Development of Aggregation Device for Next Generation Backbone Networks

Daisuke Matsubara† Satoshi Kiyotou†† Tomohiro Baba†† Yasushi Fukuda††
Atsushi Kobayashi‡ Tsuyoshi Kondoh‡

† Central Research Laboratory, Hitachi, Ltd. 1-280, Higashi-koigakubo Kokubunji-shi, Tokyo 185-8601, Japan
†† Network Systems Solutions Div., Hitachi, Ltd. 890, Kashimada, Saiwai-ku, Kawasaki, Kanagawa 212-0058, Japan
‡ NTT Information Sharing Platform Laboratories 3-9-11, Midori-cho, Musashino-shi, Tokyo 180-8585, Japan
E-mail: † daisuke.matsubara.pj@hitachi.com, †† yasushi.fukuda.me@hitachi.com, ‡ kobayashi.atsushi@lab.ntt.co.jp

Abstract This paper explains architecture of aggregation device that stores and aggregates traffic information of next generation backbone networks. As the traffic volume of the backbone network grows and the number of flows grows, the monitoring system needs to collect higher volume of traffic information to observe network conditions. The traffic information sent from 50 routers may equal up to 1M flow records per minute, which is difficult to process by a centralized monitoring server. We have designed and implemented prototype of the aggregation device that stores and aggregates traffic information to reduce volume of data sent to the monitoring server.

Keyword traffic monitoring, aggregation, IPFIX, NetFlow

1. Introduction

The recent increase in the Internet traffic and use of critical network applications such as VoIP has derived need for network platform that enables high scalability and reliability in data transfer. Based on these requirements, we have studied traffic monitoring system that enables constant monitoring of high volume traffic and detection of anomaly in next generation backbone networks.

In this paper, we will propose aggregation device that stores and aggregates traffic information and explain architecture and application for this device. The proposed device will help to enable operators to monitor network traffic and detect anomaly in large scale backbone network.

2. Architecture of Traffic Monitoring System

The proposed traffic monitoring system consists of routers, aggregation device, and monitoring server. Figure 1 shows overall view of the traffic monitoring system and Table 1 explains the each components.

Traffic is first measured at the router which has interface of up to 10Gbps, and the router then sends traffic information to the aggregation device. The aggregation device aggregates the traffic information and sends the aggregated data to the monitoring server. The traffic information that is sent from the router to the aggregation device may be NetFlow version 9[1] or IPFIX[2]. The aggregated data sent from the aggregation device to the monitoring server may be NetFlow or IPFIX, or extended version of these standard protocols.

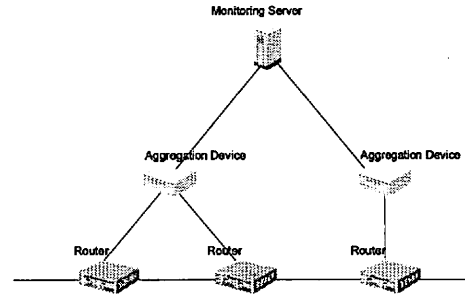

Figure 1: Overview of the monitoring system

Table 1: Components of the monitoring system

Device	Function
Router	Measure traffic and send traffic information to aggregation device.
Aggregation Device	Store and aggregates traffic information. Sends aggregated data to monitoring server.
Monitoring Server	Calculates traffic statistics based on aggregated data sent from aggregation device.

3. Overview of Aggregation Device

3.1. Role of Aggregation Device

To enable monitoring of large scale network and analysis of anomaly inside the backbone network, aggregation device performs these functions:

1. Monitoring: Monitoring server receives traffic information of the entire network to perform constant monitoring. The aggregation device aggregates the

traffic information sent from router and reduces load of the monitoring server.

2. Anomaly analysis: When anomaly is detected, monitoring server retrieves detailed traffic information that is stored inside the aggregation device to perform more detailed analysis of the network traffic.

3.2. Architecture of Aggregation Device

The basic architecture of the aggregation device is shown in Figure 2. The aggregation device has three key functionalities explained below.

1. Data store module: stores traffic information received from routers to the local HDD.
2. Aggregation module: aggregates traffic information received from the routers and sends the aggregated data to the monitoring server.
3. Query module: receives query request from the monitoring server, retrieves relevant data from the stored traffic information, and returns the data to the monitoring server.

The data received from the router is sent to both aggregation module and data store module. The aggregation module aggregates traffic information and sends the aggregated data to the monitoring server for monitoring. The data store module stores the traffic information onto the HDD device.

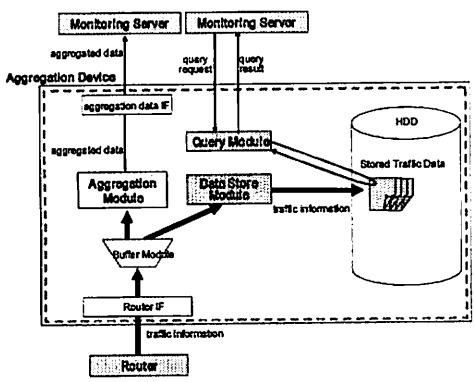

Figure 2: Architecture of aggregation device

3.3. Data Store Module

The data store module is enables store traffic information with throughput of 1M records per minute, which equals to information volume of 50 routers. There are several alternatives for storing traffic information, such as RDBMS (Relational Database Management System), in-memory DB, stream DB, and proprietary data format. Comparison of these methods is shown in Table 2.

The data store rate of a typical RDBMS only scales up to 1K records per minute, which is insufficient for the proposed usage. In-memory DB can achieve sufficient performance, but cannot store large volume of data because it uses memory for storing data. Stream DB and proprietary data format has ability to record high rate of data, which can be used for storing traffic information. Stream DB provides function to perform data search using SQL-like command, so it is fit for complex data search. Proprietary data format requires additional query function to be implemented for complex search, but the data format can be tailored to minimize data size and optimize storing performance. The comparison of these storing methods is shown in Table 2. We have decided to use proprietary data format for the initial implementation of the aggregation device, for it does not require complex query function. We will also evaluate usage of stream DB in the future if more complex search is needed.

Table 2: Methods for storing traffic information

Storing Method	performance	storage capacity	query function
RDBMS	<500 records/sec	250GB (HDD)	SQL
In-memory DB	Up to 700K records/sec	2GB (memory)	SQL
Stream DB	Up to 200K records/sec	250GB (HDD)	SQL
Proprietary	Up to 200K records/sec	250GB (HDD)	Needs to be implemented

To minimize data size and optimize performance, we chose NetFlow/IPFIX data format for storing data in proprietary data format. By using NetFlow/IPFIX format, traffic information received as NetFlow/IPFIX packets can be stored with a simple conversion process and the data size will be minimal. The data store module bundles multiple Data FlowSets received from multiple NetFlow packets as concatenated data. (Figure 3) The data store module also reassembles data fields inside each record to align with store template. Store template is a NetFlow template that is configured by the operator to specify the format for each record that is stored in the aggregation device. An example of store template is shown in Table 3.

By using data format used in standard protocol, functionalities for processing the data can be easily implemented in external systems such as the monitoring server. Also by providing store template, the operator can configure data format based on their preference. This module can be modified to fit other formats based on NetFlow/IPFIX that may be standardized in the future [3].

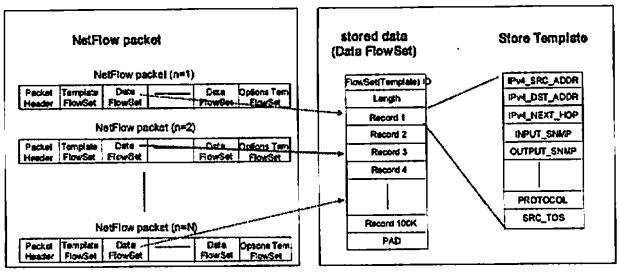

Figure 3: Method for data store

Table 3: Example of a store template

Field Type	Field Name	Size (B)
1	IN_BYTES	8
2	IN_PKTS	8
4	PROTOCOL	1
5	SRC_TOS	1
6	TCP_FLAGS	1
7	L4_SRC_PORT	2
8	IPV4_SRC_ADDR	4
9	SRC_MASK	1
10	INPUT_SNMP	2
11	L4_DST_PORT	2
12	IPV4_DST_ADDR	4
13	DST_MASK	1
14	OUTPUT_SNMP	2
15	IPV4_NEXT_HOP	4
16	SRC_AS	2
17	DST_AS	2
18	BGP_IPV4_NEXT_HOP	4
21	LAST_SWITCHED	4
22	FIRST_SWITCHED	4
34	SAMPLING_INTERVAL	4
35	SAMPLING_ALGORITHM	1
46	MPLS_TOP_LABEL_TYPE	1
47	MPLS_TOP_LABEL_IP_ADDR	4
60	IP_PROTOCOL_VERSION	1
61	DIRECTION	1
70	MPLS_LABEL_1	3
71	MPLS_LABEL_2	3
72	MPLS_LABEL_3	3
0x8002	EXP_IPV4_ADDR	4

3.4. Aggregation Module

To aggregate multiple flow records into single flow record, operator specifies one or more data fields in the NetFlow/IPFIX records as aggregation key. The operator also specifies which fields needs to be calculated upon aggregation. These specifications are configured in aggregation template. An example of aggregation template for 'any-port' aggregation (aggregate source and destination port numbers) is shown in Table 4. The method for aggregating flows is also explained in [4].

The aggregation module matches the value of the keys and aggregates flows that have the same set of values. When aggregating the flows, the aggregation module calculates the value of the fields specified in aggregation template.

Table 4: Example of an aggregation template

Field Type	Field Name	Size (B)
1	IN_BYTES	aggr(sum)
2	IN_PKTS	aggr(sum)
3	FLOW	aggr(sum)
4	PROTOCOL	key
6	TCP_FLAGS	aggr(OR)
7	L4_SRC_PORT	key
8	IPV4_SRC_ADDR	key
9	SRC_MASK	keep
10	INPUT_SNMP	keep
11	L4_DST_PORT	key
12	IPV4_DST_ADDR	key
13	DST_MASK	keep
14	OUTPUT_SNMP	keep
15	IPV4_NEXT_HOP	keep
16	SRC_AS	keep
17	DST_AS	keep
18	BGP_IPV4_NEXT_HOP	keep
21	LAST_SWITCHED	aggr(max)
22	FIRST_SWITCHED	aggr(min)
34	SAMPLING_INTERVAL	keep
35	SAMPLING_ALGORITHM	keep
46	MPLS_TOP_LABEL_TYPE	discard
47	MPLS_TOP_LABEL_IP_ADDR	discard
60	IP_PROTOCOL_VERSION	keep
61	DIRECTION	keep
70	MPLS_LABEL_1	discard
71	MPLS_LABEL_2	discard
72	MPLS_LABEL_3	discard
0x8001	AGGR_TYPE	append
0x8002	EXP_IPV4_ADDR	append
0x8003	AVE_ACTIVE_TIME	append
0x8004	MIN_ACTIVE_TIME	append
0x8005	MAX_ACTIVE_TIME	append

3.5. Query Module

The query module receives query message (XML) from the monitoring server as query request. The parameters used in the query message are based on MIB specifications for IPFIX concentrator. [5] The response of the query is sent from the aggregation device as NetFlow

packets.

4. Evaluation

We developed a prototype of the aggregation device and conducted test for evaluating system performance and aggregation ratio.

The maximum performance for NetFlow export of current router is approximately 10 to 20K records per minute. An aggregation device may aggregate 10 to 50 routers. From these conditions, aggregation device may need throughput of 100K to 1M records per second.

To evaluate system performance, we sent Netflow packets from an emulator tool to the aggregation device and measured the data rate for storing the traffic information. As seen in Figure 4, we did three repetition tests (graph 1,2,3 in Figure 4) and confirmed in all tests that the aggregation device has storing ability of approximately 1M records/sec.

Also, simulation using MAWI traffic sample [6] indicates that aggregation ratio (ratio of aggregated data to original traffic information) of any-port aggregation of may be from 40 to 70 %. We conducted a test by aggregating MAWI traffic at 24K records per minute. The aggregation cycle was set to 20 seconds. As seen in Figure 5, we confirmed that the aggregation ratio is 69.7%. This is not sufficient reduction of the traffic information, so improvement for lowering aggregation ratio is needed in the future. The aggregation ratio can be lowered by setting the aggregation frequency to longer time span (1 to 5 minutes) or by using other aggregation templates such as destination host aggregation (uses destination IP address as aggregation key), etc.

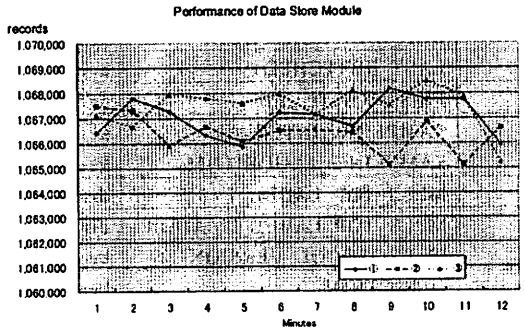

Figure 4: Results of data store throughput

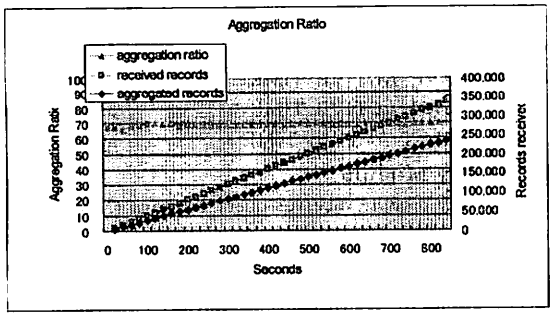

Figure 5: Results of aggregation ratio

5. Conclusion

This paper explains architecture and implementation of aggregation device that stores and aggregates traffic information of next generation backbone networks. We tested a prototype system to evaluate the performance using traffic sample of a backbone network. We have confirmed that it has throughput 1M flow records per minute, and aggregation ratio of 69.7% when using any-port aggregation at aggregation cycle of 20 seconds.

In our future study, we will improve and evaluate performance for next generation routers which may have higher NetFlow/IPFIX throughput. We will evaluate aggregation ratio using other aggregation templates, and also evaluate query function. Furthermore, we will study other functions such as filtering and load balancing using the aggregation device.

This work was supported by Ministry of Internal Affairs and Communications of the Japanese Government.

References

[1] B. Claise, Ed et al, Cisco Systems NetFlow Services Export Version 9, Internet Engineering Task Force (IETF), October 2004

[2] B. Claise, Ed et al, IPFIX Protocol Specification, <draft-ietf-ipfix-protocol-22.txt> Internet Engineering Task Force (IETF), June 2006

[3] B. Trammell, E. Boschi, L. Mark, T. Zseby, An IPFIX-Based File Format, <draft-trammell-ipfix-file-01.txt> Internet Engineering Task Force (IETF), June 2006

[4] A. Kobayashi, The reference model of IPFIX concentrators, <draft-kobayashi-ipfix-concentrator-model-01.txt> Internet Engineering Task Force (IETF), March 2006

[5] A. Kobayashi, Managed Objects for IPFIX concentrator, <draft-kobayashi-ipfix-concentrator-mib-01.txt> Internet Engineering Task Force (IETF), March 2006

[6] MAWI (Measurement and Analysis on the WIDE Internet), http://www.wide.ad.jp/project/wg/mawi.html