Energy Efficient Data-centric Networking System on JGN-X Network Virtualization Platform

Satoru OKAMOTO, Daisuke MATSUBARA, and Naoaki YAMANAKA

In the New-Generation Network, Information-centric Networking (ICN) is one of the most promising technologies for communication systems which are suitable for the content data provisioning and delivery from/to users. To create a new ICN on the JGN-X network virtualization platform, developing new networking technologies such as virtual network resources control from users, in-network processing for content data creation, and energy efficient routing on the virtual network, were challenged.

1 Introduction

In future networks, a highly distributed mobile environment used for connecting to networks from a wide variety of wireless terminals including smart phones, in-vehicle equipment, sensors, etc. is predicted to become mainstream. Unlike the conventional networks, these networks are not going to be utilized mainly for communication between terminals as typified by the telephone services, but are expected to be used mainly for delivering and fetching "content data," for example web applications and video delivery. The transport method which is best suited for using a network mainly for such "data fetching" is called Information-centric Networking (ICN)[1]. The basic principle of ICN is that the user sends to the network a request for fetching, using the content name which the user wants to acquire, and the network delivers the content in response to the request. The research for implementing ICN is being done as Content-centric Networking (CCN)[2] in the USA, Network of Information (NetInf)[3] in Europe, and Data-centric Network (DCN)[4] in Japan. The three technology components which need to be developed for achieving ICN are (i) efficient technology for finding content data (overlay network technology), (ii) underlay network technology for freely combining content data in the network, to create new content data, and (iii) technology for coordinating overlay and underlay, to minimize the energy used for transport of content data. This article gives an introduction to the efforts being made to develop technology for achieving these elemental technologies: (a) achieve DCN on the network virtualization platform provided by JGN-X, (b) achieve an enhanced type of DCN that creates content which does not exist in the network through in-network processing, and (c) on the network virtualization platform, achieve an Energy Efficient and Enhanced-type Data-centric Network (E^3-DCN) that optimizes energy consumed for transport of content, by coordinating the virtual and actual networks.

2 Data-centric networking technology

2.1 CCN and NetInf

We now give an outline of CCN/NetInf. The content name (ID), not the location oriented address (e.g. IP address), is used for communication in the various types of ICN (CCN/NetInf/DCN). Thus there is no need for the terminal to know the location of the other party. Another advantage of ICN is: since the transit node stores (caches) and relays the content, there is no need of end-to-end communication between terminals. Further, as the content itself is encrypted, and it is authenticated and validated in the transit node, there is no need for authentication and coding of the end-to-end communication channel. The largest difference between CCN and NetInf lies in the routing method. NetInf uses the name resolution service to transform the flat IDs into hierarchical addresses and the routing is done by using the addresses, whereas CCN uses the hierarchical ID to do routing through the ID. Both require devising a method for dealing with a large volume of content IDs.

CCN saves the routing information by ID in the transit node, so the routing information will become voluminous if the number of contents increases. It is assumed that when a hierarchical ID is used, the routing information can be compressed by aggregating it with the help of a prefix.

However, since the different IDs having the same prefix are dispersed widely as the user or server providing the content migrates, such aggregation cannot be performed and the route information cannot be compressed.

On the other hand, NetInf involves a name resolution service in addition to the route information saving transit node. This greatly reduces the volume of route information saved in the transit node, but the information of the name resolution service must be updated whenever the content is changed (added, moved, updated or deleted). Further, since a large number of queries are sent to the name resolution service at the beginning of each communication, the name resolution service must have sufficient processing capacity for that. Communication delays cannot be avoided because of these queries.

2.2 DCN

DCN is a method designed for resolving the issue in CCN, i.e. difficulty of aggregation by means of a prefix due to content migration, without using a name resolution service. To maximize aggregation of route information of IDs of the same name space (equivalent to the prefix of CCN), DCN distributes the merge points of the route called aggregate nodes to each name space, and does the routing up to the aggregate node by hierarchical ID. The past communication route information is recorded in the transit node, to form the optimal route for reaching the destination point, without going through the aggregate node. In other words, DCN achieves the scalability of NetInf with dispersed aggregate nodes, without the help of an external name resolution server, by expanding CCN for handling M2M (Machine-to-Machine) communication on the assumption of migration.

Figure 1 shows the structure of DCN. In a hierarchical network, the transit nodes (Nodes 11–15, and Nodes 111–122) are located in the respective network segments. Transit nodes managing the name spaces (aggregate nodes

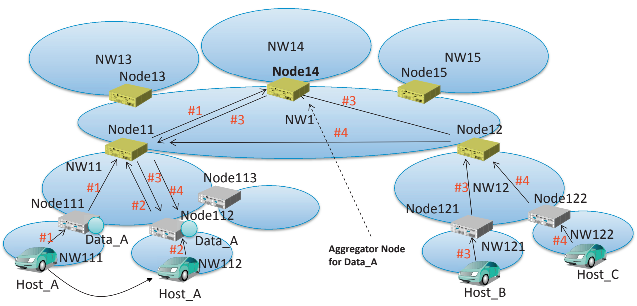


Fig. 1 Example of the DCN structure and operation

11–15) are placed in the network segments (NW 11, NW 12, ..., NW 15) below the top network segment (NW 1) so that the respective transit nodes can understand their mutual node IDs and (IP) addresses and the name spaces managed by them. Name spaces refer to all URLs (example: www.nict.go.jp/data/researchreport/) located under the specific domain name (e.g.: nict.go.jp), managed for example by DNS (Domain Name System) etc. Steps #1 to #4 in Fig. 1 show the operating procedure of DCN.

- #1 Data registration: Host_A present in NW 111 registers Data_A in Node 111. Node 111 saves Data_A and registers it in Node 11. Node 11 records the route information of Data_A (Data_A → Node 111). Node 11 knows that the aggregate node of the ID of Data_A (example: www.nict.go.jp/data/researchreport/) is Node 14, so it registers Data_A for Node 14. Node 14 records the route information of Data_A (Data_A → Node 11).
- #2 Data migration: Host_A migrates to NW 112 and registers the update data having the same ID as Data_A for Node 112, and Node 112 saves Data_A. Node 112 registers the information of Data_A in Node 11. Node 11 updates the route information of Data_A (Data_A → Node112). Thus, the latest data of Data_A is saved in Node 112. Migration of Host_A and Data_A takes place within the control domain of Node 11, so it is not registered in Node 14. Data_A exists in 2 nodes (Node 111 and Node 112), but the content is the same, so it poses no problem. If the content is not the same, the ID must be newly acquired and registered.
- #3 Data acquisition: Host_B present in NW 121 requests Node 121 to fetch Data_A. Node 121 requests Node 14 to fetch Data_A, after determining from the prefix that it is present in Node 14, via Node 12 which is a top transit node. Node 14 refers to the route information of Data_A (Data_A → Node 11) and requests Node 11 to fetch Data_A. Node 11 refers to the route information of Data_A (Data_A → Node 112) and requests Node 112 to fetch Data_A. Node 112 sends back Data_A to Host_B through the same route as that used for sending the acquisition request. On this occasion, Node 12 refers to the route information present in the returned message, and updates the route information of Data_A to the optimized route information (Data_A → Node 11).
- #4 Data acquisition through the optimized route: Host_C present in NW 122 requests Node 122 to

fetch Data_A. Node 122 requests Node 12 to fetch Data_A, and Node 12 refers to the route information (Data_A → Node 11) and requests Node 11 to fetch Data_A. Thereafter, Data_A is fetched in the same way as #3.

DCN has two features that are described here. a) Local migration process: In #2, during the migration of Host_A, the registration of information is not transmitted to Node 14, which is a transit node of the name domain to which Host_A belongs. This reduces the processing load on Node 14, and the registration time is also shorter. b) Route optimization: In #3, Node 12 updates the route information to the optimized form, so in #4, Node 12 does not send the acquisition request to Node 14. This reduces the processing load on Node 14, and the acquisition time is also shorter. The detailed quantitative evaluation of DCN is discussed in the reference document[5], in which it is reported that the load on the transit node can be reduced by up to 75% compared to CCN, and that route optimization can also reduce the transmission delay time by 30%.

The target scalability of DCN is described here. Let us suppose that the number of terminals will become 50 billion by 2020, the number of aggregate nodes will rise to 16 million (about the same as the number of DNS servers in the Internet), and the number of data in the entire network is expected to increase to 1 trillion (20 times the number of webpages in 2012). In this case, each aggregate node would manage an average of 3,125 terminals, and 62,500 data. If the route information per data is 16B, and the route information is stored in 8 GB RAM, 500 million routes can be stored. This is 8,000 times larger than the number of data that can be managed per aggregate node. Access from a big company with 40,000 employees is predicted to be about 10,000 accesses/minute[6], and presuming that one route is stored per access in the aggregate node, route information of one month can be stored in the aggregate node, which would imply that the optimized route could operate normally. In short, it means that the target DCN can be achieved by using current technology.

2.3 Enhanced type DCN

CCN/NetInf/DCN is designed to handle only content or data which is already registered on the network. In case of unregistered content or data, an address conversion error is returned in NetInf, and an unfound ID error in CCN/DCN. When the data content matching the query requested by the user is not found by Enhanced type DCN (E-DCN), the network finds the data content material, and processes the data content material in the network (i.e. in-network processing), to generate the requested data content. For example, it takes English language video of DVD quality as the data content material, converts it into a Japanese language video of Blu-ray (HD) quality, and provides it to the user. There is room for further research on how the data material is processed automatically by the network, for example, one proposal is to have the request query clearly indicate the parameters of the content after processing (such as the image format and language). The framework which generates the required data content by performing a series of in-network processes using the data content material available in the network is called Ubiquitous Grid Networking Environment (uGrid)[7]. In uGrid, data content material and in-networking process are defined as the service part, and the data content is provided to users by two mechanisms: determination of the route until the required data content is created via multiple service parts from the data content material (service routing); and the in-network service part and transport path resource reservation (service signaling). E-DCN is the outcome of adding the characteristics of uGrid to DCN. E-DCN is comprised of two networks combined: the data generation network corresponding to uGrid, and the data transport network corresponding to DCN.

3 Energy Efficient and Enhanced-type Data-centric Network (E^3-DCN)

3.1 Content transport energy consumption optimized routing

E^3-DCN is E-DCN incorporating the routing procedure for optimizing energy consumption for content transport. A prevalent routing method for optimizing energy consumption for content transport is: the content size (bits) is multiplied by the unit switching energy [J/bit] while the content is passing through the device, and the minimum energy route is selected. In E^3-DCN, a different method is incorporated for optimizing the energy consumed by the virtual network constructed on the network virtualization platform of JGN-X.

I. It provides a packet switched link and circuit (path) switched link, as the types of links between E^3-DCN nodes. In the packet switched link, the virtual link set between the nodes uses an IP router or Ethernet switch, which is a packet frame switch, for sharing the physical link and shared resources are allocated. In the circuit switched link, dedicated resources are allocated to the virtual link by the optical path switching which occupies the wavelength resource.

II. The network virtualization platform also actively saves energy. The resources for the virtual network are pre-allocated statically in the design of the network virtualization platform of the current JGN-X, but in the future, the migration of allocated resources will take place dynamically keeping the same topology of the virtual network, which is predicted to save more energy and improve the rate of utilization of resources. In E^3-DCN, it is presumed that dynamic migration will take place.

III. In the future, it will be possible to save energy by turning off the device power when the circuit switched link is not in use. This implies the objective of saving energy by minimizing use of the circuit switched link. Also, time is required to set up and tear down the optical path in the circuit switched link, so its usage time includes the content transport time + the set up and tear down time. Further, since it is a dedicated resource type, it requires [total capacity × usage time] of transport energy, to transport even one packet.

IV. Proportional distribution of energy consumed by equipment common function parts. Until now, it was presumed that the entire network is one system, so there was no need to consider energy consumed by common function parts. However, in a virtual network environment, the energy consumption for common function parts provided in the system is different when resources are shared with other virtual network users and when resources can be occupied as they are not being used by other virtual networks, so the energy consumed for transport is evaluated by performing proportional distribution. This is oriented towards saving energy for dynamic allocation of resources described in II above, by proactive sharing of the resources being used by other virtual networks.

Considering the above, for executing energy optimizing routing, formulas (1) and (2) are given respectively for calculating the transport energy when a packet switched link and circuit switched link are used.

$$J_{Packet} = \{E_{SWp} \times D \times S_p\} + \{2 \times (B_{vp}/B_{pp}) \times P_{portp} \times (S_p + 1) \times (D/B_{vp} + RTT)\} + \{\sum P_{commonp} \times B_{vp}/(B_{vp} + L_p \times B_{maxp}) \times (D/B_{vp} + RTT)\} \tag{1}$$

$$J_{Circuit} = \{E_{SWc} \times D \times S_c\} + \{2 \times P_{portc} \times (S_c + 1) \times (D/B_{vc} + RTT + \alpha)\} + \{\sum P_{commonc} \times B_{vc}/(B_{vc} + L_c \times B_{maxc}) \times (D/B_{vc} + RTT + \alpha)\} \tag{2}$$

E_{SWp} and E_{SWc} are the unit switching energy [J/bit] of the packet switch and optical path switch, D is the transmission data size [bits], S_p and S_c are the number of packet switches and optical path switches in the link, B_{vp} and B_{vc} are the capacity [bits/s] of the packet switched virtual link and circuit switched virtual link, B_{pp} and B_{pc} represent the capacity [bits/s] of the physical ports accommodating the respective virtual links [bits/s], P_{portp} and P_{portc} are the power consumed [W] by the respective physical ports and the power consumed [W/λ] per wavelength by the wavelength multiplexing port, RTT is the round trip time [s] between the E^3-DCN nodes, α is the optical path set up/tear down time [s], $P_{commonp}$ and $P_{commonc}$ are the power consumed [W] by the common function parts of the packet switch and optical path switch, Σ is the sum of S_p and $S_{s,}$ respectively, L_p and L_c are the rate of background traffic due to other virtual networks, and B_{maxp} and B_{maxc} are the maximum capacities of the packet switch and optical path switch.The link to be used for transmission between the E^3-DCN nodes is decided after determining whether more energy will be saved by using the packet switched link or circuit switched link, from data size D, link switch count S, and background traffic volume L, using equations (1) and (2). Figure 2 shows the result of average transport energy calculated by considering B_{pp} and B_{pc} as 40 Gbps, E_{SWp} and E_{SWc} as 10 nJ/bit and 0.5 nJ/bit, respectively, S_p and S_c as average 2.5 + 2 systems and average 1 + 2 systems , B_{maxp} and B_{maxc} as 320 Gbps and 2,560 Gbps, $P_{commonp}$ and $P_{commonc}$ as 225 W and 200 W, P_{portp} and P_{portc} as 300 W and 6.5 W/λ, and L as 25%–75% randomly and by fluctuating the transmission data size from 0.01 GB to 100 GB with α as 5 s. Figure 2 shows that when the transmission data size is more than approximately 300 MB, the circuit switched link is more favorable. The numbers shown in Fig. 2 are the average numbers for one link, and it is also reported from simulation of the entire network that power consumption can be reduced by up to 40% compared to when only the packet

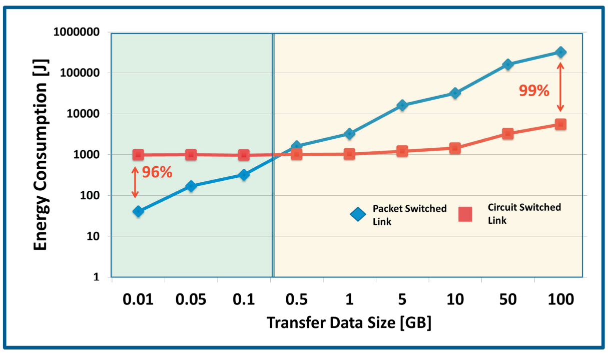


Fig. 2 Characteristics of energy consumption in the link vs. transfer data size

switched link is used[8]. The boundary size is highly dependent on α and *S*. For each numerical value assigned as a parameter, the information of the physical network configuration device is required, which is difficult for virtual network users to obtain. That is why E^3-DCN is designed with a network API (Application Programming Interface) and it was a challenge to procure the parameters by linking with the management system of the JGN-X network virtualization platform.

3.2 Network API

A virtual network is composed of virtual nodes and virtual links. As a virtual node, there are a virtual server node that can utilize computing resources, and a virtual transport node for achieving a virtual IP router or virtual Ethernet switch. DCN and E^3-DCN which are built on the network virtualization platform have a transit node (of DCN) in the virtual server node. In DCN, the transit nodes are connected with a virtual link, whereas in E^3-DCN, the transit nodes are connected using a virtual link and virtual transport node. The communication capacity and connection destination of the virtual link in a virtual network are defined, but the information of the type of link used or route taken or the number of devices the content passes through in the actual network cannot be acquired. This kind of information is not necessarily needed for most virtual network users, but it is useful for users who want to know the congestion situation of the actual network like E^3-DCN or to use the packet or circuit switched virtual links for different purposes. In case of virtual network users who operate like self-organized networks, there will be growing demand for interaction between the operating virtual network and the network virtualization platform, for example, requests for dynamic generation or deletion of virtual links or virtual nodes, and generation of virtual links with delay time constraints or capacity constraints. Network API is the interface for implementing this kind of interaction. As the first step to implementing network API in E^3-DCN, the interfaces for acquiring information from the service network controller (SNC) and transport network controller (TNC) which are present in the management system of the JGN-X network virtualization platform were jointly developed.

4 Proof of concept demonstration of DCN and E^3-DCN on JGN-X

4.1 DCN demonstration

We constructed a DCN testbed as shown in Fig. 3, by using the network visualization platform of JGN-X. Seven nodes of the network visualization platform were used, to achieve 37 DCN virtual nodes which were connected to 336 virtual terminals.The network topology consisted of a 5-hierarchical network, simulating the layout of the general ISP backbone network, and the DCN node in the 3rd hierarchy was the aggregate node. Figure 3(b) shows the data transmission topology; no link is provided for direct connection between the aggregate nodes. However,

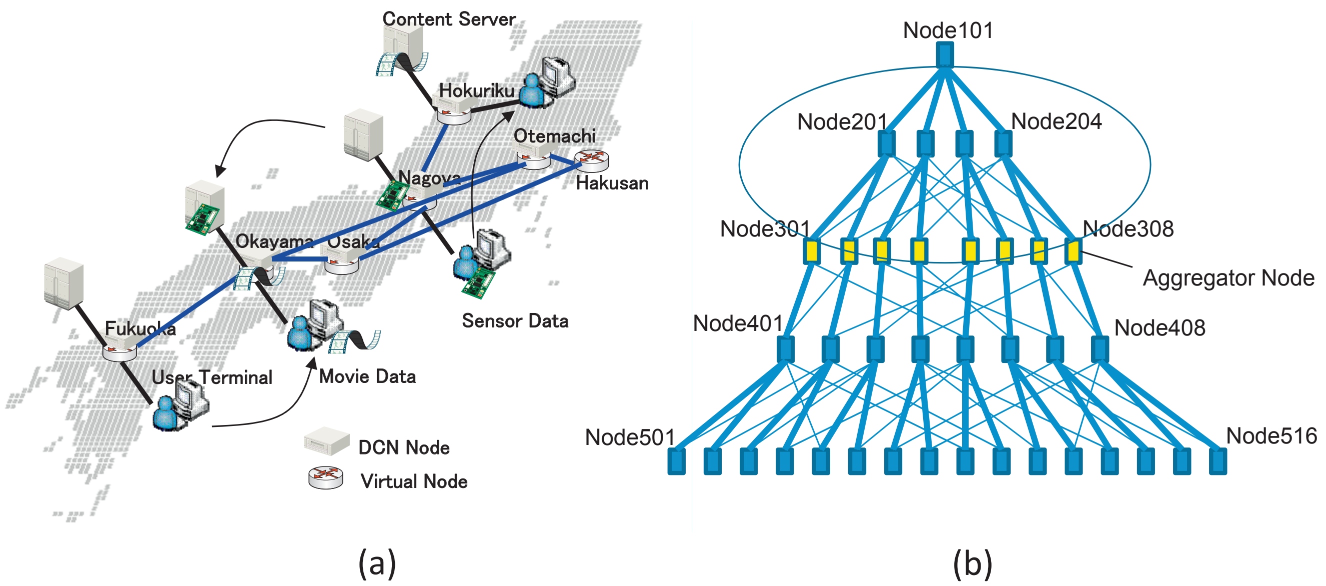


Fig. 3 DCN testbed. (a) Wide area testing system, (b) Topology of 37 DCN nodes

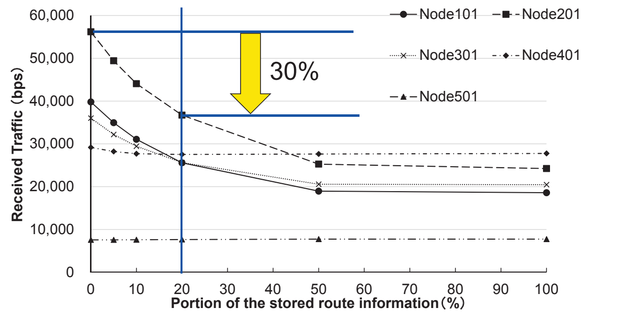


Fig. 4 Traffic load corresponding to the route information holding rate of transit node

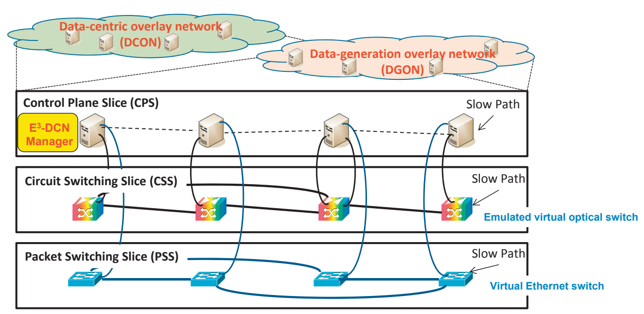


Fig. 5 Outline of the architecture of E^3-DCN

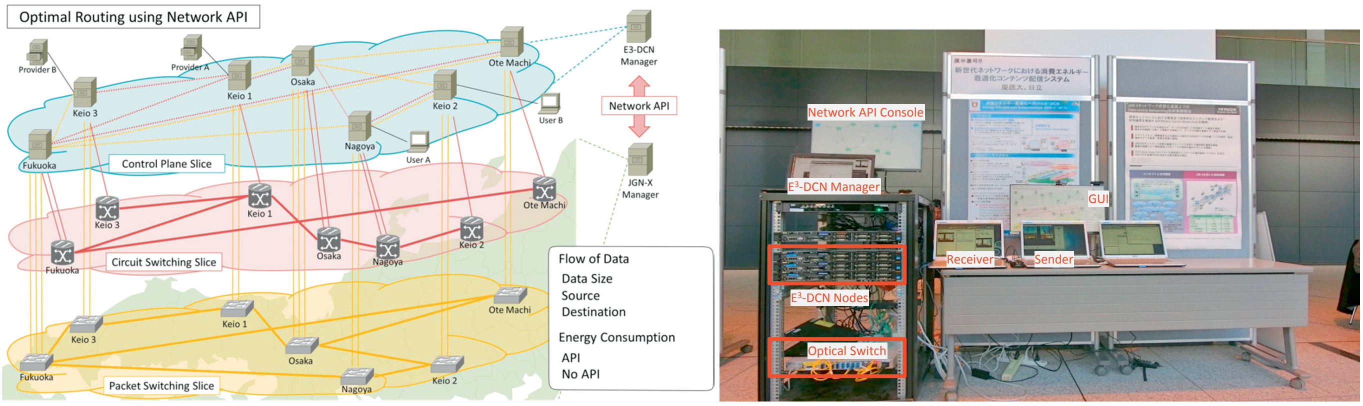
as the control network, all the aggregate nodes are placed adjacently, using IP, etc., and message exchange for route acquisition is done within this control network to reduce the burden of message transmission on the nodes. The average degree of this topology also including the logical links for contiguity between the aggregate nodes is 5.19 (maximum 11), and the average degree of the general IP backbone network is similar at approximately 2.14 to 5.02. In this demonstration, presuming delivery and acquisition of vehicle running information, we simulated a case where the vehicle is traveling at a speed of 60 km/h, moving in a 15 km diameter wireless area, and the running information is registered and acquired in 20 second intervals. The communication is done via DCN by moving 320 virtual acquisition terminals and 16 virtual registration terminals. The acquisition and registration terminals are uniformly distributed and connected to 16 DCN nodes of the 5th hierarchy, and moving after every 900 seconds to the next DCN node, i.e., Node 501 → 502 → … → 516 → 501. The data is registered in the connected DCN node, in 20-second intervals from the registered terminal. As for the data IDs to be registered, 25 types of data IDs of the data of different name space of each terminal (www.ex1.com to www.ex16.com) are randomly registered in one node. Each aggregate node from Node 310 to Node 308 manages two different name spaces. The acquisition terminals transmit the randomly acquired data IDs in 20 second intervals.

An example of a testbed result is shown in Fig. 4. The volume of data acquired from the optimized route increases with the increase in the route information stored by the DCN transit nodes, so the load reduces on Nodes 101, 201 and 301 which are in the top hierarchy. When the rate of stored information is 20%, the load on Node 201 can be reduced by 30%. As described in Subsection **2.2**, when the target number of aggregate nodes is 16 million, it is impossible to establish a full mesh logical link between the aggregate nodes. As a result, in a larger DCN, the rate of reduction of the load on the nodes is likely to be less.

4.2 E^3-DCN demonstration

4.2.1 Installation of E^3-DCN in JGN-X

E^3-DCN establishes a connection between its nodes using two types of links: a packet switched link and circuit switched link. We constructed one E^3-DCN using three slices present in the virtual network provided by the network virtualization platform of JGN-X. The outline of the architecture of E^3-DCN is shown in Fig. 5. The first slice is the control-plane slice (CPS) in which the virtual server node (slow path) provides the E^3-DCN node main function, and the virtual link provides the E^3-DCN control network. The second slice is the circuit switching slice (CSS), and it provides the circuit switched link. However, the current JGN-X does not have the function to provide a circuit switched virtual network, so we made a bridge connection between two interfaces of the virtual transport nodes to emulate an optical path switch and thereby provide a circuit switched link. The third slice is the packet switching slice (PSS) and it provides the packet switched link. The virtual transport node in the slice works as an Ethernet switch, and provides the function to switch Ethernet frames among the interfaces. In order to achieve E^3-DCN, the E^3-DCN node's main function in CPS, the circuit switched link provided by CSS and the packet switched link provided by PSS must be connected. To realize this, the Network aCcommodator (NC) equipment provided by the network virtualization platform was used. The E^3-DCN node transports the content to the adjoining E^3-DCN node via NC, through the circuit switched link (CSS link) provided by CSS and the packet switched link (PSS link) provided by PSS. The E-DCN function is provided by two overlay networks made by connecting E^3-DCN nodes, i.e., the data-centric overlay network (DCON) and the data-generation overlay network



(a) (b)

Fig. 6 E^3-DCN Demonstration. (a) GUI screen, (b) Photo of the demonstration system

(DGON). In the demonstration, we constructed an E^3-DCN with a total of seven nodes, consisting of four virtual nodes on the network virtualization platform (Fukuoka, Osaka, Otemachi, and Nagoya), and three actual nodes installed in Keio University. To expand the virtual links from NC to Keio University, JGN-X L2 service was used. At the IEICE technical conference on network virtualization in March 2015, we performed a live demonstration of the functions for sending pictures through DCON and processing video pictures through DGON (Fig. 6). In the demonstration trial, we presumed that all the parameters required for calculations in equations (1) and (2) could be fetched through the network API, and established that different links have to be used according to the size of the content data to be fetched.

4.2.2 Network API trial

In the Network API demonstration, we worked on the challenging task of obtaining the abstracted information from the physical configuration information (described in XML) of the network resource providing the virtual network, by accessing the emulated SNC and TNC from the virtual nodes in CPS. The original SNC and TNC are installed in the JGN-X Hakusan site and emulated ones are installed in the Keio University site. As the basic idea for network virtualization, we need to solve the fundamental problems that the slice is a closed system, making external access difficult, and that SNC and TNC are not supposed to be made open to users. Fortunately, it was confirmed that: the virtual nodes of a slice are accessible through the IP network called the Z-Plane from the slice operator's developer terminal (DT), the Y-Plane can be used to access the network virtualization platform operation portal from DT, and SNC and TNC exist in the Y-Plane. Using this relation, communication for the Network API can be achieved by developing a proxy for implementing the communication route from DT → Z-Plane → virtual node → Z-Plane → Proxy → Y-Plane → SNC and TNC. Direct communication to the Z-Plane from a virtual node is not possible, so it is required to login to the Z-Plane once from DT (condition 1). It is necessary to login to the Proxy from the virtual node (condition 2). Access to SNC and TNC must be authenticated (condition 3). These three conditions have helped to protect against unauthorized access from the virtual network and establish a secure communication path.

In this trial, it was not possible to fetch all the required parameters shown in Subsection **3.1**. We were able to fetch some parameters, such as maximum capacity B_p, number of switches S, and the load on the virtual node providing system. We found that in order to fetch all the parameters, it is necessary to access the management system that controls the entire JGN-X, and that an API for accessing the management system from SNC and TNC is also necessary.

5 Conclusion

In order to establish a data-centric networking technology to suit the ways future networks will be used, we performed proof of concept of E^3-DCN and DCN on JGN-X, and confirmed that the basic technology can be implemented. The next issue is the development of an operations management technology for practical use of DCN, and the deployment of a new virtual network utilizing technology using a network API.

References

1 Information-Centric Networking Research Group (ICNRG): http://irtf.org/icnrg

2 V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," Proc. in CoNEXT `09, pp.1–12., Dec. 2009.

3 C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of Information (NetInf) – An information-centric networking architecture," Elsevier Journal Computer Communications, Vol.36, Issue 7, pp.721–735, April 2013.

4 D. Matsubara, H. Yabusaki, S. Okamoto, N. Yamanaka, and T. Takahashi, "Proposal of Data-centric Network for Mobile and Dynamic Machine-to-Machine Communication," IEICE Trans. on Commun., Vol.E96-B, No.11, pp.2795–2806, Nov. 2013.

5 D. Matsubara, S. Okamoto, N. Yamanaka, and T. Takahashi, "Evaluation of Information-Centric Networking in Mobile and Distributed Environment Using Wide-Area Test Bed," Proc. APNOMS2014, TS-3-1, Sept. 2014.

6 A. Wolman, G. M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy, "On the scale and performance of cooperative Web proxy caching," 17th ACM Symposium on Operating Systems Principles (SOSP'99), pp.16–31, Dec. 1999.

7 S. Okamoto, Y. Arakawa, and N. Yamanaka, "Proposal of the Ubiquitous Grid Networking Environment (uGrid)," IEICE Society Conference, B-7-15, Sept. 2007.

8 S. Zhang, H. Takeshita, S. Okamoto, and N. Yamanaka, "Energy Efficient and Enhanced-type Data-centric Network Architecture," International Journal of Computer & Information Science (IJCIS), Vol.16, No.1, pp.60–70, March 2015.

Satoru OKAMOTO, Dr. Eng.

Researcher, Keio Leading-edge Laboratory of Science and Technology/Specially Appointed Professor, Graduate School of Informatics and Engineering, The University of Electro-Communications
Photonic Network, Network Control, Network Architecture

Daisuke MATSUBARA, Dr. Inf.

Senior Researcher, Center for Technology Innovation-Information and Telecommunications, Hitachi, Ltd.
New Generation Network, IoT, M2M Communication System

Naoaki YAMANAKA, Dr. Eng.

Professor, Faculty of Science and Technology, Keio University
Photonic Network, Network Protocol, Smart Network