Policy-Based Monitoring and High Precision Control for Converged Multi-gigabit IP Networks

Taesang Choi, Sangsik Yoon, Sangwan Kim, Dongwon Kang, and Joonkyung Lee

BcN Division, ETRI
Daejeon, Republic of Korea
{choits,ssyoon,wanni,dwkang,leejk}@etri.re.kr

Abstract. High speed real-time precise traffic monitoring and control has gained significant interests by researchers in this field recently. This is mainly due to the fact that precise traffic measurement and analysis in a high-speed IP network environment is not a simple task. Let alone, control of such traffic in real-time is another big challenge. This paper addresses such complicated issues and proposes a novel solution which can precisely measure traffic in high-speed IP networks, classify them per application, create detailed flow-aware traffic information, and control per application basis. It includes motivation, architecture, and mechanisms. We have embedded capabilities of lossless packet capturing, deep packet inspection, and flow generation into hardware level. We describe major features, design concepts, implementation, and performance evaluation result.

Keywords: policy-based monitoring and control, hardware-based flow record generation, traffic measurement.

1 Introduction

Recently, there have been various efforts such as research, development, and standardization to address issues for true convergence services and its corresponding infrastructure. Notable examples are 3GPP's efforts of defining IMS specifications, ETSI TISPAN"s efforts of fixed mobile convergence, ATIS's efforts of NGN, and finally ITU-T's efforts of NGN. Among them, ITU-T's NGN efforts are most comprehensive which coordinates all the rest activities into a single convergence platform for NGN. Convergence efforts were addressed in three perspectives: transport, service, and application aspects. In terms of transport aspect, wireless, mobile, and fixed transport network technology were converged into a common infrastructure and transport control. For service aspect, common service capabilities are defined and a service control framework is built over such capabilities. Applications take advantage of such underlying common capabilities and framework so that a variety of converged services and applications can be developed. True convergence in terms of requirements, architecture, and capabilities is still undergoing process. Cooperation between policy-based resource control and monitoring is one good example that is under study among experts in the relevant groups. QoS-aware mobility management is another example.

C.S. Hong et al. (Eds.): APNOMS 2009, LNCS 5787, pp. 112–121, 2009.

Convergence is an essential part of the next generation networks in the current market. The real requirements for convergence applications and services are coming from the NGN service providers. Mergers among fixed telecommunications service providers and mobile service providers are happening in the global market. Vertical applications, therefore, no longer satisfy new customer's needs. Service providers themselves are actively seeking new service opportunities by leveraging such mergers. This introduces other complexities in their management. For example, traffic monitoring and QoS resource control were separate issues before. However, integrated management of the two becomes essential to provide high quality converged services. Real-time feedback of the monitored results to QoS resource control is no longer optional but mandatory capability for the fixed mobile convergence environment. This requires convergence of two technologies: traffic measurement and management and traffic resource control. Up to now, these fields have been developed independently in terms of research, development, and standardization. To meet the newly emerging requirements, however, tightly-coupled coordination among them is very important. Efforts on the research and development of these fields have been quite active and various solutions were introduced in the relevant market. Traffic measurement solutions which support upto 40 Gbps speed and volume are introduced. Resource admission control solutions handling upto 10 Gbps are available from various vendors. However, research and development on the convergence of the above mentioned technology are still pre-mature stage. Noting such requirements, we have conducted research and development recently. Our research focus was on the improvement of both hardware and software capabilities to deal with high speed and volume traffic measurement and real-time traffic control.

For measurement of high speed and volume traffic, there were various research efforts to address this issue with software improvement in the host OS kernel I/O driver level. It showed significant improvements but it has limitations which can't be solved without the help of hardware acceleration for the high-speed measurement. Thus, dedicated hardware has to be designed to meet such high speed and volume measurement requirements. Major capabilities that have to be considered in the hardware are high speed reliable packet processing including no loss/duplication packet capture, multi-layer filtering, and various sampling methods (fixed, probabilistic, or flow sampling). None of the currently available hardware whether it is a card or a standalone device can meet all the requirements mentioned above. Especially when it has to deal with very high speed links such as OC-48 or above.

The control of such traffic in real-time with high accuracy is another challenge. This is because high precision control requires support of accurate traffic measurement and classification of applications. Real-time traffic control especially in-line situation can cause serious problems to application services if it is not administered appropriately. Due to the diversity of the current and newly emerging converged applications such as various P2P-based overlay applications, UCC, and IPTV, the main difficulties come from highly dynamic nature of the development and the use of the applications. They are port number independent and asymmetric nature of application transactions. This means that distinguishing flows based on a port number and other header properties is not safe and accurate enough. Also an application transaction can consist of multiple sub-transactions, a series of Requests and Replies, which may follow different routing paths. Accurate flow identification for such a case requires distributed monitoring and correlation of sub-transactions which appeared in different paths. In this paper, we propose novel mechanisms for such challenges. It consists of

hardware and software methodologies. For performance and scalability, packet inspection, filtering/sampling, traffic anomaly handling, flow generation, and traffic management for control are all conducted in our novel hardware. Various traffic analysis including multi-path correlation and high precision applications recognition is the job of our software. We have incorporated the proposed mechanisms into our proof-of-concept system called Wise[HITMAS].

The paper is organized as follows. Section 2 examines related work. Section 3 describes our novel hardware and software methodology and implementation architecture to meet the above challenges. A proof-of-concept system and deployment experiences are explained in Section 4. Finally, section 5 concludes our effort with potential future work.

2 Related Work

There have been many research and development efforts in the field of traffic measurement and analysis for the past decade. As a result, many tools were introduced to meet various objectives such as traffic profiling, traffic engineering, attack/intrusion detection, QoS monitoring, and usage-based accounting.

Various researches were conducted for high-speed traffic measurement based on Network Processors.[1] Their main focus is usually on a specific problem domain like performance study of efficient sampling and filtering algorithms, especially flow sampling.[2] Such algorithms can be important to study but is one part of large set of problems for solving high-speed traffic measurement. We have adopted some of the efficient algorithms (e.g., multi-stage filtering [3]) in our system.

For the precise recognition and classification of applications, most existing solutions are targeted for P2P, streaming applications identification, or security anomaly detection. Cisco Systems' NBAR (Network-Based Application Recognition) provides basic application recognition facilities embedded in their Internet Operating System (IOS) for the purpose of traffic control. Most intrusion detection systems are also equipped with basic application recognition modules which usually function on a signature matching-basis. For streaming and P2P application identification, Mmdump[4] and SM-MON[5] used payload inspection method. Many other methods for P2P traffic characterization [6] based on port number matching were also published. Cisco's Service Control Engine (SCE) series and Netintach's Packetlogic provide more general purpose applications recognition solutions. These systems, however, exhibit shortcomings of performance, scalability, scope of coverage, and accuracy. For resource control solutions, there are several RACF implementations available in the market from vendors such as Huawei, Alcatel-Lucent, ZTE, and Operax[7] to name a few. These solutions are specifically targeted for complying with ITU-T Y.2111[8] Recommendation.

3 Novel Methodology for High-Speed Measurement and High-Precision Control

3.1 High Level System Overview

Based on the motivation described above, we have designed our system with challenging objectives such as no loss/duplication packet capturing up to 40Gbps speed,

L2/L3/L4 filtering, packet and flow sampling, deep packet inspection, application flow generation at the hardware level, and traffic management capabilities such traffic shaping, policing, dropping, and scheduling. It is based on NP and its associated co-processors for DPI, regular expression engine, etc. Before we describe them in details, we briefly explain high level system architecture depicted in Fig. 1. The system is built as ATCA(Advanced Telecom Computing Architecture) compliant board with Network Processors and Co-processors interconnected by SPI 4 switch module and upto 40 Gbps fabric interface and flexible I/O modules which can adapt various combinations such as multiple 1G Ethernet, 2.5G POS, 10G and 40G Ethernet.

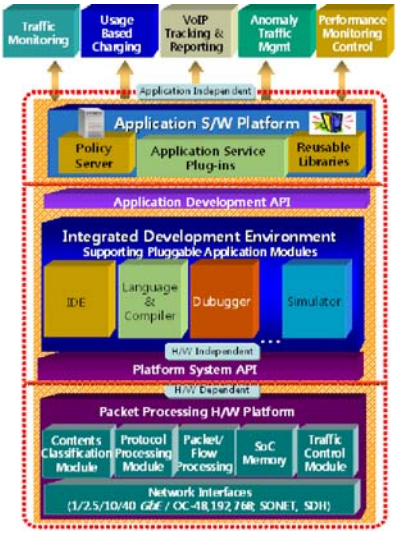

Fig. 1. High-level System Architecture

It consists of packet processing H/W platform, integrated software development environment, Application S/W platform, and platform system/application development APIs. The main design goal of the system is to provide hardware-based reliable performance and flexible software development environment so that various applications can be built in a timely manner to meet the real market requirements. For this we provide hardware dependent API and independent API which relieves hardware dependencies for the software development. It also aims to provide easy to develop IDE including high-level language and an associated compiler for system software programmers. Besides the ease of development, it allows multiple instances of applications running at the same time on the same platform. Duo to such feature, an integrated application described in this paper can be built. Finally, faster customization to meet the customer requirements is possible with the platform. In the following subsections, we describe the major novel methodology to address the above mentioned challenges.

3.2 Traffic Filtering

Traffic filtering is used in our system as a front-end module for various purposes. Our system is targeted to accommodate as many flows as possible in order to meet high-precision requirement. However, lossless traffic capturing on the link of OC-48 or higher speed is practically almost impossible especially when abnormal situation (e.g., syn flooding, DDoS attack, etc.) occurs. In fact, we don't need to capture all flows that may include traffic usage of less interest in terms of flow analysis such as virus traffic and miscellaneous control traffic (e.g., keep alive messages). Our traffic filtering functionality handles such situation. It filters out traffic which does not need flow record generation and, instead, creates a single aggregate flow record for keeping usage statistics information.

3.3 Smart Flow Sampling

Besides one-packet flow aggregation, we need more intelligence for collecting flows in extreme situations. There are many different ways of sampling traffic: simple periodic packet sampling to intelligent flow sampling. We defined a smart flow sampling based on a extended multi-stage filtering algorithm [3]. We use multiple stages of hash tables to keep tract of flows of a certain size. If a flow meets the criteria, it is selected in a stage one filter and continues until it satisfies all stages of filters. If succeeded, it is finally stored in a flow table. This reduces possibility of false judgment. When the decision on flow selection is made, we add additional criteria. Since the signature detection is performed before filtering, particular flows which meet signature criteria but does not exceed size threshold still be selected it is important flows to account its usage. The number of such flows can have a certain limitation to the threshold of flow table entry. We call it smart flow sampling. It is smart in the sense that the decision of flow selection is not simply depends on its size.

3.4 Time Bucket Based Flow Timeout

Typical flow timeout rule (e.g., Netflow rule) is FIN arrival, inactive timeout, active timeout, or flow memory full. Main drawback of this rule is that flow doesn't faithfully reflect application behavior. Application session or service may span multiple

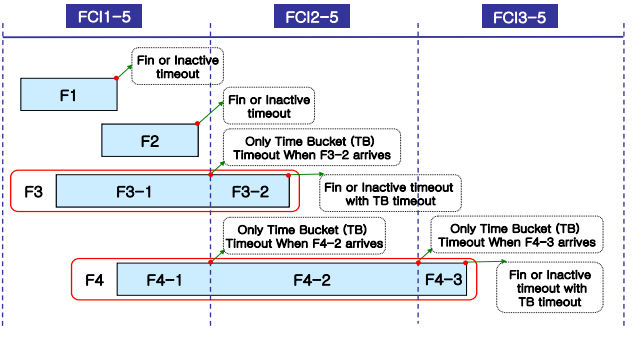

Fig. 2. Time bucket based Flow Timeout Mechanism

flows but there is no particular way of finding such relationship under current timeout mechanism. Other issues are the fact that analysis of flows may delay up to maximum active timeout period or inaccuracy of flow analysis which spans multiple analysis intervals can occur. These problems can't be solved by simply reducing the active timeout value because it causes unnecessary division of long-term flows. We have solved this problem with "time bucket based flow timeout" mechanism. Fig. 2 illustrates how it works. We define time bucket as flow creation interval (FCI). The default value is 5 minutes and can be flexibly adjusted. If a flow like F1 completes within a FCI by FIN arrival or inactive timeout, they are timed out normally. Also a flow like F2 completes within a FCI1-5 although FIN arrives or Inactive timeout occurs at FCI2-5 within inactive timeout period can normally time out within FCI1-5. However, a flow like F3 or F4 which spans multiple FCIs, they are forced to be timed out by a time bucket timeout. All the flows now contain an indicator that tells which FCI it belongs. This mechanism allows not only solving the above mentioned problems but providing accurate long-lived flow analysis and predictable performance of flow export and its associated memory usage.

3.5 Hardwired Flow Record Generation

We have defined a flow as extension of a normal flow. Our flow has a flow record and one or more associated packet records. Fig. 3 depicts the flow schematic. Flow record list is two hash tables kept in SRAM and actual values are stored in DRAM with packet records. The figure also shows that the how fragmented packets are handled for reassembly.

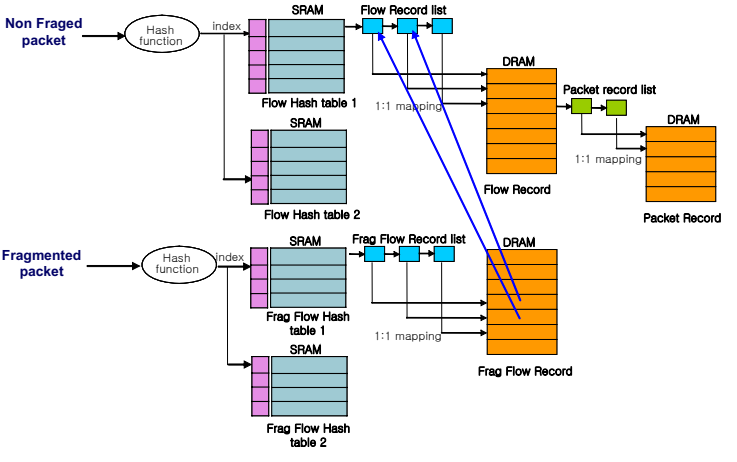

Fig. 3. Flow Data Structure

The extension of flow concept is defined for the purpose of providing more information to an analysis process for precise applications analysis. The packet record contains not only packet specific information but also has a part of payload which includes an application signature attached when it is found. Thus our flow contains

just enough information about accurate application classification without keeping the entire payload. The generation of flow records in this detail level at hardware is one of the most important features of our system. We have met system performance objective of supporting upto 4 million flows in a hash table per second and actual export rate of 600,000 flows per second for 10Gbps link.

3.6 Two-Step Flow Merge

When flows are generated normally, it considers a single direction only. However, applications are usually asymmetric. It is up to the analysis process which is responsible for identifying the bi-directional relationship of application flows. We have designed our system to conduct such flow merge in two steps. During 1st step, it checks whether two same direction flows belong to one original flow. This situation can occur when traffic is load-balanced at flow level. If such flows are found then they are merged and aggregated. As a 2nd step, it checks two opposite direction flows which has generated by the same application. If such flows are found, they are merged. Such merged flows then can be transferred to the application process for further analysis. This makes the application process job much simple and scalable. Fig. 4 illustrates how two step merge can take place.

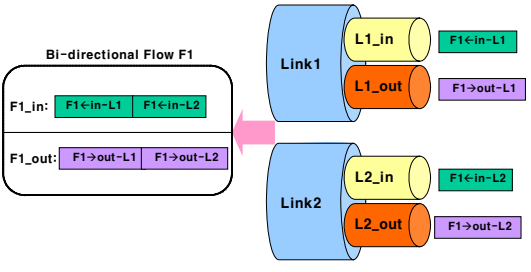

Fig. 4. Two-step Flow Merge

3.7 Application Classification Methodology

Other important novel features in our system are: general purpose Internet applications traffic classification algorithm, accurate application accounting, adaptability and expendability for newly emerging applications, and auto-detection mechanism of the new applications.

The objective of our application classification method is high-precision IP application identification. Since we are using several novel methods, the accuracy of application traffic accounting is much higher in comparison with most other currently available solutions. Since IP applications lifecycle is very dynamic, it is important to design the monitoring system to adapt such a characteristic. We designed a run-time configuration language, called Application Recognition Configuration Language (ARCL). When a new application appears, manual detection is very time-consuming and labor intensive and thus requires automation. We are currently working on such automation.

Our software methodology consists mainly of a precise application identification method, extensible applications recognition language, and flow definition extension.

Our approach is unique in that we classify the applications into the four distinctive types: Type-FP (Fixed Port-based Recognition), Type-PI (Payload Inspection-based Recognition), Type-DP (Dynamic Port-based Recognition), Type-RR (Reverse Reference-based Recognition). In the first method, recognition is performed on the basis of a predefined port number to application mapping. In the second method, recognition is performed on the basis of both port numbers and signatures. In the third method, recognition is performed on the basis of port numbers obtained by inspecting other flows' payloads. Lastly, recognition is performed on the basis of referential information obtained by recognizing a type-PI flow on the other links. We determined the types by inspecting, generalizing, formulating, and classifying more than 100 major IP applications. We gathered flow and entire packet specimens from four major networks including a large-scale campus network, an enterprise network, a major Internet exchange point, and an ISP.

The recognition language is simple and effective in that the system can be swiftly reconfigured to detect and recognize unprecedented or modified applications without the developer's writing and distributing extension modules for processing the newer applications, which usually results in fairly long period of shrunk recognition coverage even after detecting and analyzing the new applications. The basic hierarchy of ARCL semantics and syntax possesses three levels. The highest category is an application (application), the next is a representative port (port_rep_name), and the last is a subgroup (decision_group). There is a one-to-many relationship between a higher and a lower category. The basic concept can be explained by an example. Although most WWW services are provided through port 80, there are still many specific web services that are provided via other ports, such as 8080. All these representative ports pertain to the application "WWW" and have their own distinctive names; port_rep_names – "HTTP" for 80, "HTTP_ALT" for 8080, etc. Although a port_rep_name is tightly coupled with a port, the relationship between them is not one-to-one, nor fixed; a single port can be used by a number of different applications under different representative names. Packets in a flow can further be classified into a number of subgroups. For example, an HTTP flow subdivides into "HTTP_REQ" packets, "HTTP_REP" packets, ACK packets of HTTP_REP, etc. Each of these elementary subgroups constituting the entire context of a flow is a decision_group.

4 Implementation Experience and Results

4.1 Implementation and Deployment Experience

The proposed requirements and architecture is in the process of implementation as a proof-of-concept system, called Wise[*HITMAS]. It consists of a ATCA-compliant packet processing board, FM(Flow Mediator), AS(Application Server), DB server, and GUI. We are currently conducting various function and performance testing on our new system in a Lab environment with traffic generators, AX/4000 and IXIA IxLoad.

4.2 Performance Evaluation

As mentioned, our system performs real-time traffic statistics calculation. It generates top-N statistics per N number of source and destination hosts. It can handle up to 1.8Gbps 60 Byte packets without loss which matches with around 4 million pps. For 1500Byte packets, MA can fully support up to 10 Gbps even with real-time Top-N feature turned on. On the other hand, MA can support up to 2.4 Gbps with around 50 million pps without any packet loss in case that real-time Top-N statistics function turned off.

Measurement Data Accuracy Table by Monitoring Interval(1Day: 2006-0316 00:00 ~ 2006-03-16 24:00)						
Monitoring Interval	Measurement Data					
	Bps		pps		fps	
5Min	169,338,875,356		53,474,742		8,806,233	
1Min	169,854,435,559		53,587,637		8,826,019	
7.5Sec	171,737,532,549		54,873,333		9,106,379	
	5Min-X(%) Data					
5Min	-	-	-	-	-	-
1Min	-515560.202	-0.30%	-112.894	-0.21%	-19.785	-0.22%
7.5Sec	-2398657.192	-1.42%	-1398.590	-2.62%	-300.145	-3.41%
	1Day Total Sum					
Unit	Mbytes		Kpackets		Flows	
5Min	-		-		-	
1Min	-5,310,106		-9,525,431		-1,709,424,000	
7.5Sec	24,705,408		-118,006,031		-25,932,528,000	

Fig. 5. Measurement Data Accuracy by Monitoring Interval

Fig. 5 shows one interesting performance statistics which has been acquired from measurement in a real operational environment in one major ISP in Korea. Since our system can collect statistics data up to two seconds level, we conducted traffic measurement in different time intervals with the same raw data set. It shows that measurement analysis accuracy by intervals varies. The accuracy increases as the measurement time interval becomes more granular. This testing was conducted in a 1 Gbps International link. We can easily guess that the same testing with higher speed links will provide more differences. This result justifies that real-time measurement of very granular level will increase the accuracy of traffic usage statistics. Of course, this testing doesn't take other application classification algorithms into account. The accuracy will increase much more with them.

4.3 Deployment Experience in Real Field

We have deployed our system in one of major ISPs in Korea. The system is installed on the 10Gbps link that connects to another major ISP. The link is utilized in 6 – 7 Gbps in total bi-directionally. The average flow per second is less than 10,000 because the link does not carry much aggregated traffic. We have been deploying the system over 3 months now. We are testing various system functionality: packet capturing and flow generation in hardware level, flow filtering, two-step flow merging, DPI-based

application signature matching, analysis server's performance, etc. The detailed analysis work hasn't been completed similarly as shown in the section 4.2. We will update in our future version.

5 Conclusion and Future Work

Based on the novel mechanisms we introduced, we have successfully implemented our policy-based monitoring and control system supporting up to 10Gbps speed and are currently working for enhancing it to support much higher speeds upto 40Gbps. It is very challenging to support such capabilities upto 40Gbps especially at the hardware level. As far as we understand, such attempt hasn't been made before by other research. Our short-term goal is to verify our system's functionality in a real-time situation. For longer-term, we are looking for alternative hardware design which can improve the current performance limitations capable of supporting upto 40Gbps such as the lack of main memory for more flow processing, less flexible programming environment for application developers, and integration of more functionality like flow merge into hardware. Such work is for further study.

References

1. Donghua, Chuang, R., Zhen, L., Jia, C., Ungsunan, N., Peter, D.: Handling High Speed Traffic Measurement Using Network Processors. In: Proc. of International Conference on Communication Technology 2006, Beijing (November 2006)
2. Estan, C., Varghese, G.: New Directions in Traffic Measurement and Accounting: Focusing on the elephants, ignoring the mice. ACM Transactions on Computer Systems (TOCS), 270–313 (2003)
3. Bloom, B.H.: Space Time Trade-Offs in Hash Coding with Allowable Errors. Comm. ACM 13(7), 422–426 (1970)
4. van der Merwe, J., Caceres, R., Chu, Y.-h., Sreenan, C.: mmdump- A Tool for Monitoring Internet Multimedia Traffic. ACM Computer Communication Review 30(4) (October 2000)
5. Kang, H.-J., Kim, M.-S., Hong, J.W.-K.: A method on multimedia service traffic monitoring and analysis. In: Brunner, M., Keller, A. (eds.) DSOM 2003. LNCS, vol. 2867, pp. 93–105. Springer, Heidelberg (2003)
6. Sen, S., Wang, J.: Analyzing peer-to-peer traffic across large networks. In: Proceedings of the second ACM SIGCOMM Workshop on Internet Measurement Workshop (November 2002)
7. Operax Resource Controller series, `http://www.operax.com/products/default.asp`
8. ITU-T Recommendation Y.2111, Resource and admission control functions in Next Generation Networks (2006)