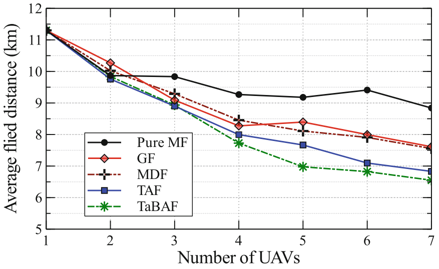


Fig. 4. Average flied distance of a UAV to finish its mission.

network. As mentioned before, TAF takes into account only the next node of a neighbor UAV and does not forward messages if the neighbor UAV will not fly to the destination of messages directly. For this reason, some messages may not be forwarded and have to travel longer in the UAV. MDF and GF show similar average message traveling delay. MDF and GF do not choose the best neighbor UAV which can deliver a message earlier and this causes longer message traveling delays comparing with TAF and TaBAF.

Figure 4 illustrates the average flied distance of each UAV in the network to deliver all generated message (message generation runs for 1000 s). The flied distance of a UAV is the cost that each UAV pays to deliver messages. By applying message forwarding between UAVs in a message ferry network, the cost decreases. However, the cost in TAF and TaBAF is less than GF and MDF due to the efficient message forwarding decisions which avoid message forwarding to a neighbor UAV if a UAV itself can deliver messages earlier. However, the flied distance of a UAV is less in all approaches which apply message forwarding between UAVs than a pure message ferrying.

8 Conclusion

In this paper, we proposed a trajectory and buffer aware message forwarding scheme for multi UAV based messages ferry networks where UAVs cooperate in message delivery by forwarding messages between each other. We introduced an exchange of trajectory and buffer state information between UAVs in form of a signaling when they visit each other on-the-air. Our proposed message forwarding scheme in a UAV exploits signaling information and forwards a message to a neighbor UAV if it can deliver the message earlier. By efficient message forwarding, we decreased end to end message delay in message ferry networks and average flied distance of each UAV which is the cost for a UAV to deliver messages. Moreover, a UAV may replicate messages in nodes or other UAVs to accelerate message delivery. This is our future work.

References

1. Simon, T., Mitschele-Thiel, A.: A self-organized message ferrying algorithm. In: 14th International Workshops on a World of Wireless, Mobile and Multimedia Networks (2013)
2. Harounabadi, M., Rubina, A., Mitschele-Thiel, A.: Cooperative on-the-fly decision making in mobility-controlled multi ferry delay tolerant networks. In: Zhou, Y., Kunz, T. (eds.) Ad Hoc Networks. LNICST, vol. 184, pp. 246–257. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-51204-4_20
3. Harounabadi, M., Mitschele-Thiel, A.: Stigmergic communication for self-organized multi ferry delay tolerant networks. Mobile Netw. Appl. 1–10 (2017)
4. Casas Melo, V., Mitschele-Thiel, A., Harounabadi, M.: On the emergence of virtual roundabouts from distributed force/torque-based UAV collision avoidance scheme. In: International Conference on Control and Automation (ICCA) (2017)
5. Zhao, W., Ammar, M., Zegura, E.: A message ferrying approach for data delivery in sparse mobile ad hoc networks. In: 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (2004)
6. Karp, B., Kung, H.T.: GPSR: greedy perimeter stateless routing for wireless networks. In: 6th International Conference on Mobile computing and Networking (2000)
7. Mansy, A., Ammar, M., Zegura, E.: Deficit round-robin based message ferry routing. In: IEEE Global Telecommunications Conference (2011)
8. Zhang, Z., Fei, Z.: Route design for multiple ferries in delay tolerant networks. In: IEEE Wireless Communications and Networking Conference (2007)
9. He, T., Swami, A., Lee, K.W.: Dispatch-and-search: dynamic multi-ferry control in partitioned mobile networks. In: 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (2011)
10. Asadpour, M., Hummel, K., Giustiniano, D., Draskovic, S.: Route or carry: motion-driven packet forwarding in micro aerial vehicle networks. In: IEEE Transactions on Mobile Computing (2017)
11. Harounabadi, M., Puschmann, A., Artemenko, O., Mitschele-Thiel, A.: TAG: trajectory aware geographical routing in cognitive radio ad hoc networks with UAV nodes. In: Mitton, N., Kantarci, M.E., Gallais, A., Papavassiliou, S. (eds.) ADHOCNETS 2015. LNICST, vol. 155, pp. 111–122. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25067-0_9

Cellular Networks, Sensor Networks

Caching and Computing at the Edge for Mobile Augmented Reality and Virtual Reality (AR/VR) in 5G

Melike Erol-Kantarci(✉) and Sukhmani Sukhmani

School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
{melike.erolkantarci,steer031}@uottawa.ca

Abstract. The enormous increase in powerful mobile devices has created hype for mobile data traffic. The demand for high definition images and good quality video streaming for the mobile users has constantly being escalated over the recent decade. In particular, the newly emerging mobile Augmented Reality and Virtual Reality (AR/VR) applications are anticipated to be among the most demanding applications over wireless networks so far. The architecture of the cellular networks has been centralized over the years, which makes the wireless link capacity, bandwidth and backhaul network difficult to cope with the explosive growth in the mobile user traffic. Along with the rise in overall network traffic, mobile users tend to seek similar types of data at different time instants creating a bottleneck in the backhaul link. To overcome such challenges in a network, emerging techniques of caching the popular content and performing computation at the edge are gaining importance. The emergence of such techniques for near future 5G networks would pose less pressure on the backhaul links as well as the cloud servers, thereby, reducing the end-to-end latency of AR/VR applications. This paper surveys the recent edge computing techniques along with the powerful caching strategies at the edge and provides a roadmap for 5G and beyond wireless networks in the context of emerging applications.

Keywords: In-network caching · Mobile edge computing
Mobile augmented reality and virtual reality · Wireless networks · 5G

1 Introduction

The exponential growth in the number of handheld devices such as mobile phones, and tablets has dramatically increased the mobile data traffic. The demand for rich multimedia applications is also rising enormously. This traffic is presumed to steadily increase over the coming years as well. The major source of traffic in the network is due to the demand of video streaming services. Asynchronous content reuse property is exhibited by these live streaming requests of popular content by mobile users that accounts for most of the data traffic. In addition mobile Augmented Reality and Virtual Reality (AR/VR) are expected to be among the first wave of killer applications in 5G. According to ABI Research, the total AR market is expected to reach $114 billion by 2021 while the total VR market is anticipated to reach $65 billion within the same timeframe [1]. AR/VR applications are highly delay sensitive and their performance can degrade

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 169–177, 2018.
https://doi.org/10.1007/978-3-319-74439-1_15

significantly with non-uniform delay and throughput [19]. The future 5G networks are expected to be more fast, flexible, reliable and resilient with round trip time of requests corresponding to 1 ms taking into account the growing mobile traffic. Successful working of technologies like device-to-device communications, millimeter wave and small cell densification can help to achieve the desired parameters for the 5G networks.

Small cell base stations are to be deployed within micro cells, pico cells and femto cells in order to achieve expected to increase capacity and coverage. The deployment of small cell base stations allows resource reuse to a larger extent. Even though placing the small cell base stations in a cell site of various sizes is the most important enabler for higher data rates, the limitation is the backhaul link capacity that provides connection to the core network. These links are most likely wireless due to rapid deployment, self-configuration and cost efficiency [2]. The backhaul links are limited in capacity due to bandwidth constraints and energy consumption while transmitting the packets over long distances.

In order to deal with such issues and better utilization of the limited available resources, the concept of caching has been exploited. Caching of popular contents at the network edge can significantly improve performance as shown in [3, 4]. Delay is reduced along with overcoming backhaul link congestion [2, 4, 5]. By efficiently exploiting the idea of caching at the edge, backhaul cost can be decreased linearly with the base station cache size according to [6, 7] which is possible by the elimination of redundant traffic as also catered by [8]. This further arises the question of what to cache? Popular YouTube videos specially top the list for caching, as they require high data rates for continuous buffering. Then arises the question of where to cache? As caching can be performed at radio access network (RAN) or the Core Network (CN), one has to decide which place would be more beneficial as well as cost efficient. The existing literature on edge-caching focuses on traditional content access, such as watching YouTube videos over mobile devices. However, in AR/VR the access pattern, content to be cached, and the location of the content could be further optimized based on specific applications. For instance, in a museum visit supported by AR, environment maps and constant content can be cached closer to the user while interactive content only might be shared with servers. This can be further extended to caching on D2D networks where communication pattern is more ad hoc.

In case of upcoming 5G networks the feature of antenna directivity [4] can cause retrieval and connection delays for the users with high mobility. Therefore, performing caching at the edge in a distributed fashion can further trigger improved quality of service. There could arise situations where the cached data is not appropriate to provide for the user requests, then the request can forward to one level up in the hierarchy, that is, to the cloud servers. Taking into account another situation where the user is travelling from one cell site to another and some nomadic user is requesting for a particular data, which the first user possesses, and not the BS of that cell site. Then in this case rather than the base station contacting the other BS of different cell site, the user that already has the requested application can share the data locally. This can happen in the context of device-to-device (D2D) communications, which further facilitates faster communication and poses less pressure on the network resources. This exchange of cached content from device-to-device as in [1, 6] allows directed communications between the users.

This however is not currently implemented in practice, as the users are not willing to share any data over D2D links without receiving any rewards in exchange from the network providers or operators.

On the other hand, cloud computing makes it quite convenient to access shared pool of services and resources that are location independent. The idea of cloud computing in the wireless mobile networks leads to the mobile cloud computing and towards cloud radio access networks. For low-latency applications with high computation load, it is not feasible to transmit large amounts of data over long distances to the cloud servers for computation purposes. To address these issues the concept of edge computing or fog computing has emerged recently [9]. Edge computing urges the deployment of computing resources closer to the end users. The edge device could be any device that resides between the data sources and the cloud based data centers. The computing tasks like processing, storing, caching and load balancing can be performed efficiently in edge computing. The benefits of using edge computing can be summarized by saying that, response time is reduced, enhances performance as computations are performed close to the source, network resources are conserved, reduced latency and minimum bottleneck probability. Cloudlets or edge-clouds are considered to be an important part of the 5G network, in particular to support AR/VR applications [10]. It is worth to note that even though, edge computing is a promising technology for achieving better services and higher data rates, they are not capable of replacing the cloud computing. All the heavy applications and intensive computations would likely to be beyond the scope of edge computing at least over the next several years. For decreasing the pressure on the network resources like bandwidth and backhaul links load offloading, in [11] several techniques are discussed for improving the overall efficiency of the network.

In this paper, we summarize the recently proposed techniques in content caching in and edge computing within the wireless networks. We categorize the edge caching techniques according to where the caching is performed; i.e. at the radio access network, core network and the devices Similarly, for edge computing, we group the techniques based on the placement of computing resources either at the cloud or closer to the users. We discuss the impact of the proposed techniques on delay and throughput. Besides, we survey several techniques that focus on energy-efficiency. In closing, we outline the future perspectives and draw a roadmap for new research directions in particular new requirements of mobile AR/VR applications.

The rest of the paper is organized as follows: Sect. 2 discusses the concepts of caching at the edge. Section 3 focuses on computations being performed at the network edge. Section 4 covers the energy-efficiency aspect of proposed techniques. Finally, Sect. 5 conclusions the paper giving future perspectives.

2 Caching at Edge

Videos of major sports events or viral videos over the social networks are accessed by many users at the same time. Bringing content each and every time from the Internet servers creates a bottleneck in the backhaul network. Considering that these users could also be geographically close, caching at the edge has been proposed in the literature. As the mobility of the user is the main concern for 5G networks, we need appropriate area where caching can be performed. This can be done at the Evolved Packet Core (EPC) or the Radio Access Networks (RAN). We first summarize the techniques that consider caching at EPC and then discuss the studies in RAN.

2.1 Caching Within EPC

EPC includes the serving gateway (S-GW), packet data network gateways (P-GW) and the mobility management entity (MME). The current deployments of cache are done mostly at the P-GW and are known as mobile content delivery networks. In [8], the authors have proposed chunk-level, TPC-level and packet-level caching for EPC. At the chunk-level, the files are initially divided into chunks and then a caching server is used to cache particular chunks. They are then specified and differentiated by hash tags. Further, if the size of the chunk and the hash is the same, the requests related to that chunk are taken care by the same cached chunk. At the TCP-level, TCP flows are managed. Caching intermediates further dividing the file into chunks of fixed or variable length, which helps in scalable cache management. At the packet-level, two middle-boxes at upstream and downstream are used. The role of upstream middle-box is to eliminate redundant bytes whereas the downstream middle-box helps in reconstructing the cached packets. The drawback of this type of caching is that the size of chunks is very small thereby probability of exploding the index sizes in high-speed networks is more.

2.2 Caching at RAN

Caching at RAN is comparatively challenging as tunnels are established between the users and the EPC by the evolved nodeBs (eNB). As the files to be transferred over the connection are first converted into packets and are further encapsulated by GTP tunneling, this make them difficult to perform content aware caching. In order to deal with this situation the concept of byte caching has been implemented. In byte caching, multiple portions of a file are cached at the network layer by searching for the common range of data in the bytes of the packet flows. It does not subdivide the packet flows into fragments, but aim at caching the frequently used bytes in the flows, thereby deleting the redundant ones.

As the caching memory of the base stations or eNBs is limited, complex caching schemes are required to be followed in order to gain more flexibility and cost effective network. There should also be some collaborative schemes installed in the neighboring eNBs for achieving successful RAN caching output. Moving further towards the distributed caching of videos at the base station of RAN, can increase the efficiency of the delivering the content even more. According to [12] caching at RAN with the help of User Preference Profile (UPP) along with the video aware backhaul scheduling, can increase the capacity significantly when compared to the traditional techniques. Considering the scenario in [4] the streaming of videos can be achieved with low handoff delays and less connection latency using caching at the RAN. The streaming of live data or any video also accounts for the base station cache size according to [6]. Their main focus is reducing the transmit power cost, with increased base station cache, which further results in linear decrement of the overall backhaul cost.

2.3 Device-to-Device (D2D) Caching

In [1], the authors consider caching at the small base stations as well as the user terminals, which can carry out their communication using D2D communications. Working of transmission and caching protocols together yield two types of gains that can be evaluated according to [1]. First being the local caching gain, this is achieved when any device withholding pre-cached information about the data locally satisfies the content requested by the user. Second, global caching gain can be achieved. This is a cost effective way of caching for the service providers. The general assumption here is that the user demands are known in advance. When caching is performed at the user terminal, energy consumption over backhaul networks can be also reduced [13, 14]. D2D caching can also borrow some concepts from ad hoc networking which has not been explored so far.

2.4 Transcoding Enabled Caching (TeC)

According to [15], another way of increasing the quality of the video output as requested dynamically by the mobile users is, by using the transcoding enabled caching technique. The combination of transcoding and caching can serve the heterogeneous users in two of the efficient ways: first, by decreasing the user estimated latency; second, by reducing the traffic between the proxy and the main server. This technique works as follows. Two types of transcoders, namely bit-rate reduction and spatial resolution reduction transcoding are introduced. Transcoding unit is placed on the content delivery path such that depending on the connection speed and processing capability of an end user, the content is converted into an appropriate format.

3 Mobile Computing

With the skyrocketing use of smart devices on-the-go, computing on mobile devices, i.e. mobile computing, became an essential part of devices. The limited resources of mobile devices require computational help either from cloud servers or other resources around them, in particular for performing computationally heavy tasks. Mobile Edge Computing (MEC) refers to performing such computations on locations closer to the user than the cloud servers. The very concept of cloud is to provide software that is capable of executing intensive and heavy computations. They are convenient to use and caters the dynamic requests by accessing a shared pool of various configurable devices [16]. Cloud servers can compute extensive applications efficiently, thereby, increasing the battery life of the mobile devices [17]. If the applications that require heavy computation are shared to the smaller version of clouds in the vicinity of the device these are referred to as cloudlets which is a key concept for MEC [18].

3.1 Computing at the Edge

Following the newly emerging AR/VR applications and many other delay-sensitive applications in 5G, it is apparent that providing the required low-latency with cloud computing and the transfer of massive amounts of data to the data centers may not be possible or could be not economical [16]. In order to overcome these drawbacks the concept of edge computing emerges as a promising tool [18]. Edge computing also referred as fog computing, comprises of proxy servers that are located at the network edge. The very idea of edge computing is to achieve low latency and location awareness. It is worth to note that edge computing is not capable of replacing cloud, as heavy applications cannot be realized through them but they can reduce the burden of the cloud servers by locally serving the requests generated by the mobile users. As the cloud and cloudlets are expected work in harmony, their interoperability emerges as a research focus. In the next section, we first focus on the techniques that study this challenge.

3.2 Cloud-Fog Interoperability

In [20], the authors suggest using software defined networking architecture to integrate the infrastructure of cloud and fog. According to the architecture, interoperability can be of two ways:

- Fog-Fog Computing: When the data requested by the mobile user is beyond the scope of a single fog server, it can seek that data from another nearby fog server. This is referred to as fog-fog computing.
- Fog-Cloud Computing: When the service being requested by the user is beyond the scope of the fog server, then the request is forwarded to the cloud server in order to provide the user with the required service.

In addition, caching when combined with MEC can increase the quality of service significantly. In the next section, we summarize the works, that focus on the energy-efficiency improvement when these two approaches are combined.

4 Energy Efficient Caching and Computing

Various techniques of offloading and load balancing are required to satisfy diverse types of requests with balanced energy consumption. For instance, the dynamic offloading framework in [21] helps to balance the load efficiently in a network. Cloud computing differs from traditional models due to the adoption of virtualization. This very feature of cloud leverages the operators to run arbitrary applications from various customers on the virtual machines. The cloud providers reduce the energy consumption on the mobile systems by providing computing cycles to the users aiming to decrease the computation at the mobile user end.

Mobile devices are equipped with multiple network interfaces. The server interface comprises of EDGE, UMTS and GPRS, which are responsible for corresponding with the network operators. The other is the peer interface that includes Bluetooth and IEEE 802.11 that connects to other computing devices [22]. The joint use of these interfaces can lead to energy efficient data applications according to [22]. In addition, as evidenced by [23], most of the studies do not consider energy consumption at the backhaul, however it also contributes a significant amount of energy consumption. Therefore, the potential of MEC to enhance energy-efficiency is supported by their reduced load at the backhaul links.

Moving further towards the caching domain, a generalized notion of saving energy is to simply turn a device down. However, for content-caching with D2D, this inflicts inconsistency in the already cached content. In that case, once caching is performed cache invalidation strategy needs to be implemented to make sure that the data cached in the mobile device is copied to the server. According to [24] cache consistency is difficult to enforce when the mobile devices are powered off.

Procuring the energy efficient mechanisms in the 5G networks, the content placement issue is addressed in [25]. The proposed framework is CAR (cache-at-relay) comprises of three integer linear programming models that aims to reduce device power consumption by uplink power optimization. The impact of caching on the backhaul links is discussed in [26] where several techniques are shown to relax the load on the backhaul. In mobile AR/VR applications battery is one of the major bottlenecks. Therefore energy-efficient techniques are expected to play a key role when it comes to adoption of edge-caching or edge-computing techniques.

5 Conclusions and Future Directions

5G and beyond networks are aiming to serve many applications that desire ultra-low latency. For instance, mobile AR/VR applications, tactile internet and autonomous driving require latency values close to 1 ms. They also require frequent video or map access and streaming content which calls for placing caching and computing resources closer to mobile users. This paper surveys the recent studies that focus on caching and computing at the edge. Various types of caching techniques have been discussed at the network core as well as the network edge. The combination of various kinds of caching techniques that reduce the duplicity and eliminate redundant traffic in the network have been summarized. Further, mobile edge computing techniques have been discussed. Computations can be done either at the cloud server or locally at the edge server. Though the cloud-based servers are designed to support heavy and complex computations, the fog servers are capable of catering less complex applications. They can be implemented at the edge, which would further take into account the often-requested content and their output.

The major focus of research in the literature has been enhancing the quality of service and energy-efficiency. Although these are relevant to AR/VR and other ultra-low latency applications, next-generation wireless networks will need more flexibility and configurability capabilities. With this regard, Software Defined Networking (SDN) is expected to play a critical role in increasing the flexibility of caching and computing in mobile networks. SDN allows heterogeneous devices to be programmed by the controller dynamically. The proper coordination of the controller management along with the device executions, caching and computations can enhance AR/VR experience.

Furthermore, it is apparent that 5G and beyond wireless networks will have significant differences than today's LTE networks. The high amount of investments suggests that the architecture needs to be flexible enough to support various applications. In other words, functionality to support AR/VR applications should, for example, support autonomous car applications. As a result, application-specific platform services need to be dynamically adapted. Caching and computing at the edge is a powerful tool to support dynamicity however scalable control of large number of distributed systems becomes the challenging part. In addition, cross-application performance uniformity will emerge as a significant challenge. As a future direction, advanced algorithms implemented over SDN are expected to play an important role in reconfiguring 5G networks to satisfy the demands of newly emerging applications.

References

1. ABI Research and Qualcomm: Augmented and Virtual Reality: the First Wave of 5G Killer Apps. White paper (2017). https://www.qualcomm.com/news/onq/2017/02/01/vr-and-ar-are-pushing-limits-connectivity-5g-our-rescue
2. Gregori, M., Gómez-Vilardebó, J., Matamoros, J., Gündüz, D.: Wireless content caching for small cell and D2D networks. IEEE J. Sel. Areas Commun. **34**(5), 1222–1234 (2016)
3. Goebbels, S., Jennen, R.: Enhancements in wireless broadband networks using smart caching an analytical evaluation. In: Proceedings of International Symposium Personal, Indoor and Mobile Radio Communications, pp. 1–5, September 2008
4. Qiao, J., He, Y., Shen, X.S.: Proactive caching for mobile video streaming in millimeter wave 5G networks. IEEE Trans. Wireless Commun. **15**(10), 7187–7198 (2016)
5. Dehghan, M. et al.: On the complexity of optimal routing and content caching in heterogeneous networks. In: Proceedings of IEEE Conference on Computer Communications, pp. 936–944, April 2015
6. Liu, A., Lau, V.K.N.: Exploiting base station caching in MIMO cellular networks: opportunistic cooperation for video streaming. IEEE Trans. Signal Process. **63**(1), 57–69 (2015)
7. Golrezaei, N., Molisch, A.F., Dimakis, A.G., Caire, G.: Femtocaching and device-to-device collaboration: a new architecture for wireless video distribution. IEEE Commun. Mag. **51**(4), 142–149 (2013)
8. Wang, X., Chen, M., Taleb, T., Ksentini, A., Leung, V.C.M.: Cache in the air: exploiting content caching and delivery techniques for 5G systems. IEEE Commun. Mag. **52**(2), 131–139 (2014)
9. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017)
10. Seam, A., Poll, A., Wright, R., Mueller, J., Hoodbhoy, F.: Enabling mobile augmented and virtual reality with 5G networks. ATT White paper (2017). http://about.att.com/content/dam/innovationblogdocs/Enabling%20Mobile%20Augmented%20and%20Virtual%20Reality%20with%205G%20Networks.pdf
11. Kumar, K., Lu, Y.H.: Cloud computing for mobile users: can offloading computation save energy? Computer **43**(4), 51–56 (2010)
12. Ahlehagh, H., Dey, S.: Video-aware scheduling and caching in the radio access network. IEEE/ACM Trans. Networking **22**(5), 1444–1462 (2014)
13. Ji, M., Caire, G., Molisch, A.F.: Wireless device-to-device caching networks: basic principles and system performance. IEEE J. Sel. Areas Commun. **34**(1), 176–189 (2016)

14. Ji, M., Caire, G., Molisch, A.F.: Fundamental limits of caching in wireless D2D networks. IEEE Trans. Inf. Theory **62**(2), 849–869 (2016)
15. Shen, B., Lee, S.-J., Basu, S.: Caching strategies in transcoding-enabled proxy systems for streaming media distribution networks. IEEE Trans. Multimedia **6**(2), 375–386 (2004)
16. Huo, R., Yu, R., Huang, T., Xie, R., Liu, J., Leung, V.C.M., Liu, Y.: Software defined networking, caching, and computing for green wireless networks. IEEE Commun. Mag. **54**(11), 185–193 (2016)
17. Goudarzi, M., Movahedi, Z., Nazari, M.: Mobile cloud computing: a multisite computation offloading. In: 8th International Symposium on Telecommunications (IST), Tehran, pp. 660–665 (2016)
18. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The case for VM-based cloudlets in mobile computing. IEEE Pervasive Comput. **8**(4), 14–23 (2009)
19. Westpal, C.: Challenges in networking to support augmented reality and virtual reality. In: ICNC 2017 (2017)
20. Yang, P., Zhang, N., Bi, Y., Yu, L., Shen, X.: Catalyzing cloud-fog interoperation in 5G wireless networks: an SDN approach. Technical report, April 2017. https://arxiv.org/pdf/1612.05291.pdf
21. Kosta, S., Aucinas, A., Hui, P., Mortier, R., Zhang, X.: ThinkAir: dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In: Proceedings IEEE INFOCOM, Orlando, FL, pp. 945–953 (2012)
22. Yeung, M.K.H., Kwok, Y.-K.: A game theoretic approach to energy efficient cooperative cache maintenance in MANETs. In: IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, Berlin, vol. 3, pp. 1500–1504 (2005)
23. Huq, K.M.S., Mumtaz, S., Bachmatiuk, J., Rodriguez, J., Wang, X., Aguiar, R.L.: Green HetNet CoMP: energy efficiency analysis and optimization. IEEE Trans. Veh. Technol. **64**(10), 4670–4683 (2015)
24. Wu, K.-L., Yu, P.S., Chen, M.-S.: Energy-efficient caching for wireless mobile computing. In: Proceedings of the Twelfth International Conference on Data Engineering, New Orleans, LA, pp. 336–343 (1996)
25. Erol-Kantarci, M.: Cache-at-relay: energy-efficient content placement for next-generation wireless relays. Int. J. Netw. Manage. **25**, 454–470 (2015)
26. Bahmani, K., Argyriou, A., Erol-Kantarci, M.: Backhaul relaxation through caching. In: Imran, M., Raza, S.A., Shakir, M.Z. (eds.) Access, Fronthaul and Backhaul for 5G Wireless Networks. IET (2017)

Evaluation of a Location Reporting System for mmWave Communication

Yudong Fang[1(✉)], Wilson Tsang[2], Bernard Doray[1], and Yonghong Huang[1]

[1] Communications Research Centre Canada, Ottawa, Canada
{yudong.fang,bernard.doray,yonghong.huang}@canada.ca
[2] Waterloo University, Waterloo, Canada
w.tsang96@gmail.com

Abstract. This paper presents a sampling alignment data processing (SADP) methodology for un-synchronized data to evaluate the precision and accuracy of a positioning system (the Google Tango system); this location system is used to control beams of directional antennas for a 5th generation communication system using millimeter Wave (mmWave) links. The test mathematical model is described in details to derive the sampling alignment data processing. The SADP is used to evaluate the indoor and outdoor scenarios for the Google Tango position system, and we conclude that the Tango system precision is impacted by the tester's behaviors, environment characteristics, and weather conditions. The evaluation results show the suitability of the Tango system as a location reporting system for our mmWave communication system.

Keywords: Position system · mmWave communication · Google Tango
Data fusion · Sampling alignment data processing · Error analysis

1 Introduction

Millimeter Wave (mmWave) communication has become a hot topic in recent years, especially for 5th Generation and Ad-hoc communication systems. But with mmWave, the signal attenuation is very high at the high frequencies used for communication. In order to compensate for this attenuation, directional antennas must be used to improve the range of the communication link. Directional antennas provide the additional benefits of reducing the interference to other nodes and improving the security of the communication link (by reducing the risk of jamming and eavesdropping [1, 2] by other parties). In order to preserve a wide coverage for the communication system, multiple directional antennas must be used and a system must be provided with the capability to select the best antenna to communicate with a given mobile station based on its position.

In the communication scenarios considered in this paper, the stationary basestation has several directional antennas to be able to concentrate the radio signal in different directions, but only one directional antenna can transmit at any given time. In order to select the best antenna to communicate with a mobile user station, accurate and precise position information about the moving station is required. This location information is used by the algorithm running on the basestation, to select the best antenna.

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 178–188, 2018.
https://doi.org/10.1007/978-3-319-74439-1_16

In this project, we use the Google Tango smartphone/tablet to provide the location information. Since the data captured is not aligned in time and location, we propose a sampling alignment data processing method to evaluate the performance.

In Sect. 2 we review indoor and outdoor position techniques and systems. Section 3 provides an overview of the architecture of the system presented in this paper. Section 4 provides an evaluation of the performance of the Google Tango systems for several test cases. It also describes the method designed to process the data so that we can derive a quantitative analysis of the results for different scenarios considered here.

2 Literature Review

Various systems are available to provide location estimation; these systems are based on satellites systems, RF detection, inertial sensors [3], image recognition, or a combination of several of these techniques.

Global Navigation Satellite System (GNSS) is the generic term for navigation systems using satellites to provide autonomous geo-spatial positioning with global coverage [4]. It is often referred to as: GPS, GLONASS, Galileo, or Beidou based on the regional satellite constellation used for positioning. By accessing the signal information from multiple satellites, the location system can provide accurate and precise location information.

GPS is a mature technology. GPS-enabled smartphones are typically accurate to be within a 4.9 m [5] if a clear view of the sky is available to the smartphone. A GPS system will not be able to work indoor since the device does not have a clear view of the sky. This can also happen in tunnels and in “urban canyons” where the signals from the satellites are blocked by high rise building.

The GNSS accuracy has significantly improved with the use of more satellite constellations and the development of wide-area augmentation systems (WAAS) [6]. To improve the accuracy of GPS systems, differential GPS systems have been developed [7]. In a differential system, a network of stationary ground-based reference stations broadcast the difference between their position indicated by their GPS and their known fixed location to provide a correction to nearby mobile GPS devices. Differential GPS can achieve accuracy up to cm level; they are used in certain industrial applications and in agricultural applications but they are very expensive.

For positioning systems based on Radio Frequency (RF) signals, we focus on indoor system since many of these systems are focused on indoor applications. These systems can be classified into several types depending on various aspects such as: where the RF positioning signal originates, the frequency of this signal, and whether the system is self-contained or it requires several units to be deployed [3]. The RF signals can be from wireless local area networks (WLAN) systems, from cellular communication towers, or generated by a moving equipment itself or by a dedicated infrastructure deployed specifically for localization (such as Near Field Communication, and Radio Frequency Identification systems). These systems use received signal strength, time of arrival or time difference of arrival (from different sources) to estimate the location of a device. During the estimation process, the system collects the fingerprint of the signal and it then uses a triangulation algorithm to calculate the position information. Since there is no good model for indoor radio multipath, some new methods are proposed, such as scene analysis, and radio mapping.

Simultaneous Localization and Mapping (SLAM) is a popular research topic, which the system constructs and/or updates a map of an unknown environment while simultaneously it keeps track of its location within it [8, 9] by using variety sensors. The sensors can be categorized into laser-based, sonar-based, and vision-based systems. The mathematical process methods include Kalman filters and Bayes' rules. Some open SLAM methods have been published in [10].

3 System Architecture

3.1 Test System

An application running on Tango device, captures the position updates from the device's operating system and hardware in the form of an (x, y, z) tuple for the coordinates and 3 angles (yaw, pitch, and roll) indicating the orientation of the device at that location. This 6-tuple is referred as Pose [11] in the Tango terminology. In our application, we use this information to infer the position of the mobile user device with respect to the basestation, and push the (x, y, z) coordinates updates to the basestation using a protocol called MQTT (Message Queue Telemetry Transport) [12].

The mobile user station provides its location to the fixed basestation over a separate control channel. The MQTT server on the basestation would receive these locations and make them available to all MQTT clients that have registered interest, in our case this being the location data analysis software. The antenna selection algorithm can then use this location information to select the best antenna to communicate with a given mobile station as shown in Fig. 1.

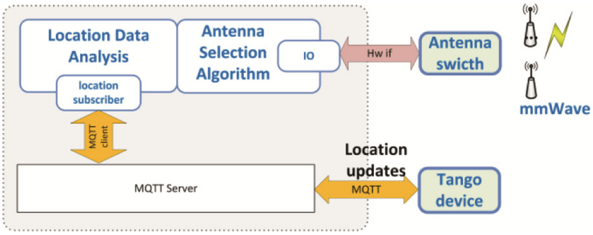


Fig. 1. System architecture

3.2 Google Tango [13]

Google has developed an Android device (tablet or smartphone) for accurate indoor positioning. The system has a wide-angle camera, a depth sensing camera, and accurate inertial sensor timestamping, as well as software application programming interface to access these capabilities. This system fuses the information from these sensors to provide location information; if area learning has previously been done in the location of interest, it will also report location with respect to the point of reference for this learned area. The Tango Area Learning gives the device the ability to see and remember the key visual features of a physical space, such as the edges, corners, other unique features and it uses this information to locate the Tango device within this frame of reference; the Tango can also recognize the area and improve the accuracy of its position reporting. The Tango system also uses the SLAM to track and save the data.

3.3 Test Scenarios

Our mmWave system works indoor and for certain cases outdoor. We tested the Tango positioning system for the following 4 scenarios to check its performance.

Indoor Office Layout: This test is a standard indoor open office layout. The layout includes cubicles with 5-feet tall partitions and common sitting area for meetings. The goal for using this layout is to check behavior of the positioning system for typical office layouts.

Indoor Multilevel Layout: This test layout includes two tunnels (connecting buildings), stairs, and a cafeteria (a large common area). The goal of this layout is to check the 3-D accuracy of the positioning system.

Outdoor, Snow Covered Parking Lot: This outdoor test layout includes a snow covered parking lot. The intention is to check whether Tango can work in this kind of outdoor set-up (where the snow cover might reduce the number of distinguishing features captured by the Tango system).

Outdoor, Sunny Courtyard: This outdoor test layout is at a city hall courtyard (City of Ottawa), and there are trees, light poles, and status as shown in Fig. 4. The intention is to check whether the Tango can work well in sunny situations where the strong sunlight might affect the Tango camera system.

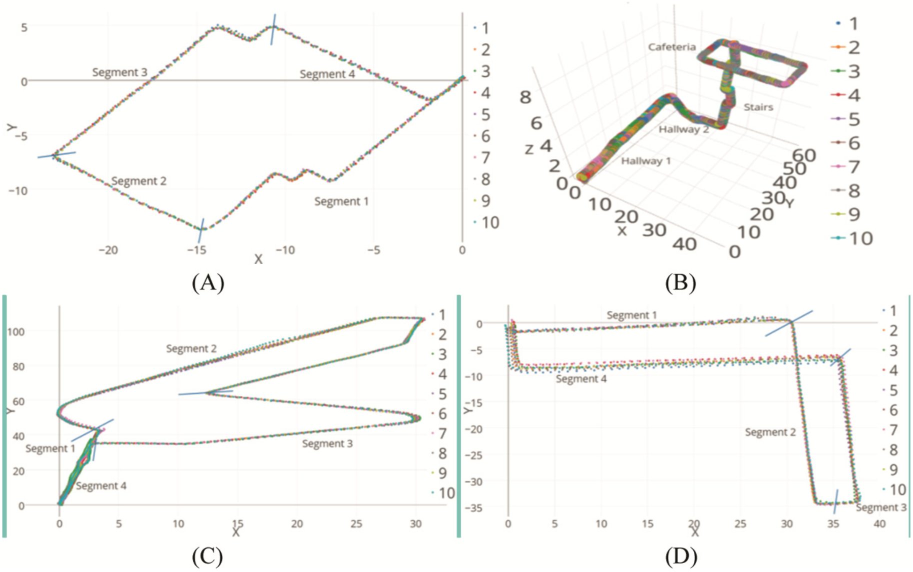
4 Location Data Analysis

4.1 Method to Capture the Data

The location data is captured by a person walking the same route at the same speed multiple times. The position data is recorded every 0.5 s. Each test scenario is captured 10 times. Since each test run is carried by a same person, this introduces variations in the parameters for each test run, which include:

- Not exactly the exact same routes for each test run
- Not exactly the same speed between different test runs
- Not exactly the same speed within one test run.

The raw data captured for the four scenarios is shown in Fig. 2.



(A) (B)

(C) (D)

Fig. 2. Test result for the four scenarios, (A) Indoor office, (B) Indoor multilevel layout, (C) Snow covered parking lot, and (D) Sunny courtyard. For each scenario, we show the segments.

4.2 Mathematical Model

Let $s(t)$ be the location at time t and let $v(t)$ be the speed at the time t. The location of the time t will be:

$$s(t) = \int_{0}^{t} v(\tau)d\tau + \partial_t \tag{1}$$

Here, $s(t)$is the 3-D vector (x, y, z) as Tango can report 3-D data relative to the starting point of the learned area. And, ∂_t is the error offset of the location with the planned true route at the time t. The offset can be due to variations in the person's standing postures or due to other factors, like avoiding other people during a test run. This offset is a random parameter. In this project, we assume that ∂_t is a variable distributed according to the Gaussian distribution with the mean value equal to zero.

$$\partial(t) = \frac{1}{\propto \sqrt{2\pi}} e^{-t^2/2a^2} \tag{2}$$

For the Gaussian distribution, we can use average value to approach the "true value". So we need to do multiple test runs to obtain the mean. In this project, we ran the tests 10 times for each test scenario.

Since our data sample interval is very short (0.5 s in our test), we can assume that the speed is uniform during this time period; calling it v_t for each test run, we can then change Eq. (1) to:

$$s^{i}_{t_j} = \sum\nolimits_{j=0}^{t_j} v^{i}_{j}\Delta + \partial^{i}_{t_j} \tag{3}$$

Here, the superscript i means the i-th test run, j means the j-th time stamp and Δ is the sample interval. For all test runs, we can calculate the average $s^{i}_{t_j}$ as follows:

$$\overline{s_{t_j}} = \frac{1}{M}\sum\nolimits_{i=1}^{M}\left(\sum\nolimits_{j=0}^{t_j} v^{i}_{j}\Delta + \partial^{i}_{t_j}\right) = \frac{1}{M}\sum\nolimits_{i=1}^{M}\sum\nolimits_{j=0}^{t_j} v^{i}_{j}\Delta + \frac{1}{M}\sum\nolimits_{i=1}^{M}\partial^{i}_{t_j} \tag{4}$$

Where M is the number of test runs. The second part is zero due to the Gaussian distribution. Also, we can change the summation order as follows:

$$\overline{s_{t_j}} = \frac{1}{M}\sum\nolimits_{j=0}^{t_j}\sum\nolimits_{i=1}^{M} v^{i}_{j}\Delta = \sum\nolimits_{j=0}^{t_j}\frac{1}{M}\sum\nolimits_{i=1}^{M} v^{i}_{j}\Delta \tag{5}$$

Now, we can use each interval average of $\overline{s_{t_j}}$ as the "**true location**" of the route.

However, we cannot assume that different test runs have the same speed. Therefore, we cannot calculate the average directly as we can see from Fig. 3. The speeds are different for the different test runs, the locations at time t_j are also different.

To adjust the data to the same relative location, we use a resampling [14] method to align all the test runs sample data in a given location segment. The sampling alignment lines are shown in Fig. 3. After alignment, the data samples can be combined to calculate the mean and standard deviation of locations.

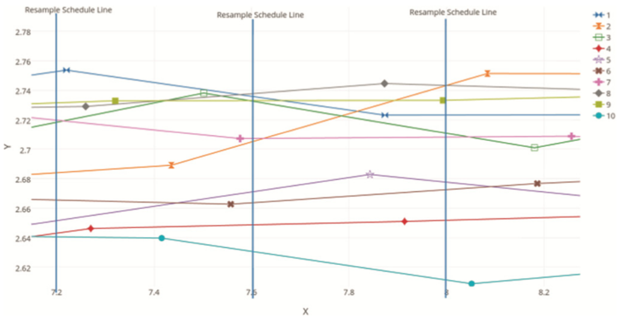


Fig. 3. Resampling data to align them together (x and y in meters)

4.3 Part1: Comparison on Different Test Scenarios Procedures

Initial Process

Since the tester may be stationary for a short moment at the beginning and the end of a test run, this stationary data should be removed. Stationary data means distance between the $s^{i}_{t_j}$ and $s^{i}_{t_{j+1}}$ are close to zero (no movement).

Segmentation

We partition the test runs in different segments to show that the location precision is affected by different factors of the environments. Therefore we need to define some segmentation points to segment the entire route. For each test scenario,

- We manually choose a suitable segmentation points in the route.
- For each walking test run, we use the Euclidean distance to find the closest to the segmentation points.
- We calculate the number of samples in each segment.
- We find the median number of the whole test runs M for the same segment.

Resampling

For each segment, we use a resampling function to ensure that a given segment has the same number of samples (the median number obtained from each segment) across the different test runs. We resample by interpolating the points using a cubic spline.

Calculating Metrics

We calculate the mean for the location accuracy and standard deviation (SD) for the location precision. Then, we calculate the Cumulative Distribution Function (CDF) for each test scenario to evaluate the precision for each segment in different test scenarios. The CDF provides a better visual representation of the results.

4.4 Part 2: System Error Estimation

The goal of this test is to estimate the SD of ∂_{t} in Eq. (1), which is the system error due to the person's position and the Tango accumulated error. To do this test, we run n times test runs. For each test, the person starts and ends a route at the exact same location. At the time t_j, we calculate the SD of the location s_{t_j}.

We know the SD equation is:

$$\sigma = \sqrt{\frac{1}{M}\sum\nolimits_{i=1}^{M}\left(s^{i}_{t_j} - \overline{s_{t_j}}\right)^2} \tag{6}$$

Considering Eqs. (1) and (2), we have

$$\sigma = \sqrt{\frac{1}{M}\sum\nolimits_{i=1}^{M}\left(s^{i}_{t_j} + \partial_{t_j} - \overline{s_{t_j}}\right)^2} \tag{7}$$

Since $\overline{s_{t_j}}$ is approximate to S_{t_j}

$$\sigma\sqrt{\frac{1}{M}\sum\nolimits_{i=1}^{M}\left(S_{t_j} + \partial_{t_j} - S_{t_j}\right)^2} = \sqrt{\frac{1}{M}\sum\nolimits_{i=1}^{M}\left(\partial_{t_j}\right)^2} \tag{8}$$

4.5 Test Results

Segments in an indoor office environment

As shown in Table 1, we can see that in an indoor setting, all the segments yield good precision reporting. In the indoor environments there are many objects in the field of view of the Tango system, and the system uses this information to get better results. In segment 4 we see that the SD is double compared to the other segments because there is a long hallway with few distinguishing features in segment 4. With fewer distinguishing features the accuracy of Tango drops.

Table 1. SD of the all four scenarios

Scenarios	Seg. 1	Seg. 2	Seg. 3	Seg. 4
Indoor office	0.162474	0.151679	0.148068	0.287468
Indoor multilevel	0.48877	0.44967	0.18591	0.42809
Sunny outdoor	0.547403	0.271107	0.386391	0.837084
Snow covered	0.545117	0.941061	0.405412	0.567393

Segments in a multilevel environment

As shown in Table 1, we can see that the Tango can work well in the indoor multilevel environment. The hallway segments (Seg. 1 and Seg. 2) have a similar accuracy. The stairs (Seg. 3) have the best accuracy because there are many features for the Tango to recognize such as steps and hand rails. Each step on the stairs has a height of 0.18 m which is similar to the SD of the test. The cafeteria (Seg. 4) has poor accuracy in this scenario. This is due to the cafeteria being very open. The accuracy of the hallways is poor because the hallways have few distinguishing features for the device to recognize. We notice that the change in elevation does not affect the accuracy of the Tango.

Segments in a sunny outdoor courtyard environment

As shown in Table 1, there are four segments in the outdoor courtyard scenario. In Fig. 4, segment 1 is in red, segment 2 is in green, segment 3 is in blue, and segment 4 is in yellow. Segment 1 and 4 are wide areas. The camera in segment 1 is in the shades and the camera in segment 4 is pointing toward the sunlight. Segment 2 and 3 have many trees which help the Tango system recognize the environment, segment 2 is closer to a building and segment 3 is in the middle of the area. We can see that results for segment 2 and 3 are better than that for segment 1 and 4; segment 2 obtains the best results and segment 4 has worse results than segment 1 because of the sunlight affects the camera.

Segments in a snow covered parking lot

As shown in Table 1, there are four segments in the snow covered parking lot as shown in Fig. 2C. For this scenario, segment 1 and 4 are very similar since they correspond to the same path in opposite directions. In segment 2 we see a drop in accuracy because in this segment the Tango system is in an open area parking lot where there are few objects to help the Tango recognize the environment. In segment 3 we see that the accuracy is better since the Tango system is on a sidewalk with many objects for the Tango to use as reference but the accuracy on the sidewalks is reduced because of the sunlight reflection on the snow.

CDF

In the CDF shown in Fig. 6, the y-axis is the cumulative probability within the distance error specified by x-axis. So, the curve with higher position has better performance. From the CDF figure, we can see that

- Tango works best for indoor scenarios.
- Tango, outdoor, works better in constrained areas with many distinctive features.

System Error Estimation

In this test, we estimate the system error offset listed in Eq. (1). The tester walks to a specific point and stay there for several seconds. The resulting position data is plotted in Fig. 5. The different colors show the final locations for different test runs. We can see when the tester tries to reach the same final locations; he ends up standing at slightly different places. The points of the same color represent the captured data while the tester is standing still at this location for several seconds. The position variations in the x and y directions are shown in centimeters. Those errors could be introduced by the tester's posture change while holding the Tango device. As we discussed before, we assume these errors follow a normal distribution. We calculate the SD of all the data and the value is less than 0.08 m, which includes the position error, the final location decision error, and also the Tango device error for this area leaning.

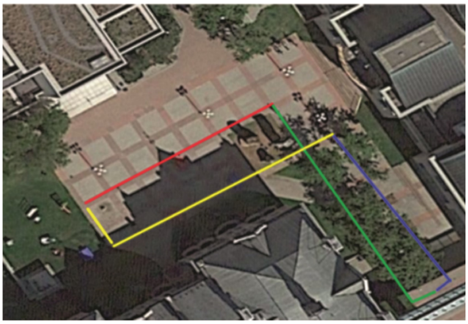


Fig. 4. Segments for the outdoor courtyard (Color figure online)

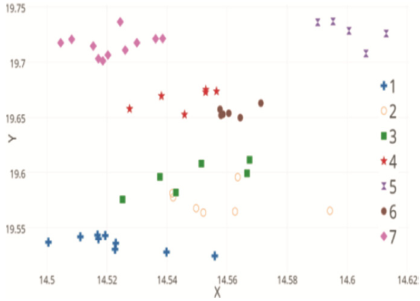
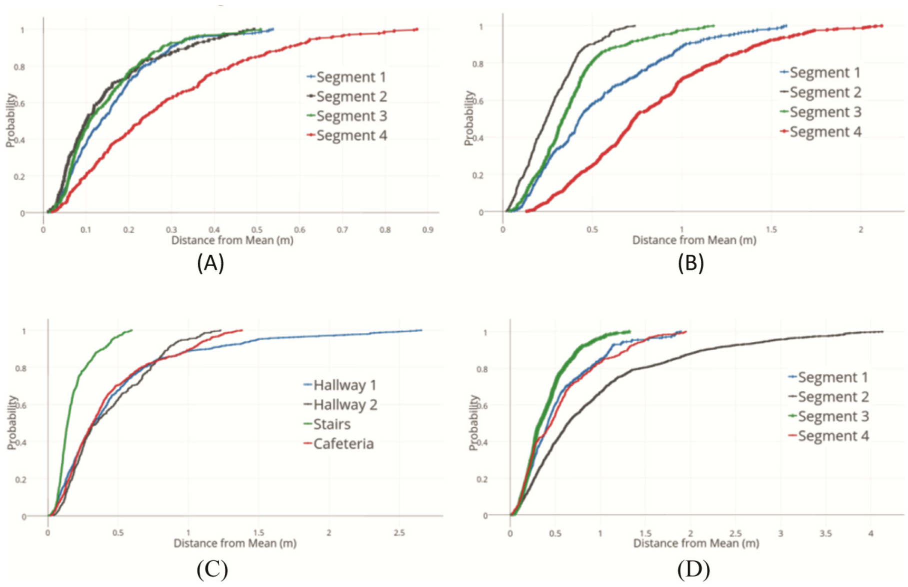


Fig. 5. Cluster of destination points (Color figure online)



(A) (B) (C) (D)

Fig. 6. CDFs for: (A) Indoor office, (B) Indoor multilevel layout, (C) Snow covered parking lot, and (D) Sunny courtyard.

5 Conclusions

In this paper, we have developed a sampling alignment data processing method to evaluate the Tango location system used to control the directional antenna beams in an mmWave communication system. It has been found that the Tango system gives good location accuracy for both indoor and outdoor scenarios (if enough distinctive features are available in the scenery). This system provides better accuracy and precision for the scenarios with more distinctive features, such as an indoor multilevel scenario with staircases and handrails. The location precision is impacted by the tester's behavior, environment characteristics, and weather conditions. The calibration procedure proposed in the paper can be used to further minimize the location error and improve the final accuracy. The evaluation results show the suitability of the Tango system as a location reporting system for mmWave communication system.

References

1. A survey on MAC protocols for wireless adhoc networks with beamforming antennas. IEEE Xplore Document. http://ieeexplore.ieee.org/abstract/document/5755215/. Accessed 18 Apr 2017
2. Zhou, Y., Fang, Y., Tanguay, M.: An optimized link scheduling technique for mobile ad hoc networks using directional antennas. In: 2015 IEEE Military Communications Conference, MILCOM 2015, pp. 714–719 (2015)

3. Harle, R., Farrell, J.A.: A survey of indoor inertial positioning systems for pedestrians. IEEE Commun. Surv. Tutor. **15**(3), 1281–1293 (2013)
4. Groves, P.D.: Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, 2nd edn. Artech House, Norwood (2013)
5. GPS.gov: GPS Accuracy. http://www.gps.gov/systems/gps/performance/accuracy/. Accessed 12 Apr 2017
6. Loh, R., Wullschleger, V., Elrod, B., Lage, M., Haas, F.: The U.S. wide-area augmentation system (WAAS). Navigation **42**(3), 435–465 (1995)
7. Zhao, S., Chen, Y., Zhang, H., Farrell, J.A.: Differential GPS aided inertial navigation: a contemplative realtime approach. IFAC Proc. Vol. **47**(3), 8959–8964 (2014)
8. Dissanayake, M.W.M.G., Newman, P., Clark, S., Durrant-Whyte, H.F., Csorba, M.: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Trans. Robot. Autom. **17**(3), 229–241 (2001)
9. Younes, G., Asmar, D., Shammas, E.: A survey on non-filter-based monocular visual SLAM systems. arXiv:160700470 [Cs], July 2016
10. OpenSLAM.org. https://www.openslam.org/. Accessed 13 Apr 2017
11. What Are Tango Poses?—Tango. Google Developers. https://developers.google.com/tango/overview/poses. Accessed 08 May 2017
12. MQTT. http://mqtt.org/. Accessed 04 May 2017
13. Tango Concepts—Tango. Google Developers. https://developers.google.com/tango/overview/concepts. Accessed 13 Mar 2017
14. Oppenheim, A.V.: Discrete-Time Signal Processing. Pearson Education, London (1999)

Fair Scheduling of Two-Hop Transmission with Energy Harvesting

Andrey Garnaev[1,2(✉)] and Wade Trappe[2]

[1] Saint Petersburg State University, St. Petersburg, Russia
garnaev@yahoo.com
[2] WINLAB, Rutgers University, North Brunswick, USA
trappe@winlab.rutgers.edu

Abstract. In this paper, we consider a two-hop network with a source node (SN) and a relay node (RN) who want to communicate data to a destination node (DN). The SN cannot be directly connected to the DN, but rather is connected only via the RN. The RN does not have an external source of energy, and thus needs to harvest energy from the SN to communicate, while the SN has an external source of energy and can harvest energy straight from it. Thus, a dilemma for the SN arises: how much to share harvested energy with the RN to make it relay the SN's data to the DN. Fair performing of their communication tasks is considered as an incentive for the SN and the RN to cooperate. The optimal α fair schedule is found for each α. It is shown that an altruistic strategy for one of the nodes comes in as a part of the cooperative solution (corresponding $\alpha = 0$), while the maxmin strategy (corresponding α tending to infinity) is proved to be egalitarian. Using Nash bargaining over the obtained continuum of fair solutions, we design a trade-off strategy.

Keywords: Adhocnets · Energy harvesting · Fairness · Bargaining

1 Introduction

Using nodes powered by green energy in wireless networks (e.g., sensor networks) allows one to prolong their lifetime and increase their sustainability. Besides of harvesting energy from such sources of green energy as solar radiation or piezoelectric devices, nodes can also harvest energy using radio frequency (RF) transmissions from other wireless nodes [13]. Such energy transfer technologies can serve as a basis for nodes cooperating, thereby leading to improved overall network performance [18]. In [11], a problem of two-hop relaying with energy harvesting nodes was considered and the optimal transmission scheme with the source having a single energy packet was found for a half-duplex relay. In [16], a directional waterfilling algorithm was derived for a Gaussian fading channel with an energy harvesting transmitter. In [17], a game-theoretical approach was used to minimize the non-renewable energy consumption in a multi-tier cellular network. In [19], a stochastic energy trading game was developed with two types of energy-harvesting devices: sellers harvesting more energy than they can

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 189–198, 2018.
https://doi.org/10.1007/978-3-319-74439-1_17

use, and buyers that have to buy energy to support their required communication services. In [20], an optimal packet scheduling problem for a single-user transmission with discrete energy harvesting was considered.

In [18], a question was put forward: *how can one make the nodes cooperate?* To find an answer, in [18], a game-theoretical model for the relay node, powered solely by wireless energy transfer from the source node, was suggested. A pricing scheme was considered as an incentive for cooperation. It was shown that altruistic operation of the nodes can be facilitated by the proposed pricing. In this paper, we consider fairness (namely, α-fairness) in performing the communication tasks by the nodes as an incentive to cooperate. The optimal α-fair schedule is found for each α. We show that an altruistic strategy for one of the nodes comes in as a part of the cooperative solution corresponding to a boundary value of α equals to zero. The benefits of a cooperative strategy is that it maximizes total network performance, while the drawback is that it is not energy safe. On the other hand, the benefits of a maxmin strategy (corresponding to the other boundary case of α tending to infinity) is that it is egalitarian and energy safe compared to cooperative, but the drawbacks of the maxmin strategy is that it supports lower total network performance. Hence there is a fundamental problem associated with selecting a fairness coefficient, which arises from the trade-off between altruistic and egalitarian strategies. A core contribution of this paper is that, by applying Nash bargaining approach, we design such a trade-off strategy.

The organization of this paper is as follows: in Sect. 2, a model where the source node sends data directly to the destination node is studied. In Sect. 3, the model is generalized for the scenario where transmission is performed via a relay node. The α-fair schedule is found for each α. In Sect. 4, trade-off value for fairness coefficient using Nash barging approach is obtained. Finally, in Sect. 5, conclusions are offered, and, in Appendix A, the proof of the obtain results are given.

2 The Source Node Sends Data Directly to the Destination

Let a source node (SN) send data to a destination node (DN), but have to harvest energy from an external source. During energy harvesting the SN cannot send data. The rate of energy harvesting p_h, reflects the energy harvested per time unit, is fixed. The goal of the SN is to harvest energy to maintain sending data to the DN within a time slot of duration T. The SN, to send data, applies a fixed power p_s per unit of time, called power (transmission) rate. Thus, the time slot is split into two phases: (a) *energy harvesting* (duration T_h) and (b) *communication* (duration T_s), where

$$T_h + T_s = T. \tag{1}$$

The total energy accumulated by the SN within energy harvesting phase is given as follows:

$$E = p_h T_h. \tag{2}$$

The total energy $p_s T_s$ used by the SN in communication phase cannot be larger than the accumulated energy E, i.e., by (2),

$$p_s T_s \leq p_h T_h. \tag{3}$$

Due to the SN, to send data, applies a fixed power p_s per unit of time within the communication phase having duration T_s, the total throughput is given as follows:

$$v(\boldsymbol{T}) = T_s \ln(1 + h_s p_s) \text{ with } \boldsymbol{T} = (T_h, T_s). \tag{4}$$

Then, in the framework of this model, two sequential optimization problems arise: (a) to optimize the phase schedule to maximize throughput for a fixed power rate, and (b) to optimize the power transmission rate to send data.

(a) To optimize the phase schedule to maximize throughput for a fixed power rate: Here, the goal of the SN is to find the phase schedule $\boldsymbol{T} = (T_h, T_s)$ to maximize throughput (4) for a fixed p_s.

Theorem 1. *For a fixed power rate p_s, the optimal phase schedule $\boldsymbol{T}$ is given as follows:*

$$T_s = p_h T/(p_h + p_s) \, and T_h = p_s T/(p_h + p_s). \tag{5}$$

This schedule yields the total SN throughput as a function of p_s is equal to:

$$\overline{v}(p_s) = p_h T \ln(1 + h_s p_s)/(p_h + p_s). \tag{6}$$

(b) To optimize the power transmission rate to send data. Here, the power rate p_b is considered as a variable controlled by the SN. Thus, the phase schedule (5) and the total SN's throughput (6) are functions of p_s. The goal of the SN is to find p_s to maximize this total SN throughput.

Theorem 2. *The optimal power rate to send data is equal to:*

$$p_s = (\exp\left(LambertW((h_s p_h - 1)/e) + 1\right) - 1)/h_s.$$

In particular, Theorem 2 implies that the optimal power rate does not depend on duration of time slot. Figure 1(A) illustrates that an increase in quality of communication reflected by fading channel gains h_b leads to an increase in the SN throughput. It is interesting that the SN achieves this increase by reducing

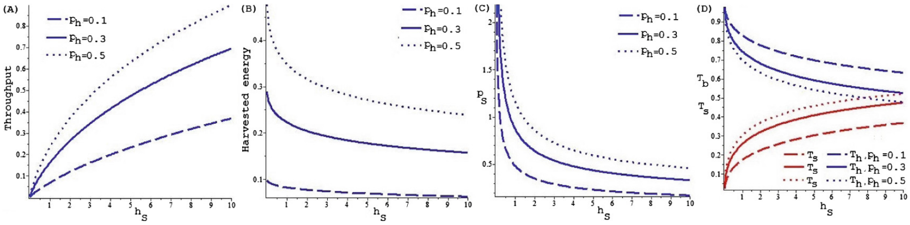


Fig. 1. Throughput, harvested energy, power rate and schedule as functions of h_s.

energy harvesting (Fig. 1(B)). This could be done by applying *energy safe* schedule, namely, by reducing the power transmission rate (Fig. 1(C)) while increasing the duration of transmission (Fig. 1(D)). Thus, here, like in economics where investment in public infrastructure fuels economic growth and attracts new technologies, improving network infrastructure reflected by improving channel gains (e.g. better antennas) encourages using energy safe strategies.

3 The Source Node Sends Data to the Destination Node via a Relay Node

In this Section extend the model for a two-hop scenario where the SN cannot send data directly to the DN but only via a relay node (RN). The RN does not have an external source of energy, and needs to harvest energy from the SN to communicate, while the SN has such external source of energy and can harvest energy straight from it. Thus, the harvested energy is a resource for the SN to communicate with the DN as well as the resource to motivate the RN to cooperative with the SN in performing this task. As incentive to cooperate we consider fair performing of communication tasks by the SN and the RN.

Let us describe the model in detail. We assume that the SN and the RN cannot perform their energy harvesting and communication operations simultaneously. All of the operation takes place within a time slot with duration T. Thus, the time is split into four phases: (a) *energy harvesting by the SN* from an external source (duration T_h), (b) *energy harvesting by the RN* from the SN (duration T_{hr}), (c) *sending data by the SN to the DN via the RN* (duration T_{sd}), (d) *sending data by the RN to the DN* (duration T_{rd}). Thus,

$$T_h + T_{hr} + T_{sd} + T_{rd} = T. \tag{7}$$

Let h_{rd} be the fading channel gains for the RN to send data to the DN. Let h_{sr} be the fading channel gains for the SN to send data or energy to the RN. Let $h_{sd} := h_{sr}h_{rd}/(1+h_{rd})$ be the fading channel gains for the SN to send data to the DN via the RN. Let p_s be the power rate to send either energy supply to the RN or to send data to the DN via the RN by the SN. Let p_r be the power rate to send data by the RN to the DN.

The total energy accumulated by the SN during the energy harvesting phase is:

$$E = p_h T_h. \tag{8}$$

The total energy E_r sent by the SN to the RN during energy harvesting by the RN phase is

$$E_r = p_s T_{hr}, \tag{9}$$

while the total energy $\overline{E}_r$ accumulated by the RN for this phase is

$$\overline{E}_r = \gamma h_{sr} p_s T_{hr}, \tag{10}$$

where γ is the coefficient of energy accumulation.

The total energy used by the SN to send data to the DN via the RN is given as follows:

$$E_{sd} = T_{sd}p_s, \tag{11}$$

which yields the total SN throughput, given as follows:

$$v_s = T_{sd}\ln(1 + h_{sd}p_s). \tag{12}$$

The total energy $T_{rd}p_r$ employed by the RN to send data to the DN has to be equal to the energy $\overline{E}_r$ harvested from the SN, i.e.,

$$\overline{E}_r = T_{rd}p_r, \tag{13}$$

which yields the total RN throughput given as follows:

$$v_r = T_{rd}\ln(1 + h_{rd}p_r). \tag{14}$$

Thus, a dilemma for the SN arises *how much harvested energy has to be used for communication and how much to share with the RN to make it relay the SN's data to the DN*? As an incentive for the SN and the RN to cooperate, we consider the fair performing of communication tasks for each of them, and the α -fairness utility is considered as such a fairness criterion. Thus, the goal of the SN is to find schedule $\boldsymbol{T} = (T_h, T_{hr}, T_{sd}, T_{rd})$ to fulfil fairly each of these communication tasks. In the considered model, the α-fairness utility for these communication tasks is given as follows: $v(\boldsymbol{T}) = (v_r(\boldsymbol{T}))^{1-\alpha}/(1-\alpha) + (v_s(\boldsymbol{T}))^{1-\alpha}/(1-\alpha)$ for $\alpha \neq 1$ and $v(\boldsymbol{T}) = \ln(v_r(\boldsymbol{T})) + \ln(v_r(\boldsymbol{T}))$ for $\alpha = 1$. The α-fair schedule is given as the solution of the following problem:

$$\text{maximize } v(\boldsymbol{T}), \text{ subject to} \tag{15}$$

$$T_h \geq 0,\, T_{hr} \geq 0,\, T_{sd} \geq 0,\, T_{rd} \geq 0, \tag{15a}$$

$$T_h + T_{hr} + T_{sd} + T_{rd} = T, \tag{15b}$$

$$p_hT_h = T_{hr}p_s + T_{sd}p_s, \tag{15c}$$

$$\gamma h_{sr}p_sT_{hr} = T_{rd}p_r. \tag{15d}$$

Note that α-fairness criterion provides a unified framework for considering a wide array of fairness considerations, ranging from maximizing cooperative solution (for $\alpha = 0$) through proportional fairness (for $\alpha = 1$) to the maxmin solution (for α tending to infinity). As a survey on fairness criteria applied in wireless network we refer to [12], while as examples of α-fairness criteria, we refer the reader to [15] for a throughput assignment problem, and to [9] for bargaining over the fair trade-off between secrecy and throughput in OFDM communications. In [5,7,8], maxmin strategies were designed as solution of the corresponding zero-sum games. In [14], in the context of LTE-A networks, cooperative bargaining solutions for resource allocation over the available component carriers was investigated. In [10], bargaining problem over fair performing dual radar and communication task was solved. In [1,6], fair power control was applied for resources allocation by base station under uncertainty. In [3,4], fair channel sharing strategies by WiFi and LTE-U networks were designed.

Theorem 3. *(a) The α-fair schedule $\boldsymbol{T}_\alpha = (T_{h,\alpha}, T_{hr,\alpha}, T_{sd,\alpha}, T_{rd,\alpha})$ is unique and given as follows:*

$$\begin{aligned} T_{h,\alpha} &= T\frac{P_s A_s^{1/\alpha}/L_s + P_s P_r A_r^{1/\alpha}/L_r}{A_s^{1/\alpha-1} + A_r^{1/\alpha-1}},\ T_{hr,\alpha} = T\frac{P_r A_r^{1/\alpha}/L_r}{A_s^{1/\alpha-1} + A_r^{1/\alpha-1}}, \\ T_{sd,\alpha} &= T\frac{A_s^{1/\alpha}/L_s}{A_s^{1/\alpha-1} + A_r^{1/\alpha-1}},\ T_{rd,\alpha} = T\frac{A_r^{1/\alpha}/L_r}{A_s^{1/\alpha-1} + A_r^{1/\alpha-1}}, \end{aligned} \tag{16}$$

$$\begin{aligned} \textit{where } L_s &:= \ln(1 + h_{sd}p_s),\ L_r := \ln(1 + h_{rd}p_r),\ P_r := p_r/(\gamma h_{sr} p_s), \\ P_s &:= p_s/p_r,\ A_s := L_s/(1 + P_s),\ A_r := L_r/(1 + P_r(1 + P_s)). \end{aligned} \tag{17}$$

(b) The following relations hold between the SN throughput $v_{s,\alpha}$ and the RN throughput $v_{r,\alpha}$ corresponding to $\boldsymbol{T}_\alpha$:

$$v_{s,\alpha}/v_{r,\alpha} = (A_s/A_r)^{1/\alpha} \text{ and } v_{s,\alpha}/A_s + v_{r,\alpha}/A_r = T. \tag{18}$$

(c) The maxmin solution corresponds to α tending to infinity and is given as:

$$\begin{aligned} &v_{s,\infty} = v_{r,\infty} = v_\infty := T/(1/A_s + 1/A_r), \\ &\boldsymbol{T}_\infty = \left(\frac{T(P_s/L_s + P_s P_r/L_r)}{1/A_s + 1/A_r}, \frac{TP_r/L_r}{1/A_s + 1/A_r}, \frac{T/L_s}{1/A_s + 1/A_r}, \frac{T/L_r}{1/A_s + 1/A_r}\right). \end{aligned} \tag{19}$$

(d) The cooperative solution corresponds to $\alpha = 0$ and is given as:

$$\boldsymbol{T}_0 = \begin{cases} \left(\frac{TP_s}{1 + P_s}, 0, \frac{T}{1 + P_s}, 0\right), & A_s > A_r, \\ \left(\frac{TP_r}{1 + P_h(1 + P_s)}, \frac{TP_s P_r}{1 + P_r(1 + P_s)}, 0, \frac{T}{1 + P_r(1 + P_s)}\right), & A_s < A_r, \end{cases} \tag{20}$$

$$v_{s,0} = \begin{cases} TA_s, & A_s > A_r, \\ 0, & A_s < A_r, \end{cases} \quad v_{r,0} = \begin{cases} 0, & A_s > A_r, \\ TA_r, & A_s < A_r. \end{cases} \tag{21}$$

Thus, in the cooperative solution, either the SN or the RN has to be a full altruist totally sacrificing its own communication task in the name of reaching the largest joint throughput. While the maxmin solution equalizes both throughput, or, in the other words, it is aimed at the equality of outcomes for the nodes. Thus, the fairness coefficient reflects a trade-off between altruism and equality of outcome. Further, between the throughput as functions of α there is a liner relation (18), and an increase in one throughput yields a decrease in the other Fig. 2(C). Also, in the considered example, for $\alpha = 0$ the RD has to be altruist focusing only on the relaying operation. An increase in α results in (i) a decrease in the SN throughput, (ii) an increase in the RN throughput, and (iii) a decrease in energy harvesting by the SN. The latter means that full altruism of one of the nodes (the RN) leads to employing a less energy safe strategy by the other (the SN); while an increase in selfishness for the SN (reflected by an intention to get a larger throughput for itself) makes the SN reduce energy harvesting and to switch to a more energy safe strategy Fig. 2(B). Also, an increase in α makes the SN spend more time supplying the RN by energy, thereby supporting an increase in duration for the RN communication with the DN.

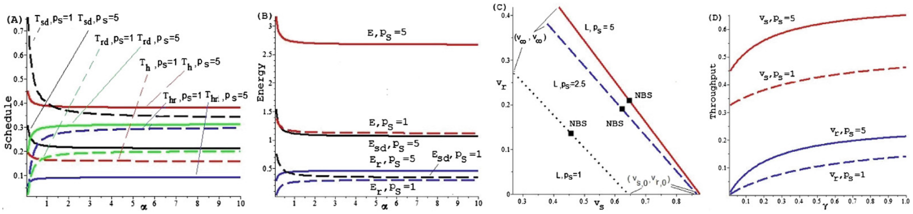


Fig. 2. (A) The α-fair schedule, (B) harvested energies used for harvesting and communication as functions on α, (C) relation between throughput in plane (v_s, v_r) for $T = 1$, $p_h = 7$, $p_r = 4$, $h_{sr} = 3$, $h_{rd} = 0.7$, $\gamma = 0.9$, $p_s \in \{1, 5\}$ and (D) bargaining throughput as function on γ.

4 Trade-off for Fairness Coefficient

Since, for each fixed α the fair solution $\boldsymbol{T}_\alpha$ is derived, a problem arises to find the most fair $\boldsymbol{T}_\alpha$. An answer to this problem will be obtained as the Nash bargaining solution (NBS) [2] over all of the fair throughputs [2]. *First* let us define the feasibility set $\boldsymbol{L}$ for all of the fair throughputs for the SN and the RN, i.e., $\boldsymbol{L} := \{(v_{s,\alpha}, v_{r,\alpha}) : \alpha \geq 0\}$. By (18) and (19), $\boldsymbol{L}$ is the line connecting the point $(v_{s,0}, v_{r,0})$ and the point (v_∞, v_∞). *Second* let $(v_s^d, v_r^d) = (\min\{v_{s,0}, v_\infty\}, \min\{v_{r,0}, v_\infty\})$ be the point composed by minimal throughput in $\boldsymbol{L}$. This point can be considered as *a disagreement point* [2]. Then, the NBS is given as given as $\arg\max\{NP(v_s, v_r) : (v_s, v_r) \in \boldsymbol{L}\}$ where $NP(v_s, v_r) := (v_s - v_s^d)(v_r - v_r^d)$ is called *the Nash product.*

Theorem 4. *The bargaining throughput is unique and given by*

$$(v_s, v_r) = \left(\frac{TA_sA_r}{2(A_s + A_r)}, \frac{TA_sA_r}{2(A_s + A_r)}\right) + \begin{cases} (TA_s/2, 0), & A_s > A_r, \\ (0, TA_r/2), & A_s < A_r. \end{cases}$$

The bargaining value for fairness coefficient is $\alpha = \ln(A_s/A_r)/\ln(v_r/v_s)$.

Figure 2(C) illustrates the NBS for $p_s \in 1, 2.5, 5$, while Fig. 2(D) illustrates that both throughput gains correspond to an increase in the coefficient of energy accumulation γ, and, thus, on an improvement for the RF technology us for energy harvesting.

5 Conclusions

To obtain insight into the cooperation between nodes with different sources of energy (from external sources or through radio frequency transmissions from other nodes), a simple two-hop network model was investigated. First we showed that, much like in economics where investment in public infrastructure fuels economic growth and attract new technologies, improving network infrastructure reflected by improved channel gains fuels the use of energy safe strategies. Then, based on α-fairness a problem of node cooperation was investigated. It is shown that an altruistic strategy for one of the nodes comes as a part of a cooperative solution, while the maxmin strategy is proven to be egalitarian. Using Nash bargaining over the obtained continuum of fair solutions, a trade-off between altruistic and egalitarian behaviors is found. Further, the gains associated with bargaining throughput correspond to an improvement in the RF technology used for energy harvesting.

A Appendix

Proof of Theorem 1. By (1) and (3), $T_s \le p_h T/(p_h + p_s)$. Then, due to v given by (4) is increasing on T_s, (5) follows. ■

Proof of Theorem 2. To find the optimal p_s we have to find derivation of v on p_s: $d\,v(p_s)/d\,p_s = (h_s(p_h + p_s)/(1 + h_s p_s) - \ln(1 + h_s p_s))\, T p_h/(p_h + p_s)^2$. Thus, $dv/dp_s\{>, =, <\}0$ if and only if $1 + a/x - \ln(x)\{>, =, <\}0$, where $x = 1 + h_s p_s$ and $a = h_s p_h - 1$. It is clear that $a > -1$ and $x \ge 1$. For a fixed $a > -1$ the function $1 + a/x - \ln(x)$ is decreasing on $x \ge 1$. Moreover, the equation $1 + a/x - \ln(x) = 0$ has the unique root $x = \exp(\mathrm{LambertW}\,(a/e) + 1)$, and the result follows. ■

Proof of Theorem 3. Since $v(\boldsymbol{T})$ is concave, to find the optimal $\boldsymbol{T}$ the KKT Theorem can be applied. First we define Lagrange function $L_{\omega_1,\omega_2,\omega_3}(\boldsymbol{T})$ with ω_1, ω_2 and ω_3 are Lagrange multipliers as follows:

$$\begin{aligned} L_{\omega_1,\omega_2,\omega_3}(\boldsymbol{T}) = \frac{(T_{rd}L_r)^{1-\alpha}}{1-\alpha} + \frac{(T_{sd}L_s)^{1-\alpha}}{1-\alpha} + \omega_1(T - T_h - T_{hr} - T_{sd} - T_{rd}) \\ + \omega_2(p_h T_h - T_{hr} p_s - T_{sd} p_s) + \omega_3(\gamma h_{sr} p_s T_{hr} - T_{rd} p_r). \end{aligned} \tag{22}$$

Then, for $\boldsymbol{T}$ to be optimal, besides of conditions (15b)–(15d), the following relations have to hold:

$$\partial L/\partial T_{rd} = L_r^{1-\alpha}/(T_{rd})^\alpha - \omega_1 - p_r\omega_3 \begin{cases} = 0, & T_{rd} > 0, \\ \le 0, & T_{rd} = 0, \end{cases} \tag{23}$$

$$\partial L/\partial T_{sd} = L_s^{1-\alpha}/(T_{sd})^\alpha - \omega_1 - p_s\omega_2 \begin{cases} = 0, & T_{sd} > 0, \\ \le 0, & T_{sd} = 0, \end{cases} \tag{24}$$

$$\partial L/\partial T_h = -\omega_1 + p_h\omega_2 = 0, \tag{25}$$

$$\partial L/\partial T_{hr} = -\omega_1 - p_s\omega_2 + \gamma h_{sr} p_s \omega_3 = 0. \tag{26}$$

By (25), we have that

$$\omega_2 = \omega_1/p_h. \tag{27}$$

By (26) and (27), we have that $\omega_3 = (1 + p_s/p_h)\omega_1/(\gamma h_{sr} p_s)$. Then, (17) and (23) yield that:

$$T_{rd} = \frac{L_r^{1/\alpha-1}}{(\omega_1 + p_r\omega_3)^{1/\alpha}} = \frac{L_r^{1/\alpha-1}}{(1 + P_r(1 + P_s))^{1/\alpha}\omega_1^{1/\alpha}} = \frac{A_r^{1/\alpha-1}}{(1 + P_r(1 + P_s))\omega_1^{1/\alpha}}. \tag{28}$$

By (24), in notation (17), we have that

$$T_{sd} = \frac{L_s^{1/\alpha-1}}{(\omega_1 + p_s\omega_2)^{1/\alpha}} = \frac{L_s^{1/\alpha-1}}{(1+P_s)^{1/\alpha}\omega_1^{1/\alpha}} = \frac{A_s^{1/\alpha-1}}{(1+P_s)\omega_1^{1/\alpha}}. \tag{29}$$

By (15d), (17) and (28) the following relation holds

$$T_{hr} = T_{rd}p_r/(\gamma h_{sr}p_s) = P_r T_{rd} = P_r A_r^{1/\alpha-1}/((1+P_r(1+P_s))\omega_1^{1/\alpha}). \tag{30}$$

By (15c), (29) and (30), using notation (17) we have that

$$T_h = \frac{p_s}{p_h}\left(T_{hr}+T_{sd}\right) = P_s\left(T_{hr}+T_{sd}\right) = \frac{P_sP_rA_r^{1/\alpha-1}}{(1+P_r(1+P_s))\omega_1^{1/\alpha}} + \frac{P_sA_s^{1/\alpha-1}}{(1+P_s)\omega_1^{1/\alpha}}. \tag{31}$$

Then, summing (28)–(31) and taking into account (15c) imply that $\omega_1^{1/\alpha} = (A_s^{1/\alpha-1} + A_r^{1/\alpha-1})/T$. Substituting this ω_1 into (28)–(31) implies (16). Then,

$$v_{s,\alpha} = L_sT_{su,\alpha} = TA_s^{1/\alpha}/(A_s^{1/\alpha-1} + A_r^{1/\alpha-1}), \tag{32}$$

$$v_{r,\alpha} = L_rT_{ru,\alpha} = TA_r^{1/\alpha}/(A_s^{1/\alpha-1} + A_r^{1/\alpha-1}). \tag{33}$$

Dividing (32) by (33) yields the first relation in (18). Note that

$$\frac{A_s^{1/\alpha}}{A_s^{1/\alpha-1}+A_r^{1/\alpha-1}} = A_s\frac{A_s^{1/\alpha-1}}{A_s^{1/\alpha-1}+A_r^{1/\alpha-1}} = A_s\left(1-\frac{1}{A_r}\frac{A_r^{1/\alpha}}{A_s^{1/\alpha-1}+A_r^{1/\alpha-1}}\right).$$

This, jointly with (32) and (33), implies the second relation in (18), and the result follows. ■

Proof of Theorem 4: By (21), two cases arise: $A_s > A_r$ and $A_s < A_r$. Let $A_s > A_r$. Then, by (18), (19) and (21), $NP(v_s, v_r) = (v_s - v_\infty)v_r = (TA_s - A_sv_r/A_r - v_\infty)v_r = A_s(TA_s/(A_s+A_r) - v_r/A_r)v_r$. Thus, the $(TA_s - A_sv_\infty/(2A_r), v_\infty/2)$ is the NBS, and the result follows from (19) and the first relation in (18). ■

References

1. Altman, E., Avrachenkov, K., Garnaev, A.: Fair resource allocation in wireless networks in the presence of a jammer. Perform. Eval. **67**, 338–349 (2010)
2. Fudenberg, D., Tirole, J.: Game Theory. MIT Press, Boston (1991)
3. Garnaev, A., Sagari, S., Trappe, W.: Fair channel sharing by Wi-Fi and LTE-U networks with equal priority. In: Noguet, D., Moessner, K., Palicot, J. (eds.) CrownCom 2016. LNICST, vol. 172, pp. 91–103. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40352-6_8
4. Garnaev, A., Sagari, S., Trappe, W.: Bargaining over fair channel sharing between Wi-Fi and LTE-U networks. In: Koucheryavy, Y., Mamatas, L., Matta, I., Ometov, A., Papadimitriou, P. (eds.) WWIC 2017. LNCS, vol. 10372, pp. 3–15. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61382-6_1

5. Garnaev, A., Trappe, W.: The eavesdropping and jamming dilemma in multichannel communications. In: IEEE International Conference on Communications (ICC), Budapest, Hungary, pp. 2160–2164 (2013)
6. Garnaev, A., Trappe, W.: Fair resource allocation under an unknown jamming attack: a Bayesian game. In: IEEE International Workshop on Information Forensics and Security (WIFS), Atlanta, GA, pp. 227–232 (2014)
7. Garnaev, A., Trappe, W.: Secret communication when the eavesdropper might be an active adversary. In: Jonsson, M., Vinel, A., Bellalta, B., Belyaev, E. (eds.) MACOM 2014. LNCS, vol. 8715, pp. 121–136. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10262-7_12
8. Garnaev, A., Trappe, W.: To eavesdrop or jam, that is the question. In: Mellouk, A., Sherif, M.H., Li, J., Bellavista, P. (eds.) ADHOCNETS 2013. LNICST, vol. 129, pp. 146–161. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04105-6_10
9. Garnaev, A., Trappe, W.: Bargaining over the fair trade-off between secrecy and throughput in OFDM communications. IEEE Trans. Inf. Forensics Secur. **12**, 242–251 (2017)
10. Garnaev, A., Trappe, W., Petropulu, A.: Bargaining over fair performing dual radar and communication task. In: 50th Asilomar Conference on Signals. Systems, and Computers, Pacific Grove, CA, pp. 47–51 (2016)
11. Gunduz, D., Devillers, B.: Two-hop communication with energy harvesting. In: 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pp. 201–204 (2011)
12. Huaizhou, S., Prasad, R.V., Onur, E., Niemegeers, I.: Fairness in wireless networks: Issues, measures and challenges. IEEE Comm. Surv. Tutor. **16**, 5–24 (2014)
13. Karalis, A., Joannopoulos, J.D., Soljacic, M.: Efficient wireless nonradiative mid-range energy transfer. Ann. Phys. **323**, 34–48 (2008)
14. Militano, L., Niyato, D., Condoluci, M., Araniti, G., Iera, A., Bisci, G.M.: Radio resource management for group-oriented services in LTE-A. IEEE Trans. Veh. Technol. **64**, 3725–3739 (2015)
15. Mo, J., Walrand, J.: Fair end-to-end window-based congestion control. IEEE/ACM Trans. Netw. **8**, 556–567 (2000)
16. Ozel, O., Tutuncuoglu, K., Yang, J., Ulukus, S., Yener, A.: Transmission with energy harvesting nodes in fading wireless channels: optimal policies. IEEE J. Sel. Areas Commun. **29**, 1732–1743 (2011)
17. Reyhanian, N., Maham, B., Shah-Mansouri, V., Tusher, W., Yuen, C.: Game-theoretic approaches for energy cooperation in energy harvesting small cell networks. IEEE Trans. Veh. Technol. **66**, 7178–7194 (2017)
18. Varan, B., Yener, A.: Throughput maximizing games in the two-hop relay channel with energy cooperation. In: 49th Annual Conference on Information Sciences and Systems (CISS), pp. 1–6 (2015)
19. Xiao, Y., Niyato, D., Han, Z., DaSilva, L.A.: Dynamic energy trading for energy harvesting communication networks: a stochastic energy trading game. IEEE J. Sel. Areas Commun. **33**, 2718–2734 (2015)
20. Yang, J., Ulukus, S.: Optimal packet scheduling in an energy harvesting communication system. IEEE Trans. Commun. **60**, 220–230 (2012)

EEHCCP: An Energy-Efficient Hybrid Clustering Communication Protocol for Wireless Sensor Network

Rohit Pachlor(✉) and Deepti Shrimankar

Department of Computer Science and Engineering, Visvesvaraya National Institute of Technology, Nagpur, India
rohit.pachlor@gmail.com

Abstract. Depending upon the application type, wireless sensor network has to work for months to years with a finite energy source. During this time, sensor nodes sense useful information from their surrounding and transmit it to the Base Station. Due to the sensor's finite energy source, accumulation and transmission of sensed information in an energy efficient manner is very significant. Therefore, energy efficient communication protocols are a major research concern in a wireless sensor network to prolong the network lifetime. This paper presents an Energy-Efficient Hybrid Clustering Communication Protocol (EEHCCP). It is an improved LEACH protocol. It evenly distribute the energy load among all the sensor nodes. It uses a novel parameter average cluster-member energy ($ACME$) to delay the re-clustering time. The experimental results show the effectiveness of the proposed protocol over prominent Low Energy Adaptive Clustering Hierarchy (LEACH) and its descendant protocols in terms of network lifetime and energy consumption.

Keywords: Clustering · Network lifetime · Data aggregation
Energy efficient routing · Scalability · Communication protocols
Wireless Sensor Network

1 Introduction

Wireless Sensor Network (WSN) consist of hundreds to thousands of sensors (nodes) deployed over a target area for environment monitoring or event tracking. The job of nodes is to gather the information about the events occurring in their close surrounding, and periodically communicate it to the Base Station (BS) for further processing [1]. The nodes are energy constrained and possibly deployed in such a manner that the replacement of the energy source is not possible always. Therefore, it is utmost important for a node to efficiently utilize its energy so as not to quickly run out of energy and hence prolongs the network's lifetime. Network lifetime is the time elapsed until the last node (or the first node or prefix percentage of nodes) in the network run out of its energy (dies) [2].

The simplest technique to reduce energy consumption during transmission is to send packets in a multi-hop fashion rather than single-hop to the BS [3].

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 199–207, 2018.
https://doi.org/10.1007/978-3-319-74439-1_18

In the literature clustering based [4–7], chain based [8,9] and tree based [10,11] techniques are discussed to achieve multi-hop communication. In a cluster-based network, fixed percentage of sensor nodes are elected as leader called the cluster head (CH) based on a preset CH selection criteria. CH serves as a BS for cluster members and transmits the aggregated data to the actual BS [4]. Rest of the sensor nodes join one of the CHs based on a preset cluster joining criteria. Each cluster member periodically sends its data to the cluster head, which in turn aggregates the received data before transmitting it to the BS.

The rest of the paper is organized as follows: Sect. 2 describes LEACH and its descendant protocols. Section 3, introduces the proposed EEHCCP protocol. Section 4, presents the performance evaluation of EEHCCP. And finally, Sect. 5 concludes the findings.

2 LEACH and Its Descendant Protocols

LEACH [4] is the most renowned cluster-based communication protocol for WSN. LEACH uses a randomized rotation of cluster heads to evenly distribute the energy load among nodes in the network. The election of cluster head is based on the number of times the node has served as a cluster head and the desired number of cluster heads, determined apriori, for the network. LEACH divides the communication process into rounds. Each round is composed of setup phase and steady-state phase. In the setup phase, each node independently selects a random number between 0 and 1 and if the chosen number is less than a threshold (T) given by Eq. 1, then, the node selects itself as a cluster head for current round.

$$T\left(n\right)=\begin{cases}\frac{p}{1-p*(r mod p^{-1})} & \forall n\in G\\ 0 & \forall n\notin G\end{cases} \tag{1}$$

Where n is a random number between 0 and 1. p is the desired percentage of cluster heads. G is the set of nodes that weren't the cluster heads in the last $1/p$ rounds.

Each cluster head, then, broadcasts an advertisement message to share their status with other sensor nodes. Each node chooses a cluster to which they belong by choosing a cluster head that requires least communication energy. The decision is based on received signal strength of the advertisement message. The one with the largest received signal strength is the cluster head that can be reached using least communication energy. After cluster formation, head of each cluster creates a TDMA schedule for associated cluster members and broadcasts it back to them. In the steady-state phase, each cluster member node transmits its sense data to the corresponding cluster head in their allocated transmission slot and turns off the radio components for rest of the time to reduce the energy dissipation. Once cluster heads receive the data from associated cluster members, they perform aggregation on it before transmitting it to the BS.

LEACH-C [5] is the centralized version of LEACH. The only difference lies in the fact that LEACH-C uses the BS to decide the cluster heads for each round. In the beginning of each round, all nodes send their IDs, location information, and energy status to the BS. The BS then analyzes the received data and selects energy-rich nodes as cluster heads to form optimal clusters using the simulated annealing algorithm. LEACH-F [6] is also the centralized version of LEACH. The only difference lies in the fact that LEACH-F uses Static clusters. The clusters formed in the first round are static and used throughout the network lifetime. In each cluster, the role of cluster head rotates among the associated cluster members.

VCH (Vice cluster head) [12,13] is very much identical to LEACH but it delays the re-clustering time by prolonging the steady-state phase duration. In VCH [13], clusters are static for two consecutive rounds and re-clustering occurs at the end of every second round. In steady-state phase, the current cluster head selects a VCH among cluster members and thus, reduces the clustering setup overhead for next one round.

LEACH and LEACH-C have no control over the clustering frequency. The setup phase is an overhead over the actual transmission of sensed data. Performing clustering in each communication round increases this overhead. LEACH-C suffers from high energy dissipation of nodes due to long distance because at the beginning of each round, each node sends its ID, location, and energy information to the BS. In LEACH-F, clusters are static and only the cluster heads are rotated according to the rotation order given by the BS. Therefore, it may happen that a new cluster head for the next round is located far away from a member node compared to other cluster heads. Consequently, the node has to use the large communication energy cluster head when there is another clusters cluster head nearby.

In this paper we extend the concept of VCH protocol and use a novel average cluster-member energy ($ACME$) parameter to increase the number of rounds that uses the same clusters. Therefore, it delays the re-clustering time to save the setup phase overhead cost.

3 Energy-Efficient Hybrid Clustering Communication Protocol (EEHCCP)

EEHCCP divides the communication process into several rounds. Each round composed of setup and steady-state phase. In each round, EEHCCP rotates cluster heads based on the residual energy of the nodes. EEHCCP incorporates both static and dynamic formation of clusters and selects either of these methods based upon a user control parameter average cluster-member energy ($ACME$). In the very first round of EEHCCP, each node sends its ID, location, and energy information to the BS station. The BS then selects the desired number of cluster heads, determined apriori, and their associated cluster members. The CH selection is based on the location and energy status of all the nodes. The BS, first, divides the network into k-cluster using k-mean clustering, where k, is equal to the desired number of cluster heads. The BS then finds the highest residual energy node for each cluster and appoints them for the job of cluster heads. At last, the BS forgets the cluster formed using k-mean clustering and then, performs re-clustering for chosen cluster heads and determines cluster members for each cluster head using minimum distance criterion so as to minimize the within-cluster sum of squares. The k-mean clustering is used just to find initial cluster heads and not the initial clusters. The k-mean ensures good distribution of cluster heads. After the formation of initial clusters, the BS calculates the $ACME$ and set $ACME$ threshold ($ACME_{th}$) using Eqs. 2 and 3. The $ACME$ parameter decides when to form new clusters: if $ACME$ is below $ACME_{th}$ then new clusters are formed by passing remaining energy status of nodes to the BS. Otherwise, the clusters are static and used for subsequents rounds with new CHs.

$$ACME = \frac{\sum_{i=1}^{K} E(i)_{remE}}{K} \quad \forall i \in C_m \tag{2}$$

$$ACME_{th} = \frac{ACME * P}{100} \tag{3}$$

Where $E(i)_{remE}$ is the residual energy of node i belongs to m^{th} cluster. K is the total number of cluster members in m^{th} cluster. P is the user control parameter varies from 0 to 100.

The BS then broadcasts a message in the network which includes (cluster head) CH's ID for each node, $ACME_{th}$ for each cluster and the TDMA schedule for each cluster. Upon receiving the message each node compares its own ID with the CH's IDs part of the broadcast message to find its role in the cluster. If the CH's ID matches the node's ID, the node becomes the CH and reads the $ACME_{th}$ for its cluster, otherwise, it reads its TDMA slot and goes into sleep mode, to save its energy, till its slot comes. In the data transmission phase, each node sends its data to corresponding CH in its respective TDMA slot. Once the CH receives data from all cluster members, it aggregates the data and sends it to the BS. After the data transmission, each cluster member sends its current energy status to the corresponding CH. The CH of each cluster then calculates the $ACME$ and selects the cluster head for next round using following rules:

1. If the $ACME$ is above the $ACME_{th}$, for all clusters, then the current CH of each cluster selects new CH for the next round from the associated cluster members based on the residual energy such that
   (a) The residual energy of new CH must be greater than current $ACME$ and residual energy of others CMs of the cluster.
2. If no such node is found or the $ACME$ is below the $ACME_{th}$ of any cluster then call for re-clustering.

3. The cluster head whose $ACME$ is below its $ACME_{th}$ broadcast a re-clustering message in the network.
   (a) Each node then sends the same information to the BS as the one sent in the first round.
   (b) BS forms the new clusters and cluster heads as the one formed in the first round.

   After the selection of new CH, the current cluster head of each cluster waits for the re-clustering message. If no such message occurs then the current CH informs all cluster members about the new CH, new TDMA schedule for the next round, and becomes the normal node of the cluster. After this, steady-state phase takes place in order to complete data transmission process.

4 Performance Evaluation

This section presents the performance evaluation of the proposed EEHCCP protocol to show its effectiveness in terms of network lifetime and average energy consumption per round. The performance of EEHCCP has been evaluated using MATLAB simulation and compared with prominent LEACH [4] and its decedent protocols namely, LEACH-C [5], LEACH-F [7], VCH [12]. The BS is located at (50,175) in a 100∗100 m field. The value of network parameters during simulation is specified in Table 1.

Table 1. Network model parameters

Network parameter	Value
Network area	100*100 m^2
Total numbers of nodes (n)	100
CH percentage (p)	0.05
Initial energy of nodes (E_{init})	0.5 J
P (to set ACM_{th})	80
Coefficient for free-space fading (ξ_{fs})	10 pJ/bit/m^2
Coefficient for multi-path fading (ξ_{mp})	0.0013 pJ/bit/m^2
Data packet size	6400 bits
Control packet size	200 bits
Idle state energy ($E_{T,X} = E_{R,X}$)	50 nJ/bit
Data aggregation energy	5 nJ/bit

We ran the simulations to determine the number of rounds of communication when 1%, 10%, 25%, and 50% of the nodes are dead using LEACH, LEACH-C, LEACH-F, VCH and EEHCCP with each node having the same initial energy.

Once the energy of a node goes below zero it is considered dead for the rest of the simulation. Our simulation results, as shown in Fig. 1, show that EEHCCP achieves:

- Approximately 2.2× the number of rounds compared to LEACH when 1%, 10%, 25%, and 50% of the nodes dead.
- Approximately 1.5× the number of rounds compared to LEACH-F when 1%, 10%, 25% and 50% of the nodes dead.
- Approximately 1.3× the number of rounds compared to LEACH-C when 1%, 10%, 25% and 50% of the nodes dead.
- Approximately 1.05× the number of rounds compared to VCH when 1%, 10%, 25% and 50% of the nodes dead.

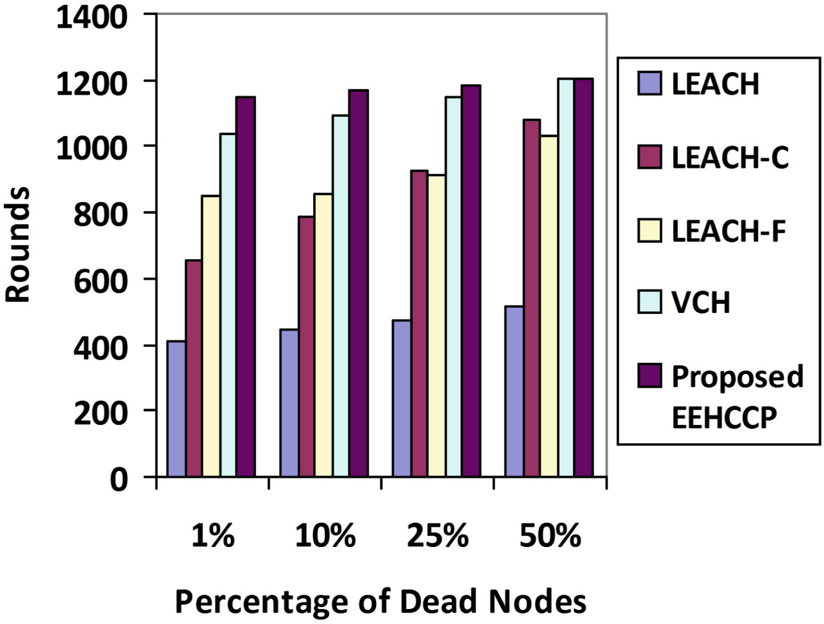


Fig. 1. Network's lifetime

Table 2 shows the number of rounds when 1%, 10%, 25%, and 50% of the nodes dead for different values of P:

Figure 2 shows the number of nodes alive over simulation rounds. The simulation results demonstrate that numbers of active nodes after 1200 rounds are 81, in

Table 2. Performance of EEHCCP for different values of P

P	1%	10%	25%	50%
10	1100	1133	1162	1209
30	1089	1139	1171	1215
50	1099	1145	1169	1214
70	1139	1160	1177	1206
90	1156	1171	1180	1188

EEHCCP protocol, but in the cases of LEACH, LEACH-C, LEACH-F, and VCH, the number of active nodes are 0, 46, 24, and 68 respectively. Figure 3 shows Energy consumption over simulation rounds. The simulation results demonstrate that the energy usages till 700 simulation round are 99.9800%, 60.5197%, 66.7439, 56.3459%, and 55.2500% of the total energy for LEACH, LEACH-C, LEACH-F, VCH, and EEHCCP respectively.

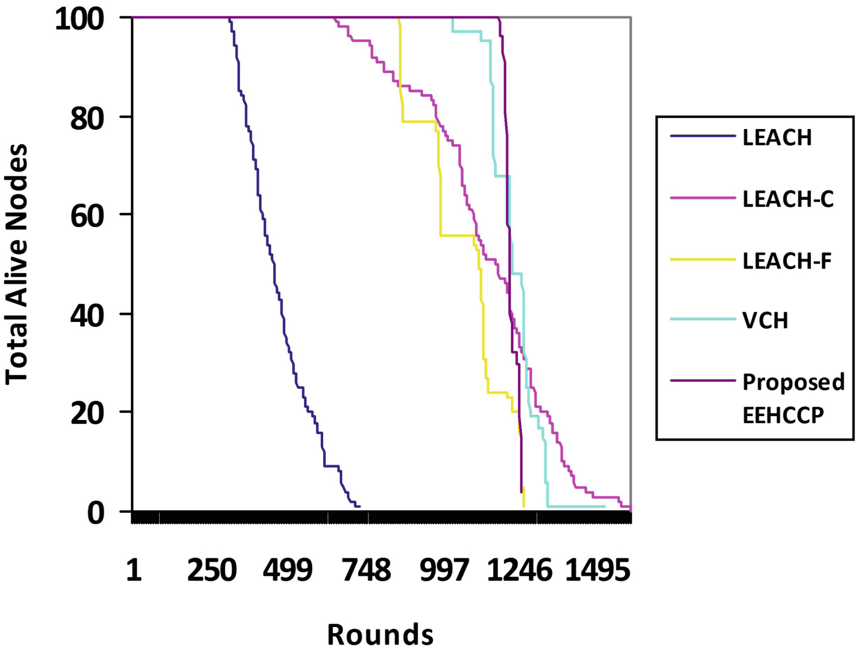


Fig. 2. Number of nodes alive over simulation rounds.

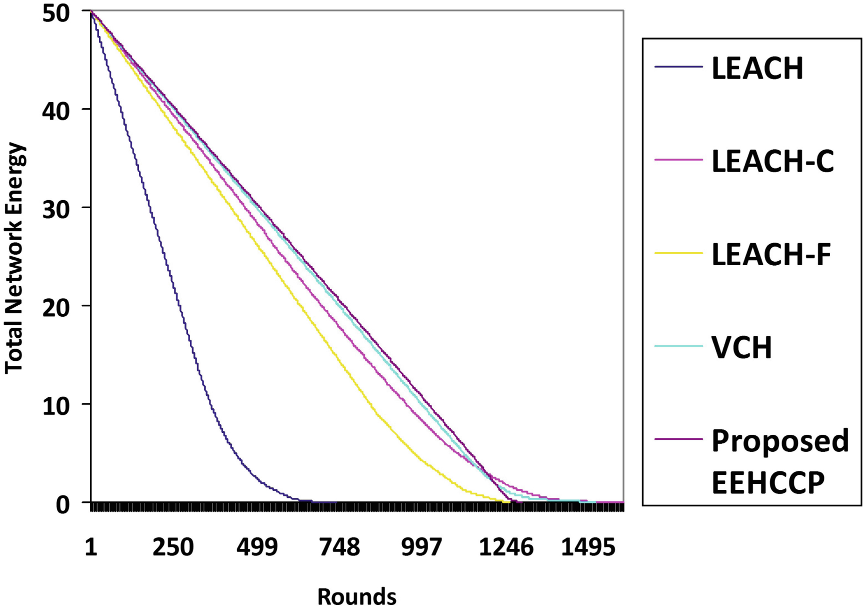


Fig. 3. Energy consumption over simulation rounds.

5 Conclusions

In this paper, we present EEHCCP, a self-organizing protocol which uses static as well as dynamic clusters to evenly distribute the energy load among all nodes to essentially balance the energy consumption on long distances within and outside the cluster. EEHCCP uses k-mean clustering to find well-distributed CHs. EEHCCP uses these CHs to find initial clusters and uses these clusters for communication purpose with a new cluster head for each communication round until their $ACME$ goes below a threshold thus, delays the re-clustering time. As soon as the $ACME$ goes below a threshold for any of the clusters the new cluster heads are selected once again using k-mean.

Our simulation results show that the EEHCCP outperforms LEACH, LEACH-C, LEACH-F and VCH.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Comput. Netw. **38**(4), 393–422 (2002)
2. Younis, O., Fahmy, S.: HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Trans. Mob. Comput. **3**(4), 366–379 (2004)
3. Gong, B., Li, L., Wang, S., Zhou, X.: Multihop routing protocol with unequal clustering for wireless sensor networks. In: International Colloquium on Computing, Communication, Control, and Management, pp. 552–556 (2008)

4. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd International Conference on System Sciences (HICSS 2000), p. 8020 (2000)
5. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Trans. Wirel. Commun. **1**(4), 660–669 (2002)
6. Heinzelman, W.: Application-specific Protocol Architectures for Wireless Networks. Ph.D. Cornell University (2000)
7. Manjeshwar, A., Agrawal, D.P.: TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In: Proceedings of the 15th International Parallel and Distributed Processing Symposium, pp. 2009–2015 (2001)
8. Lindsey, S., Raghavendra, C.S.: PEGASIS: power-efficient gathering in sensor information systems. In: Proceedings of IEEE Aerospace Conference, pp. 3:1125–3:1130 (2002)
9. Chen, K.H., Huang, J.M., Hsiao, C.C.: CHIRON: an energy-efficient chain-based hierarchical routing protocol in wireless sensor networks. In: Wireless Telecommunications Symposium, pp. 183–187 (2009)
10. Satapathy, S., Sarma, N.: TREEPSI: tree based energy efficient protocol for sensor information. In: IFIP International Conference on Wireless and Optical Communications Networks, p. 4 (2006)
11. Park, Y., Jung, E.-S.: Plus-tree: a routing protocol for wireless sensor networks. In: Szczuka, M.S., et al. (eds.) ICHIT 2006. LNCS (LNAI), vol. 4413, pp. 638–646. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77368-9_62
12. Mehmood, A., Lloret, J., Noman, M., Song, H.: Improvement of the wireless sensor network lifetime using LEACH with vice-cluster head. Ad Hoc Sens. Wirel. Netw. **28**(1–2), 1–17 (2015)
13. Zhao, F., Xu, Y., Li, R.: Improved LEACH routing communication protocol for a wireless sensor network. Int. J. Distrib. Sens. Netw. **2012**, 1–6 (2012). https://doi.org/10.1155/2012/649609

A Model for Self-deployment of Autonomous Mobile Sensor Network in an Unknown Indoor Environment

Khouloud Eledlebi, Dymitr Ruta, Fabrice Saffre, Yousof Al-Hammadi, and A. F. Isakovic(✉)

Emirates ICT Innovation Center (EBTIC),
Khalifa University of Science and Technology, P.O. 127788, Abu Dhabi, UAE
{khouloud.edlebi,dymitr.ruta,abdel.isakovic,fabrice.saffre, yousof.alhammadi}@kustar.ca.ae

Abstract. Wireless sensor networks (WSN) emerge at the center of the fast expanding Internet of Things (IoT) revolution, and hence increased research efforts are being directed towards its efficient deployment, optimization and adaptive operation. Rapid deployment of WSN in an unknown open environment is a critical challenge that involves finding optimal locations for the network nodes to deliver optimally balanced sensing and communication services at the maximum possible coverage subject to complex mutual constraints. We address this challenge with a variant of the Voronoi-based algorithm that leverages the converged movement towards Voronoi cells' centers with the intelligent nodes' provisioning algorithm to deliver fully automated and autonomous WSN that rapidly self-deploys itself to any finite indoor environment without using any prior knowledge of the size and structure of the target space. Sequential provisioning supports realistic implementation that accounts for collision avoidance and mitigates the risk of wasteful over-deployment. The preliminary comparative simulation results carried out in a simplified environment indicate very fast convergence to the well balanced WSN at the fairly small deployment cost and thereby validate our model as a very promising compared to the previous approaches to WSN deployment.

Keywords: Mobile sensor nodes · Self-deployment network
Voronoi-based algorithm · Coverage optimization

1 Introduction

Wireless sensors networks have gained a lot of attention since 1990s and been introduced into several fields of technology [1–5]. They are becoming a dominant part in military and civil sensing and communication applications, as well as a part of healthcare and environmental monitoring [1–6].

The ability of a coordinated set of sensors to monitor a particular area efficiently and effectively are at the center of sensor networks' coverage capability [2]. The fragmented information about the monitored area collected by individual nodes requires effective network connectivity in order to be mutually exchanged and efficiently transferred towards the data sink. In order to achieve flexible and robust communication systems

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 208–215, 2018.
https://doi.org/10.1007/978-3-319-74439-1_19

of mobile nodes, the sensors should remain within the communication range of each other so that all the nodes maintain connectivity readiness with multiple neighboring nodes and the network is fully connected.

Many challenges arise in achieving the goal of optimal coverage with moving nodes, while simultaneously trying to minimize consumed energy and time spent on optimizing the WSN of ever smaller and versatile mobile nodes [5, 6]. A well designed WSN should allow sensors to operate autonomously through local operation policy, efficient communication with the local neighbors only, remain independent of sensor's initial position, adaptable and flexible – adjusting rapidly to the changing conditions.

In this report, we introduce a specific variant of the Voronoi-based coverage algorithm [7, 8] augmented with specific nodes provision procedure, which allows mobile nodes to be distributed efficiently and conflict–free inside any unknown two-dimensional bounded (indoor) area of interest. Compared to the other approaches like virtual force-based algorithms [5], our approach achieves the stable, converged, fully connected network much sooner while investing much fewer deployment energy measured by the total amount of nodes displacement from the entry point. The network composed of homogeneous nodes converges to the near-perfectly balanced, uniformly distributed, blanket-coverage WSN only slightly distorted around the bounds of the target area. The most robust feature of our agile WSN deployment method is that is fully decentralized, autonomous and can rapidly self-adjust to any bounded indoor area without any prior knowledge of the geometrical structure of the target space. The reflection of this is that the number of nodes involved in the deployment process is unknown at the start but is dynamically determined by gradual and conflict-free absorption of as many nodes as required to achieve full connectivity.

We evaluate the deployment performance of our algorithm based on two metrics: (1) the total distance travelled by nodes from their starting point until their final allocation, additionally compared to the absolute optimal possible, and (2) the degree of coverage achieved by network measured by the percentage of the total target area that is within sensing reach nodes in terms of area covered.

The rest of the report is organized as follows: Sect. 2 covers previous related work; and Sect. 3 discusses our Voronoi-based algorithm in more details. Section 4 reveals the simulation results and comparison with previous work results; and Sect. 5 finalizes the paper with conclusions and future work plans.

2 Discussion of Related Work

Researchers have gained a substantial inspiration through analysis of animal behavior in coverage of their territories [11, 12]. Animal strategies of self-organization within the surrounding environment, reaching a dynamic steady state coverage, have inspired the application of modelled versions of such behaviors in mobile networks. Voronoi tessellations are among the behaviors observed in such studies. The authors in [11, 12] discussed the behavior of animal territories such as male *tilapia mossambica* and how they form the polygonal shapes at high densities. Assuming all members of the species have same strength, the pressure between the neighbors balances their perpendicular bisector and creates the boundary between them, which lead to formation of Voronoi polygons, whereby each member of the flock tends to occupy its immediate surroundings in a way to be as far from its neighbors as possible. By applying the behavior of such territories into mobile network, the sensor nodes tend to achieve uniform distribution as the particles are at the center of mass of their Voronoi polygonal region. The overall Voronoi regions may not be equally sized, but the total covered area and the number of edges follows some computable definite distribution with finite standard deviation [12].

The Voronoi tessellation technique improves the WSN network uniformity and enhances the system lifetime and energy usage, especially when dealing with homogeneous sensor nodes (nodes of same properties and capabilities). The authors in [9, 10] implemented node-spreading Voronoi algorithm (NSVA) on the sensor nodes to move them toward their Voronoi centroid inside an unknown area of interest. The tessellation cells are bounded by the nodes' sensing range to maintain connectivity during the deployment. Their NSVA algorithm showed to be efficient in terms of average distance travelled by the nodes in comparison with another technique based on genetic algorithm, which they applied as well.

Another approach is to rely on laws of physics to arrange sensor nodes, rather than Voronoi tessellations, so in [13] an algorithm is implemented based on the equilibrium of molecules where the nodes move according to distance-dependent forces coming from their neighbors. The new position for each node is indicated by summing all forces affecting the nodes. The process reach its end if one of the following conditions is met: either nodes move less than a certain threshold value, or if the nodes keep moving back and forth around the same location for many times.

Our algorithm is based on re-calculating Voronoi diagrams every time a new sensor node enters the area of interest; and the movement of these nodes is guided towards the center of their cells. The differences from [9, 10] are: (1) the Voronoi cells are generated after discovering dimensions of the area, and (2) not bounded by the nodes sensing range based on information about their neighbors. In addition to that, nodes will enter one at a time to the area based on requirements other than having a fixed number of nodes. The communication range and sensing range are not identical but follow a certain relationship. More details are in the following section.

3 Voronoi-Based Algorithm

Our newly introduced technique for self-deployment of mobile sensor nodes is based on Voronoi tessellations with the goal of maximizing coverage while maintaining connectivity inside the area of interest [7, 8]. We call it Bio-Inspired Self-Organized Network (BISON) algorithm. The assumptions for BISON are the following:

1. All the nodes are homogeneous, having the same sensing, communication computation and mobility abilities. This helps greatly in optimizing the network lifetime and reducing the overall power consumed.
2. The deployment of the sensor nodes happens one at a time from either edge of the area to make the experiment more realistic.

3. Each node has the ability to distinguish its location and other nodes' locations through GPS and multilateration method, in order to ease the process of plotting Voronoi diagrams with respect to available sensor nodes in the area [14].
4. The communication range (R_C) and sensing range (R_S) form circular shapes with the following relationship between them [13, 15, 16]:

$$R_C = \sqrt{3} R_S \tag{1}$$

   This relation aims to provide large communication abilities among sensor nodes while minimizing the overlapping between their sensing ranges; which also helps in covering as much as possible from the interested area, without wasting resources of adding more nodes to the system.
5. No errors are assumed to be available in terms of determining location of nodes or in data transmission.

The self-organized deployment process starts with launching two sensor nodes from the starting point (area entry) to discover the space dimensions and broadcast the information to the incoming nodes. A third and following sensor nodes are inserted to follow standard Voronoi Center-Mass guided transition policy throughout the explored area i.e. they move towards the centers' of their corresponding Voronoi polygons.

The Voronoi boundaries are determined based on the following definition:

$$V_i = \left\{ x \in A{:}d\left(x, n_i\right) < d\left(x, n_j\right) \right\} \tag{2}$$

where *A* is the area of interest, *x* is a random point inside the area, V_i is the Voronoi cell that belongs to sensor node n_i. The process of moving nodes towards their Voronoi centroid is simply a sequence of identical operations of recalculating the center of mass for each updated Voronoi polygon and moving to it from node's previous location. This process stops when the sum of movements of all nodes become smaller than a certain threshold value related to the size of the explored space i.e. 1% of the space length.

Each new sensor is inserted to join the coverage network if all of the following conditions are met:

1. Any node has its nearest neighbor node further than the communication range R_C (i.e. not all nodes are within the communication range)
2. The distance between the entry point and its closest node is greater than the sensing range R_S (i.e. the network evolved to make space for new entry)

These conditions will ensure mutual communication connectivity among all nodes and avoid premature over provisioning that create inefficiently crowded entry point area.

The results presented in the following section prove that our algorithm allowed nodes to be distributed uniformly covering the entire area after travelling much shorter transition distances compared to the algorithms discussed in [9, 10].

4 Simulation Results

Our simulation analysis is done using MATLAB. The assigned parameters are sensing range $R_S = 1$, area of 10 × 10, and the shifting threshold of $R_S/100$. The code allows us to monitor the movement of nodes until reaching optimum positions, while calculating the required time, number of nodes, total distance travelled, and overall coverage achieved. The simulation is repeated 20 times and the results are averaged. Figure 1(a) shows the two nodes discovering the area of interest in order to broadcast the dimensions to the incoming nodes. In Fig. 1(b) and (c) more nodes are entering the area and are being distributed to maintain coverage and establish connectivity among them. Figure 1(d) shows the final distribution of nodes in the area, and the lines connecting the nodes together represent how many nodes are connected in the system. Our approach required 43 nodes for the given geometry, all of which are connected to at least one neighboring node. This implies that our system is reachable by any node.

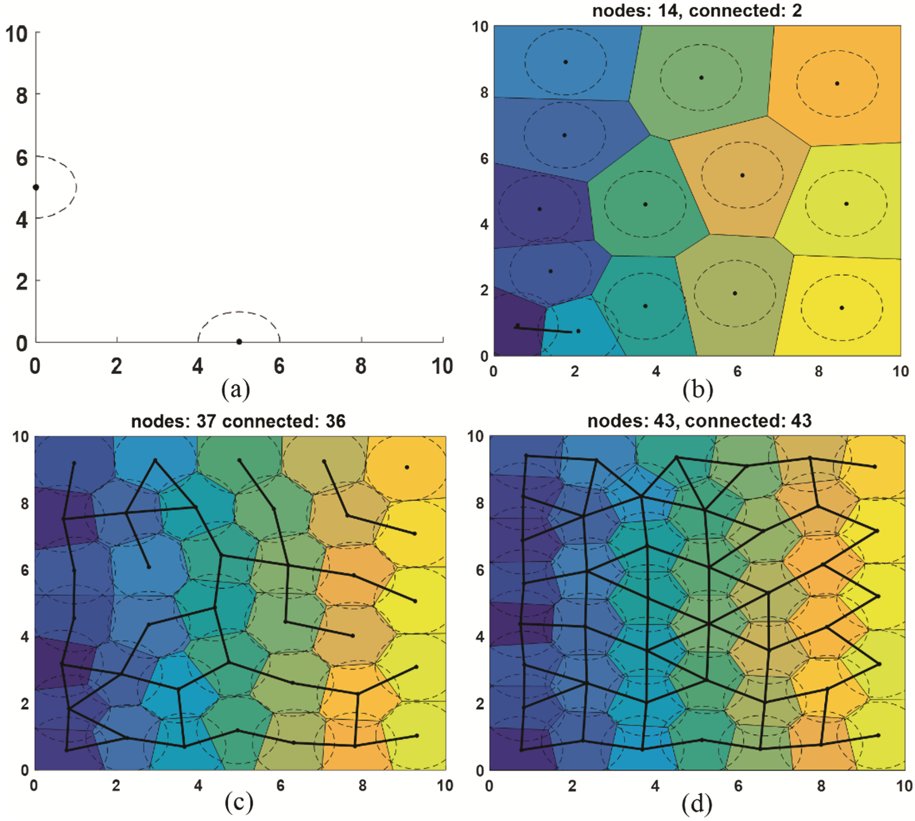


Fig. 1. Snapshots of the distribution of nodes throughout the simulation. (a) Two starting nodes discovering the area of interest. (b) and (c) The distribution of nodes entering the area during intermediate stages, where (c) comes after (b) in timeline, (d) the final distribution of nodes covering the entire area and maintaining connectivity among them

The metrics of evaluation computed to evaluate the system are the following:

$$\textit{percentage area coverage} = \frac{\textit{total area covered by sensor nodes}}{\textit{area of interest}} \tag{3}$$

$$\textit{Average distance travelled}(t) = \frac{1}{N}\sum\nolimits_{i=1}^{N} d\left(L_{n_i}^{0}, L_{n_i}^{t}\right) \tag{4}$$

Where N is total number of nodes at time t, $L_{n_i}^{0}$ is the initial location of nodes at time 0, and $L_{n_i}^{t}$ is the node's location at time t.

Figure 2(a) clearly shows how our algorithm on average requires much shorter distance travelled to find the optimal nodes' positions compared to NSVA algorithm [9, 10], which greatly helps in increasing the system lifetime and reduce the consumed deployment power. Figure 2(b) demonstrates the time evolution of the coverage area of our BISON network during the deployment and compares it to the equivalent process realized by NSVA algorithm. The percentage area coverage is found from the superposition of coverage areas of each node within its sensing range and compared to the total area of the target space. We have used a combination of geometric calculus and the Monte Carlo approximations to arrive at the results that are accurate to the 10th of a percent, allowing for a meaningful comparison.

The results illustrated in Fig. 2 clearly prove that our BISON proceeds with the much more efficient deployment of the network coverage for the target space, overall requiring several times shorter sum of total transitions of the network nodes. Moreover, while both algorithms eventually reach similar final levels of target area coverage, BISON achieves it much quicker, typically maintaining between 10% and 20% advantage in the relative coverage throughout most, especially earlier stages, of the deployment process. The reason behind the unsmoothed curve for BISON algorithm is due to the insertion of sensor nodes throughout the simulation, which in turn changes the distribution of the existing sensor nodes in the target area.

In order to examine the distribution of the typical polygonal shapes obtained with the BISON, a histogram for the number of Voronoi polygon edges vs. relative

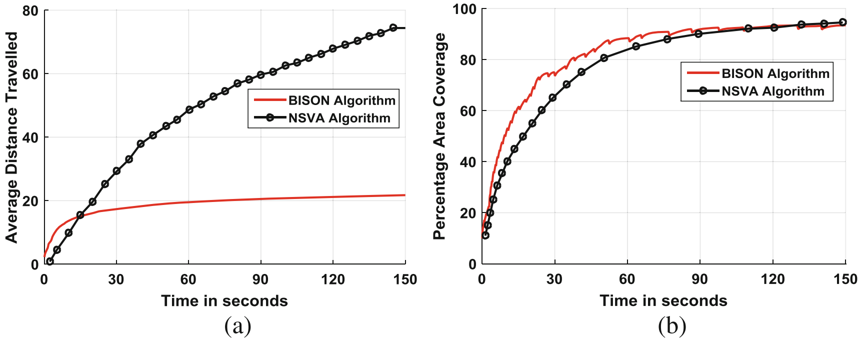


Fig. 2. A comparison between (a) the averaged travelled distance and (b) the relative coverage area between the BISON and NSVA algorithms throughout the simulation.

frequencies is plotted and compared it with the coverage of animals' territories behavior in [12]. The results in Fig. 3 revealed similar behavior to [12] in terms of distribution of the number of edges, and further examination of a larger scale systems indicates that BISON follows uniform distribution. The figure reveals a distribution of particles following a definite distribution with finite standard deviation [12].

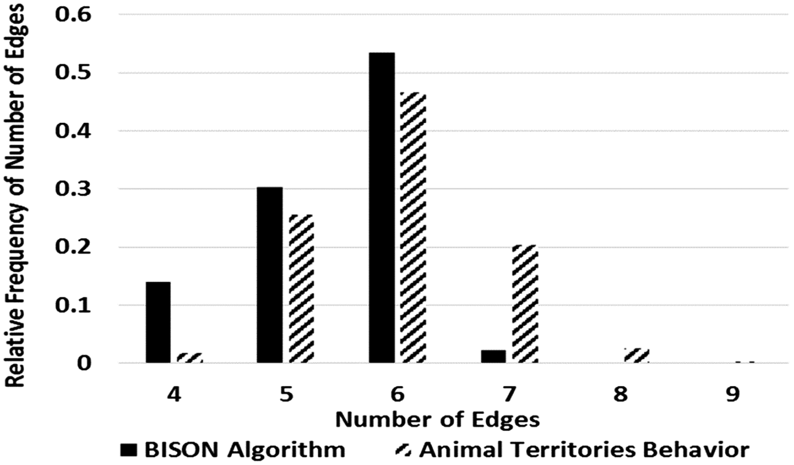


Fig. 3. A comparison of the number of end-state Voronoi polygon edges between BISON algorithm and the known animal territories behavior [12].

5 Conclusion and Future Work

A modified Voronoi algorithm based technique (BISON) is developed establishing a blanket coverage for an unknown area over a short time with minimum distance travelled by mobile sensor nodes. BISON is based on inserting sensor nodes one after the other until achieving optimum coverage while maintaining connectivity among them. The nodes' movements are directed toward their Voronoi center of mass and they stop when the average sensors shift is below a defined threshold. The performance of the network is compared with recent previous work called node-spreading Voronoi algorithm (NSVA) and evaluated based on percentage area coverage and average distance travelled by the nodes. The simulation results revealed that BISON algorithm achieved higher coverage with much less average distance travelled by nodes compared to NSVA algorithm.

Our future work will focus on managing the distance between the sensor nodes together in terms of balancing between approaching the Voronoi center of mass and considering the maximum distance where nodes can be far from each other but still be connected. We will also consider 3D environment, where our algorithm is expected to rapidly self-compose or reorganize itself based on continuously changing conditions.

Acknowledgements. We acknowledge the support from UAE ICT Fund grant on "Biologically Inspired Self-organizing Network Services". DR and AFI acknowledge inspiring exchange with Dr. Vijay Kumar (University of Pennsylvania).

References

1. Chuan, Z., Chunlin, Z., Lei, S., Guangjie, H.: A survey on coverage and connectivity issues in wireless sensor networks. J. Netw. Comput. Appl. **35**(2), 619–632 (2012)
2. Poudyal, L., Sen, B.: A survey on localization and covering techniques in wireless sensor networks. Int. J. Comput. Appl. **67**(7), 23–27 (2013)
3. Amitabha, G., Sajal, D.: Coverage and connectivity issues in wireless sensor networks: a survey. Pervasive Mob. Comput. **4**(3), 303–334 (2008)
4. Nema, S., Shukla, N.: A review on coverage factors in wireless sensor networks. Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET) **2**(12), 1–5 (2013)
5. Zhang, H., Hou, J.C.: Maintaining sensing coverage and connectivity in large sensor networks. Ad Hoc Sensor Wireless Netw. **1**, 89–124 (2005)
6. Maraiya, K., Kant, K., Gupta, N.: Application based study on wireless sensor network. Int. J. Comput. Appl. **21**(8), 9–15 (2011)
7. Mahboubi, H., Aghdam, A.G.: Distributed deployment algorithms for coverage improvement in a network of wireless mobile sensors: relocation by virtual force. IEEE Trans. Control Netw. Syst. **99**, 1–19 (2016)
8. Jing, L., Ruchuan, W., Haiping, H., Sun, L.: Voronoi-based coverage optimization for directional sensor networks. Wirel. Sensor Netw. **1**, 417–424 (2009)
9. Kusyk, J., Zou, J., Gundry, S., Sahin, C.S., Uyar, M.U.: Metrics for performance evaluation of self-positioning autonomous MANET nodes. In: 35th IEEE Sarnoff Symposium (2012)
10. Zou, J., Kusyk, J., Uyar, M.Ü., Gundry, S., Sahin, C.S.: Bio-inspired and Voronoi-based algorithms for self-positioning autonomous mobile nodes. In: IEEE Military Communications Conference, MILCOM 2012 (2012)
11. Du, Q., Faber, V., Gunzburger, M.: Centroidal Voronoi tessellations: applications and algorithms. SIAM Rev. **41**(4), 637–676 (1999)
12. Hasegawa, M., Tanemura, M.: On the pattern of space division by territories. Ann. Inst. Stat. Math. **28**(1), 509–519 (1976)
13. Heo, N., Varshney, P.K.: A distributed self spreading algorithm for mobile wireless sensor networks. In: 2003 IEEE Wireless Communications and Networking Conference: The Dawn of Pervasive Communication, WCNC 2003, New Orleans (2003)
14. Sohrabi, K., Manriquez, B., Pottie, G.J.: Near ground wideband channel measurement. In: Proceedings of the 49th Vehicular Technology Conference, Houston, TX, 16–20 May 1999
15. Wang, Y.-C., Hu, C.-C., Tseng, Y.-C.: Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks. In: First International Conference on Wireless Internet (WICON 2005) (2005)
16. Falcon, R., Li, X., Nayak, A.: Carrier-based coverage augmentation in wireless and robot networks. In: 2010 IEEE 30th International Conference on ICDCSW (2010)

Author Index