A Network Monitoring Tool for CCN

Wonchul Kang	Byungryeol Sim	Joohee Kim	Eunkyoung Paik	Youngseok Lee
CNU	CNU	KT	KT	CNU
Daejeon, Korea	Daejeon, Korea	Seoul, Korea	Seoul, Korea	Daejeon, Korea
teshi85@cnu.ac.kr	simsiss@cnu.ac.kr	jooheekim@kt.com	euna@kt.com	lee@cnu.ac.kr

Abstract—**Recent discussions on the revolutionary Future Internet architecture have led to Content-Centric Networking (CCN) that puts emphasis on contents for the efficient information delivery. Since CCNx has been released as an early stage prototype of CCN, many projects are being developed on CCNx. However, a network monitoring function for CCN nodes or faces has not been implemented. In this paper, we propose a simple network monitoring method based on IETF IPFIX and SNMP for CCN. From IPFIX, we could easily find the detailed information on contents delivered in interest or data packets in the format of flows, while retrieving information on the CCN node and its tables in SNMP. Through our monitoring scheme, we could clearly diagnose the current status of CCN nodes and their data such as Content Stores, Pending Interest Table (PIT), and Forwarding Information Base (FIB).**

Index Terms-- CCN, CCNx, montoring, IPFIX, SNMP

I. Introduction

The explosive increase of Internet traffic has been observed during recent years. According to Cisco [1], global IP traffic has multiplied eightfold over the past 5 years and annual global IP traffic will reach the zettabytes by the end of 2015. In [2], it is also reported that digital universe in 2011 will surpass 1.8 zettabytes, growing by a factor of 9 in just five years. It is expected that driving factors such as video traffic, mobile computing, and cheap storage will enlarge the information volume and the exchanged traffic in networks. The new trend of traffic on the Internet architecture has been studied in [3], where Google, Comcast, and Content Delivery Network (CDN) companies are ranked among top 10 of Internet backbone cores. This shows that the information exchange in the Internet has evolved to the content-centric manner from a host-centric communication model.

Recent discussions on building the Future Internet architecture against the data traffic explosion have triggered the clean-state design of the Internet. One of the proposals is CCN [4]. The motivation of CCN is to remove the redundant contents requested by multiple clients at the same time, retrieving the cached content at nearby routers. At popular web servers such as CNN or New York Times, the same contents will be delivered to each subscriber. Similarly, when hot events such as Olympics are broadcasted in the Internet, a lot of people will access to the websites at the same time, downloading the same videos. This same content distribution will be observed at the routers along the path from a server to clients. CCN has focused on the content-centric communication model and proposed an architecture that decouples location from identity and retrieves contents by name instead of addresses. In CCN, requests to contents could be resolved by intermediate routers along the path to the origin content servers. Therefore, in the aspect of network management, it is not easy to monitor the traffic flows and their performance issues, because the contents could be delivered from an intermediate node. For example, recent studies [5]–[7] on caching policies at CCN nodes should be supported by network monitoring functions in the real network environment. For this purpose, in this work, we propose an integrated monitoring method for CCN. First, we adopt IETF IP Flow Information eXport (IPFIX) [8] to capture and analyze flow information of content distribution.

Second, we use a well-known SNMP to probe the states of CCN nodes. We define CCN-related IPFIX flow templates and SNMP MIBs. By using both flows and MIBs, we could show the big picture of CCN network status. The remainder of the paper is organized as follows. Section II briefly introduces the CCN protocol, its implementation, IETF IPFIX, and SNMP. Our monitoring method will be explained in Section III and its prototype results will be presented in Section IV. Finally, Section V concludes this paper.

II. Background

A. CCNx

The CCNx [9] is an open-source project to develop CCN and its applications. It supports IP for testing and developing new features or applications on CCN. In CCNx, two important types of packets are Interest and Data. An Interest packet will contain content name requested by clients or CCNx nodes, and the corresponding content will be delivered in Data packets sent by origin servers or intermediate CCNx nodes. When a user requests content, this packet will be created and sent to the next-hop CCN nodes. If the node does not have the content, then it will send an Interest packet to the next-hop nodes. Otherwise, when a node has the requested content, it will reply with Data packets. Each CCN node manages its Content Store, Pending Interest Table (PIT), and FIB for content delivery. A Content Store is the buffer memory for the content, and The PIT is a table for tracking Interest packets that have not been satisfied and sent to upwards origin content servers. The FIB is used to forward Interest packets toward source of matching Data. The CCNx code provides an overlay prototype of CCN over IP. The *ccnd* encodes packets in the compact binary XML representation using dictionary-based tag compression. The packet forwarding function is implemented in *ccnd* as a user-space daemon. Interest and Data packets are encapsulated in UDP.

B. *IPFIX and SNMP*

Based on Cisco NetFlow, IETF has standardized IPFIX that could export flow information from routers, probes, and other network devices. Whereas the NetFlow protocol provides only IPv4 flow, IPFIX considers a flexible flow format that could deliver IPv6, MPLS, or user-defined flow attributes. A flow will consist of any number of packets observed in a specific timeslot and sharing a number of properties likes five-tuple of source IP, destination IP, source port, destination port and protocol.

SNMP is a widely used network management protocol for monitoring and controlling devices on IP networks. Usually, SNMP agents will collect information on devices and send the appropriate data to the collector. Each information element is defined in MIB II.

III. A NETWORK MONITORING METHOD FOR CCN

In this section, we introduce a network monitoring architecture for monitoring CCN nodes and flows.

A. *Architecture*

Our CCN monitoring tool consists of IPFIX/SNMP agents and a server as shown in Fig 1. Each agent for IPFIX or SNMP will work on every CCN node. An IPFIX agent will detect a flow from packet arrivals on faces. On Interest packets, the IPFIX agent will create a flow entry that includes CCN-related attributes such as message type, content name, chunk number, timestamp and addresses as well as IP-layer five tuples. Similarly, on Data packets, it creates/updates the content name and its performance information such as bytes, packets, or data rate in the flow table. From an SNMP agent, we will collect the Content Store, PIT, and FIB tables.

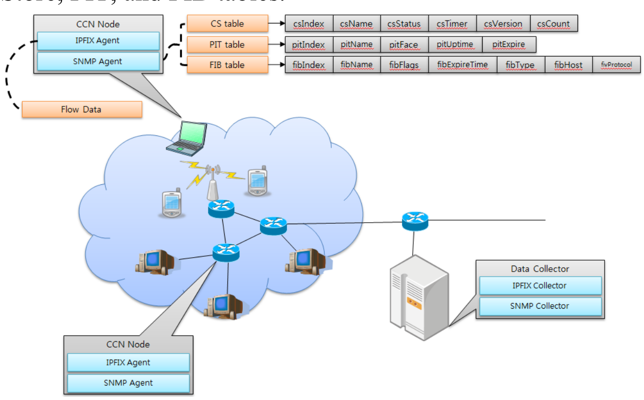

Fig 1. CCNx Monitoring Architecture

B. *IPFIX-based CCN Monitoring*

In this section, we introduce how IPFIX is extended for monitoring CCN flows. First, we define a new IPFIX template for CCN packet type, CCN contents name and chunk number. The IPFIX agent captures CCN packets on the IP network. Therefore, we assume that CCN packets are sent with the fixed UDP port number of 9695. On the CCN packet arrival, the IPFIX agent decodes a packet and parses its CCN fields. Interest or Data packets in CCN are encoded in XML. This XML structure is composed of a 4-bit header, a 3-bit value, an 1-bit terminator, and the ending delimiter for the end of XML. The IPFIX agent creates a flow data including message type, contents name and chunk number with encoding rule in Table 1.

When a flow expires with the timeout, the IPFIX agent exports the flows to the collector. After an IPIFX flow collector receives flows from agents, it saves flow data at files or database. Then, we can analyze flow performance and its related information with the flow data stored at the collector. From the flow data, we design to analyze how Interest or Data flows are exchanged on the path, which will be important for trouble-shooting or network management.

Table 1. IPFIX template attributes for flows in CCN

Attributes	Description
IPV4_SRC_ADDR	IPv4 source address
IPV4_DST_ADDR	IPv4 destination address
L4_SRC_PORT	IPv4 source port
L4_DST_PORT	IPv4 destination port
CCN_MESSAGE_TYPE	CCN message type (Interest / Data)
CCN_CONTENT_NAME	CCN content name with chunk number

C. *SNMP-based CCN Monitoring*

With SNMP, we collect various types of information on CCN nodes. In addition to CCN node information such as CPU, memory, HDD and network interface usages, we suggest add-on MIB II for CCN which includes CCN-related tables. A CCN node maintains Content Store, PIT, and FIB tables for data transfer. A Content Store table entry includes contents name and contents version. With SNMP, newly created or deleted entries in PIT or Content Store will be collected by triggering SNMP agents to send trap messages to the SNMP server.

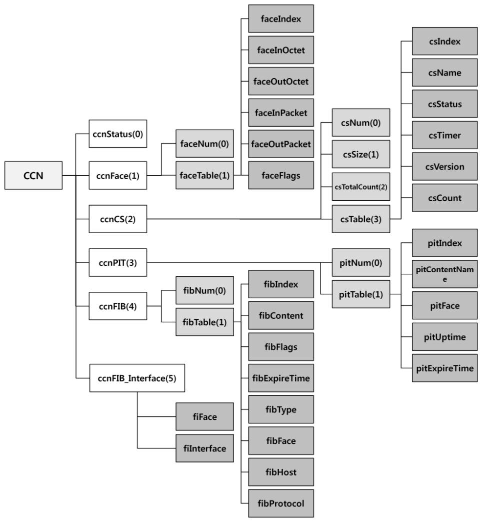

Fig 2. A MIB II structure for CCN

Fig 2 shows our CCN MIB II definition that contains tables for faces, Content Store, PIT, and FIB. We also include CCN status and faces which show face matching with interfaces. We use two types of collecting SNMP data. Periodically, the SNMP server will send get messages for tables. However, this typical period may be too long (e.g., 5 minutes) to understand real-time network events. On configured events such as new Content Store or PIT table entries, SNMP agents will send trap messages for updating table entries.

IV. Implementation

We are developing CCN monitoring agents/server with nProbe [10] for an IPFIX open source and net-SNMP [11]. As shown in Fig 3, an Interest packet will be converted by an IPFIX agent into a flow that will be exported to the server.

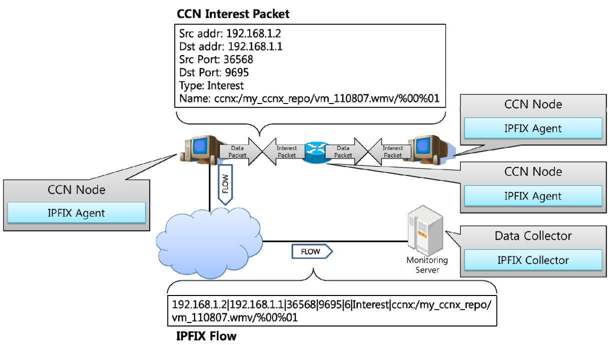

Fig 3. A CCN packet and its flow Information

We also consider the visualizer with the web interface. From the web interface displaying the network topology, SNMP tables, and flow tables, we could show the performance of content delivery in CCN networks. For example, a content-delivery path and the performance statistics such as bit rate for a flow will be shown.

Fig 6. Web interface for network monitoring for CCN

V. Conclusion

In this paper, we present a network monitoring method for CCN. By using IPFIX and SNMP, we could easily understand the current status of CCN nodes and their traffic statistics. As our work is still under progress, we plan to develop the full-version of the monitoring tool. Though IPFIX and SNMP are useful in the current IP-overlay network, they are not well suited with CCN, because IP addresses are used in IPFIX and SNMP for delivering management contents. Hence, we need to design a CCN-aware network monitoring framework.

Acknowledgment

This research was supported by the KCC(Korea Communications Commission), Korea, under the R&D program supervised by the KCA (Korea Communications Agency) (KCA-2011- (11913-05002)).

References

[1] Cisco White Paper, “Cicso Visual Networking Index: Forecast and Methodology, 2010-2015,” June 2011.

[2] J. Gantz and D. Reinsel, “Extracting value from chaos,” IDC, 2011, http://www.emc.com/digital_universe.

[3] C. Labovitz, S. Johnson, D. McPherson, J. Oberheide, and F. Jahanian, “Internet inter-domain traffic,” ACM SIGCOMM, 2010

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, “Networking named content,” ACM CoNEXT 2009.

[5] G. Carofiglio, V. Gehlen, D. Perino, “Experimental evaluation of storage management in Content-Centric Networking,” Next Generation Networking and Internet symposium, IEEE ICC 2011, Kyoto, Japan.

[6] M. Diallo, S. Fdida, V. Sourlas, P. Flegkas, L. Tassiulas, “Leveraging caching for Internet-scale content-based publish/subscribe networks,” IEEE ICC 2011, Kyoto, Japan.

[7] G. Carofiglio, L. Muscariello, and M. Gallo, “Bandwidth and storage sharing performance in information centric networking,” ACM Sigcomm Workshop on Information-Centric Networking (ICN) 2011

[8] G. Sadasivan, N. Brownlee, B. Claise, and J. Quittek, “Architecture for IP Flow Information Export,” IETF RFC5470, March 2009.

[9] CCNx, http://www.ccnx.org/.

[10] nProbe, http://www.ntop.org/nProbe.html/.

[11] Net-SNMP, http://www.net-snmp.org/.