

Semantic router using data stream to enrich services

Koichi Inoue, Dai Akashi, Michihiro Koibuchi, Hideyuki Kawashima
and Hiroaki Nishi, *Member, IEEE*

Abstract—In this paper, we propose a new router architecture that enables router to interact with services. The proposed router observes the traffic data stream, inspects the packet payload as well as packet headers, and stores the designated data in the associated database. We also propose a new query language SSRQL for operating massive traffic data stream in the proposed router. Application programmers can access to the database with SSRQL queries to develop new services in the data-oriented Web2.0 world.

Index Terms—Information services, Network routing, Database

I. INTRODUCTION

DATA becomes contents and enriches web services. In the Web2.0 world, data includes text, image, video, and information of geographic location and commercial products, and is generated by both service providers and users. Mashup enabled by API (Application Programming Interface) combines data from enormous sources into web services, taking advantage of network effects, which enhances data interaction as well as further data generation.

Routers are the computing systems having a great processing power, and currently take a responsibility for packet transmission and converting protocol. Although routers have been central to the IP-based communication network, for a long time routers have not been involved aggressively in the application layer where users enjoy services. However, data acquired by routers is unique. Data provided by routers where tremendous amount of packets are transmitted can be considered to make web services richer with data based on actual traffic and with wider and deeper data coverage, while data provided by service provides is currently collected by just an end host.

Manuscript received April 18, 2008.

K. Inoue is with School of Integrated Design Engineering, Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohokuku, Yokohama 223-8522 Japan (E-mail: kinoue@west.sd.keio.ac.jp)

D. Akashi and H. Nishi are with School of Integrated Design Engineering, Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohokuku, Yokohama 223-8522 Japan.

(E-mail: akashi@west.sd.keio.ac.jp, west@sd.keio.ac.jp)

M. Koibuchi is with the Information Systems Architecture Research Division, National Institute of Informatics, National Center of Sciences, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan.

(E-mail: koibuchi@nii.ac.jp)

H. Kawashima is with the Graduate School of Systems and Information Engineering, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573 Japan. (E-mail: kawasima@cs.tsukuba.ac.jp)

Our goal is to design service-friendly router that provides unique data based on actual traffic data stream to enrich services. And we describe the concept of the proposed router and its architecture. The proposed router inspects specified packets and stores them in the associated database efficiently. We also propose a new query language SSRQL (Semantic Switch Router Query Language) that operates the traffic data stream and manages the database.

The remainder of this paper is structured as follows. Section 2 describes background and related work. The concept of the semantic router and key technological challenges on it are presented in Section 3 and 4 respectively. In Section 5, expected application is described. Finally this paper is concluded and outline directions for future research are described in Section 6.

II. RELATED WORK

In this section, we briefly survey underlying technologies that can form a symbiotic relationship to achieve the proposed service-friendly router.

Search Engine is the key technology to help to minimize the time for finding information and the amount of information to be consulted. Data on the web pages is retrieved by web crawler and is stored in an index database, and eventually search engine builds and maintains a huge database. Google [1] is taking advantage of the data to provide APIs, and obviously it enriches web services. Although search engine and its algorithm are proposed by academic world, commercial service providers such as Google and Yahoo! [2] have been technologically advanced and widely used. However, in the case of existing major commercial search engines, “transparency” which means that openness in how the systems and algorithms operate and “quality” which means that relevancy and accuracy of the search results require more improvement. Search Wikia [3] is an open source search engine, and thereby the transparency is less problematic and the quality continues to be improved by adapting to the specific purposes with its open architecture. But these search engines are still based on the web crawler managed by end hosts. On the other hand, personalized search such as iGoogle has a functionality to collect the history of the search results with time stamp. It is an additional way of data collection besides web crawler, which can be considered to improve the quality of search result. Few search engines, however, have implemented actual traffic data for the improvement of the search quality.

Advanced Internet backbones, such as a nationwide

academic information network called SINET3 in Japan [4], increasingly provide a rich variety of services, because cutting edge applications, such as Grid, sometimes require high-quality secure connections using VPN (virtual private networks), QoS (quality of services), and high-priority multicasting. Fortunately, as technology scaling is improved, a large number of advanced routers have been equipped with the above various services that will enable to make highly advanced router-centric systems. Thus, it can be said that various advanced technologies can be included in a single router, and the router continues to employ additional services for heavy users and Internet service providers.

Application-friendly routers are commercially available. Cisco has AXP (Application eXtension Platform) [5] for Cisco ISR (Integrated Services Router), and recently released the API. AXP is an extension module of Cisco ISR including for the enterprise use. With the API and its SDK, system integrators and user companies can develop application tightly integrated with routers. The target applications are security functions and integration with mission critical system, while our approach places database at the center of the target application to enrich services for consumers.

Routers typically inspect packet headers to collect information of layer 1 through 4 to route packets. However, interaction between routers and packet payload has been discussed in the architecture of content-based routing [6-7]. Moscola, Cho and Lockwood presented content-based router [8] by reconfigurable hardware architecture to route packets based on the content that appear in the packet payload rather than the IP address of the destination, and its hardware design was proved to have the certain processing power required for deep content inspection to maintain multi-gigabit networks. This work is the same research domain in the sense of deep packet inspection including packet payloads. However, our proposal focuses on the valuable data that routers acquire and the methodology of storing, querying, and providing it in an open and service-friendly manner. From a long-term perspective, the proposed routers that interact with services via APIs have a potential capability to determine the routing table of the content-based router which currently requires pre-defined routing table manually.

To cope with frequently arriving data, stream processing [9] has been developed. Though the first and second generation stream processing engines (SPE) [10] ignore applications, the latest and the third SPE are motivated by specific applications such as real-world event stream processing [11]. To the best of our knowledge traffic streams are not discussed, and it should be noted that our proposition is the first work that researches the real traffic streams.

III. SEMANTIC ROUTER

People publish, share, search, and receive information on the Internet and the huge database on the server is taking an important roles as search engines or web services to help people's these transactions. Although routers have been located in between computer networks and currently deal with these transactions by inspecting most packet headers and transmitting

them, routers have not been involved in these processes in the application layer. Considering the facts that majority of the search engines and web services are based on the client-server model and that data acquired by routers is unique, it is possible to regard that routers have an interface in the application layer to help people's publishing, sharing, searching and receiving information.

Our long-term goal is to design service-friendly router, semantic router, which provides unique data based on actual traffic data stream to enrich services. Router is no longer just a routing hardware that transmits data and converts protocols, but becomes semantic router that inspects traffic data stream including packet payloads and provides functionalities in the application layer to servers, clients, and neighboring routers. Here, high-level discussion of the design concepts is described.

First, while conventional routers are dealing with IP protocols, semantic router has a capability to deal with contents in the packet payload. With richer information to determine routing table, semantic router improves the efficiency and controllability of Internet traffic. Semantic router is involved in applications of P2P and overlay network by providing topology related information in order for applications to help fast and efficient routing.

Secondly, for the same reason, seamless integration of IP-based routing and content-based routing enables application programmers to create new services.

Third, semantic routers organize a large-scaled distributed database among neighbor routers. While existing search engine is based on the web contents crawled by end hosts, information discovery taking advantage of the proposed database may help improving the quality of the search result.

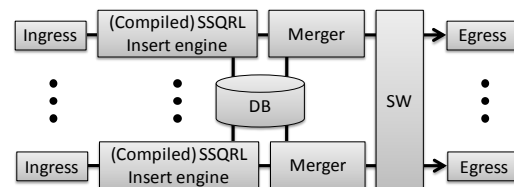


Fig. 1 Semantic router architecture

The following observation forms the basic of high-level design of semantic router and key requirements.

A. Massive data processing

We are designing semantic router having a capability of deep packet inspection including packet headers and payload. Eventually the database associated with semantic router needs massive data processing capability.

B. Privacy issue

Since the database of semantic router is accessible by application programmers and service providers, privacy protection scheme is indispensable. Only authentic user can access to semantic router via API authentication. Other users need to use guest account and most of all contents are anonymous under the guest account.

IV. REALIZATION OF SEMANTIC ROUTER

The first step to realize the semantic router is to define the solutions to ensure the requirements listed in Section 3.

A. SSRQL defined for operating traffic data stream for massive data processing

SSRQL (Semantic Switch Router Query Language) is a control and administration language of both semantic router and its database to operate traffic data stream. SSRQL is an extension of SQL and XML, which defines the semantic router specific functionalities besides standard SQL statements.

Default SSRQL is the statement issued by router administrator and it keeps storing traffic data for a long time. It is also used for the filter of privacy protection, hardware resource management, and inter-semantic router data sharing.

User SSRQL is a statement for application programmers to collect specific data that may enrich their services or research activities. Also it is used for issuing queries of database.

Active SSRQL is based on the idea of Active Messages [12]. SSRQL request can make another request to a remote semantic router and also SSRQL can be a handler of targeted SSRQL message. This inter-semantic router message exchange enables to build a distributed semantic router database and also controls total network performance and function. This SSRQL also provides a database sharing methodology.

Compiled SSRQL is a statement executed by reconfigurable hardware engine. The statement is performed fast and in parallel. It is mainly used for wire-rate packet capturing in the front-end of semantic router. Compiled SSRQL is a subset of SSRQL and it is transferred to a finite state transition machine. Semantic router basically does not assure wire-rate packet capturing and storing. However, it directly affects to the quality of database and services. Semantic router provides certain mechanisms for assuring quality of capturing, and this Compiled SSRQL is one of these mechanisms. Meanwhile, normal SSRQL provides higher level operation; data mining, semantic search, and hardware resource aware operation. However, these processes take time relatively.

There is a possibility where multiple SSRQL statements are issued to a same data stream. To eliminate the duplicated acquisition process, MRO (Multiple Request Optimization) is required. As shown in Fig. 2, MRO integrates the same multiple query messages into one message.

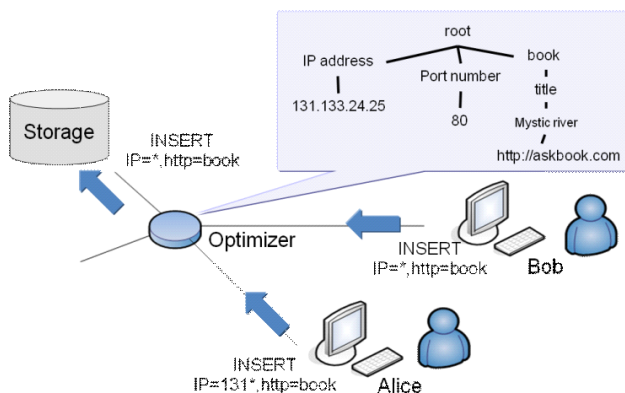


Fig. 2 Multiple Request Optimization

To have massive data processing capability in the data acquisition process, SSRQL allows restrictions somehow such as storage space limitation or expiration date of statement. Besides, except Compiled SSRQL, SSRQL works as software, and accordingly integrity of the acquired data is not guaranteed (best-effort). Data stored in the database also has an expiration date due to the limited storage space. Storage space has the alternative of on-board memory or HDD, and there is a tradeoff between access speed and capacity.

To remove potential issue of massive database access when to interact with services due to the limited processing power, semantic router creates virtual table for User SSRQL. Virtual table not only store data in the storage space that router own, but store into other routers and the databases which are managed by the application programmers issued User SSRQL.

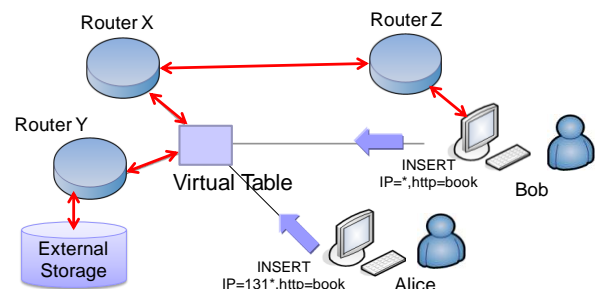


Fig. 3 Virtual Table

B. Privacy Protection

Since semantic router cannot analyze the contents of encrypted packet such as HTTPS, it ignores encrypted data transfer and never stores the data to its database. Though encrypted access is a simple way to protect privacy, some page uses normal HTTP to exchange privacy information. In this case, AHP (authority hand-over protection) are required to keep the privacy.

AHP-pf (AHP by post fetch) uses post-fetch access to protect the leakage of privacy information. If the web page requires an authentication, authentic user can obtain a special web page that contains privacy information. In this case, semantic router accesses the URL again and if the face of the web page is different from the stored information which was captured by the access of authentic user, it can recognize that the page contains privacy information and discard the page.

AHP-sa (AHP by semantic analysis) uses semantic analysis to protect the leakage of privacy information. The access of GET or POST method may contains the information of password or login account and this information is undesirable to be apparent. Additionally, input method with password type and cookie transaction may contain privacy information too. These privacy information are anonymized.

AHP-pf and AHP-sa can be described by using default SSRQL and are achieved by a default-SSRQL executor. If AHP-sa enabled, all INPUT method is filtered by it. Default SSRQL can be used as a primary filter of data acquisition. All information that is not permitted from an access of unauthenticated user are masked and become anonymous.

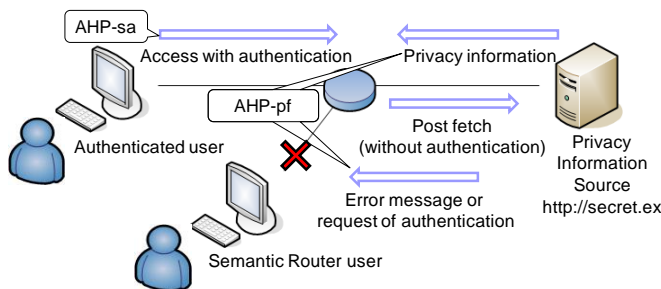


Fig. 4 Authority Handover Protection

LAC (Layer based Access Control) makes the information anonymous according to OSI layer model. The information of lower layer contains IP address or protocols and upper layer contains application specific data. Information of each layer has different style of utilization and is hopeful to be protected by specialized privacy protection or authentication. LAC requires user to have special authentication dedicated to each OSI layer.

V. APPLICATION

In this section, we briefly describe a possible application that will be created by our proposed architecture.

RSS reader based on the real popularity stakes

RSS reader is the tool to know the updated contents of the user's favorite web site. RSS is based on XML, and it is formed in a structured way with the industry standard tag.

RSS tags can be found on the web site or RSS portal. The popularity can be found in the portal, but it isn't easy to know which RSS feeds are actually the most popular.

Semantic router can identify the most popular tag based on the actual traffic. The duplicated RSS packet to the same destination IP address can be removed according to the database information.

Below is the sample User SSRQL to create table and collect data.

1. CREATE TABLE *table_name* (*title* CHAR(256), *link* CHAR(256), *description* CHAR(256), *times* INT)
2. INSERT INTO *table_name* (*title*, *link*, *description*, *times*) WHERE payload = `xmlns="http://purl.org/rss/1.0/"`
3. UPDATE *table_name* SET *times* = *times* + 1 WHERE *link* = `aaa`

When the data is collected, the real popularity stakes can be calculated as follows,

- 4 SELECT *title*, *link*, *description* FROM *table_name* WHERE *times* > 10

VI. CONCLUSION

Many types of data enrich web services. In these circumstances, routers that are central to the IP-based network infrastructure may contribute to enrich services with providing its data. In this paper, we presented an application/service friendly router. As the first technological challenge, we described the router architecture with integrated database, and proposed a new query language dedicated to operate massive traffic data stream. As the next work, we are planning to implement several applications on a simulation with using raw Internet traffic data.

The authors have worked on the research domain of router

architecture [13-14]. Our long-term goal is to design a truly service-friendly router. Otherwise, router will remain apart from application layer and will remain mere expense for telecom carriers. The vision includes content-based routing functionality, but our approach is starting with implementing the interaction between services and routers as presented in this paper, and then moving towards dynamic routing table optimization for content-based router through the interaction with services. There is a related work for the advanced SQL which enables users to develop services where information from the web and devices in the real world work collaboratively by writing SQL codes [15]. In the same sense, we plan to extend SSRQL to manage functionalities and configuration of routers to enable content-based routing.

ACKNOWLEDGMENT

This work was partially supported by National Institute of Information and Communications Technology in Japan and Joint Research Fund of National Institute of Informatics (NII), Japan.

REFERENCES

- [1] Google <http://www.google.com>
- [2] Yahoo! <http://www.yahoo.com>
- [3] Search Wikia <http://search.wikia.com>
- [4] Shigeo Urushidani, Shunji Abe, Kensuke Fukuda, Jun Matsukata, Yusheng Ji, Michihiro Koibuchi, Shigeki Yamada, Architectural Design of Next-generation Science Information Network, IEICE Transactions on Communications, VOL.E90-B No.5, pp.1061-1070, May 2007.
- [5] Cisco Application eXtension Platform <http://www.cisco.com/en/US/products/ps9701/index.html>
- [6] Chu-Sing Yang and Mon-Yen Luo, "Efficient Support for Content Based Routing in Web Server Clusters," in Proceedings of USENIX Symposium on Internet Technologies & Systems (USITS), Boulder, CO, Oct. 1999.
- [7] A. Carzaniga, M. J. Rutherford, and A. L. Wold, "A routing scheme for content-based networking," in Proceedings of IEEE INFOCOM 2004, Hong Kong, China, Mar. 2004.
- [8] J. Moscola, Y. H. Cho, and J. Lockwood, "A Reconfigurable Architecture for Multi-Gigabit Speed Content-Based Routing," in Proceedings of the 14th IEEE Symposium on High-Performance Interconnects (HotI'06), 2006.
- [9] A. Arasu, S. Babu and J. Widom, "CQL: A language for continuous queries over streams and relations", Proc. of International Workshop on Database Programming Language, pp 1-19, 2003.
- [10] D. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing and Stan Zdonik, "The Design of the Borealis Stream Processing Engine", Proc. of International Conference on Innovative Data Systems Research, 2005.
- [11] E. Wu, Y. Diao, and S. Rizvi, "High-Performance Complex Event Processing Over Streams", Proc. of ACM SIGMOD International Conference on Management of Data, pp. 407-418, 2006.
- [12] T. von Eicken, D. E. Culler, S. C. Goldstein, and K. E. Schauer. Active Messages: a Mechanism for Integrated Communication and Computation In Proc. of the 19th Int'l Symposium on Computer Architecture, May. 1992.
- [13] M. Okuno and H. Nishi, "Network Processor Accelerator Using Temporal Locality of Traffic" (in Japanese), IPSJ Transactions on Advanced Computing System, 45(SIG6):pp.45-53, May. 2004.
- [14] H. Toyoda et al, "High-speed Signal Transmission and Coding Architecture for Next-Generation Ethernet," IEICE Transactions on Informaton and Systems, E86-D(11):pp.2317-2324, Nov. 2003.
- [15] Yuhei Akahoshi, Hiroaki Ohshima, Yutaka Kidawara and Katsumi Tanaka, "PeRDB: A Collaborative Service Development Platform for the Web and the Devices in Real World" (in Japanese), Proc. Of Data Engineering Workshop, DEWS2008 D7-6, 2008.