A Real-time Large-volume Content Delivery System using Globally Distributed Storage

Tsuyoshi OGURA, Kunitake KANEKO, Masahiko KITAMURA,
Hiroyuki KIMIYAMA, Tatsuya FUJII, and Atsushi TAKAHARA

We are studying the method of developing a platform that delivers large-volume contents efficiently in a many-to-one transmission style, by combining network virtualization technology and the handling of LDGM error correction code. This paper describes its key technologies such as packet routing by using multiple virtual networks, data transmission using redundancy coding, and evaluation of communication qualities, and shows their current outcomes.

1 Introduction

In the digital content market, with globalization of supply chains and increasing demand for swift provision of content, the need for innovative platforms—capable of transmitting and distributing digital content much more efficiently than the conventional technologies in place on the existing Internet—is increasing. We consider an approach based on a many-to-one content distribution scheme[1]—characterized by distributed storage of large-volume data on multiple storage devices and on-demand gathering of necessary data for browsing—to be highly promising to meet such demand because the dividing of content data leads to the following: smaller bandwidth requirements in the regions near transmission sites, lower probability of excessive loading on storage devices due to concentrated access to a particular content, higher safety against data leakage, and better fault-tolerance. However, due to the increased number of flows characteristic of many-to-one type delivery systems, TCP (especially for retransmission processing) or other conventional traffic control methods become inappropriate for guaranteeing communication quality as these tend to be more complex and to increase control load.

In this research, we propose an approach to establish a platform capable of highly efficient transmission quality control taking advantage of the many-to-one type delivery. It employs data redundant technology (by means of error correction code) and virtual network technology, assuming the use of a transmission protocol that does not require retransmission control.

The proposal consists of three element technologies (see Fig. 1): “Multi-slice based traffic control technology” that achieves efficient traffic diversion through combined use of multiple routing planes, or virtual networks (i.e. slices); “Redundant encoding and transmission technology for

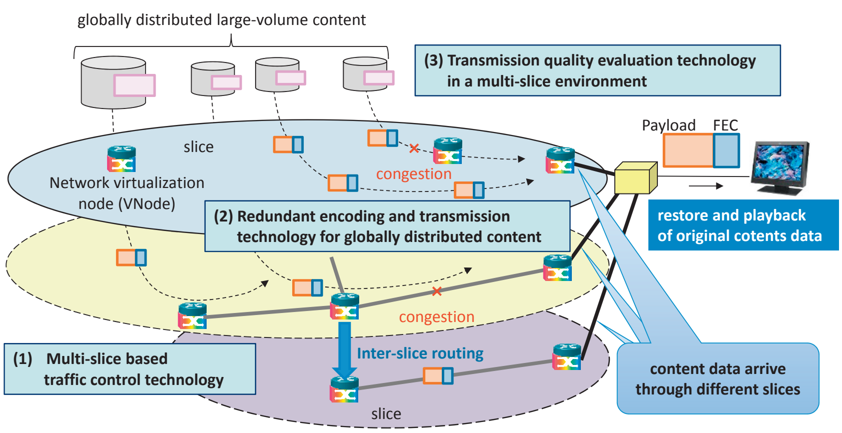


Fig. 1 Component technologies in our system

globally distributed content" that enables robust data transmission and dynamic traffic volume control through the use of error correction code and its processing on network nodes, and, "Transmission quality evaluation technology in a multi-slice environment" that enables traffic quality evaluation in the many-to-one type content delivery system that includes the two technologies mentioned above. This paper provides an overview of the proposed technologies and their verification results.

2 Multi-slice based traffic control technology

2.1 Problems with conventional technologies

In general, there are two methods available to distribute digital content, such as video streams, without suffering from any degradation in transmission quality: one that reserves bandwidth resources along the transmission route beforehand, and the other that explicitly modifies the transmission route in response to problems occurring en route, such as heavy traffic and line congestion. The former method is not suitable for the many-to-one type distribution because of the complexity of the transmission path topology. The latter method—often called explicit routing—includes widespread technologies such as OpenFlow[2] and Multiprotocol Label Switching (MPLS)[3]. OpenFlow is basically a flow-by-flow control mechanism incapable of sub-flow fine manipulations, i.e. dividing a traffic flow into arbitrary sized blocks for allocating them to multiple of routes. In addition, in hop-by-hop type configurations, flow tables of all switches along a newly established transmission route must be modified. As for MPLS, a fundamental principle that enables handling multiple Label Switch Path (LSP) instances on a packet-by-packet basis is provided. However, detailed definitions for its actual implementation are not provided. In terms of changing transmission path topologies, users can change them—except for the case of Fast Rerouting to bypass failed locations—only by switching LSPs being used at an initial data-sending point, or, by end-to-end LSP reconfiguration, because MPLS is incapable of partial modification of an established LSP. In these circumstances, users cannot flexibly change the transmission path of each packet in the middle of its transmission.

2.2 Proposed traffic control technology

In this research, we propose a new traffic control technology characterized by combined use of multiple virtual networks, in order to enable more flexible traffic control with less control load than the conventional technologies described above. The virtual network in this context represents one of the logical networks overlaid on a physical network in a mutually independent fashion. An illustrative implementation of such virtual networks is shown in the system described in [4].

Unlike the conventional technologies that perform route modification or path re-establishment within a monolithic network, the new technology introduces a packet-by-packet route diversion scheme in which a packet crosses between virtual networks during the data transmission process on an as-needed basis. As each virtual network has its own bandwidth resources separated from those of the others and its own routing plane, a transfer of a portion of traffic from one virtual network to another entails the use of topology and routing of the virtual network to which the packets move. This mechanism enables route modifications and traffic dispersion without resorting to complex route searching and path setup signaling procedures.

Specifically, the proposed traffic control technology

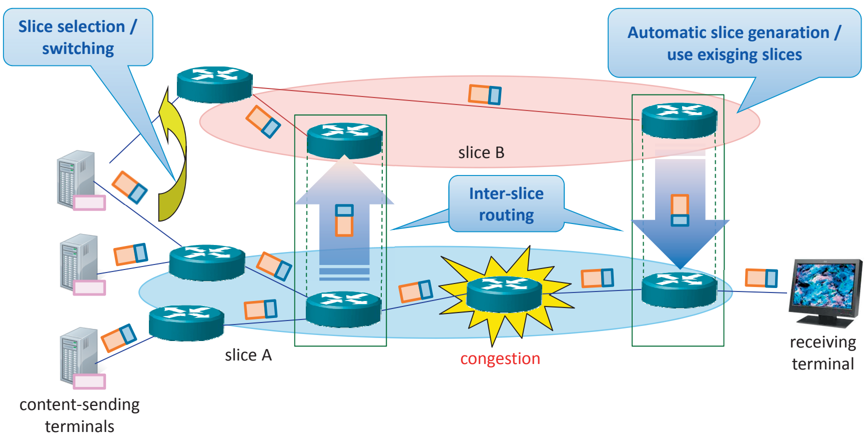


Fig. 2 Traffic control based on a combination of multiple slices

utilizes a combination of basic operations shown in Fig. 2 to handle multiple virtual networks (hereafter, each is referred to as a slice). The most significant operation is the inter-slice routing. This is an operation used, in case traffic congestion takes place in the currently used slice, to divert all or a part of traffic to a different slice, thus enabling bypassing the bottleneck. The destination slice, to which packets are transferred, should be chosen on the ground that the slice ensures efficient packet transfer, supported by sufficient bandwidth en route, as long as packets transfer along the IP routing path defined by the IP routing protocol running in the slice. The slice selection/switching in the figure represents an operation that determines the first slice to which the data is inserted after being sent from the content-sending terminal. This is equivalent to selecting/switching LSPs at a content-sending terminal in MPLS. The automatic slice generation represents an operation in which a new slice is dynamically generated for use when, while performing the abovementioned operations, an appropriate slice can no longer be found.

The proposed scheme entails frequent packet reordering because multiple slices are used in transmitting packets of a flow. Therefore, we assume the use of a UDP-like protocol instead of TCP as the transport layer protocol in order to avoid throughput degradation caused by the packet reordering.

2.3 Implementation on JGN-X network virtualization platform

2.3.1 System architecture

We implemented a system based on the proposed scheme on the JGN-X[5] network virtualization platform[4]. Figure 3 shows the overview of the system. This implementation adopted a central control model in which a control unit commands the view of all the traffic in the slices and launches the appropriate basic operations previously mentioned. The congestion avoidance controller in the figure represents the control unit on which the control software—the embodiment of the proposed scheme—is installed. In addition, a software router, which runs within a Node Sliver (NS) on a VNode, is implemented to realize the inter-slice routing and other functions. We used a router-development tool[6] in the implementation.

The Service Network Controller (SNC) is one of the management nodes of the JGN-X network virtualization platform. It provides a command API that is designed for the control of configuration of virtual networks on the platform. The congestion avoidance controller uses this API (by way of the Y-Plane: an L2 VLAN for control communication) to acquire information on slice topologies or others, and to launch slice-controlling operations such as establishing an inter-slice routing link (i.e. a data-transmission link between different slices). In this research, we newly added a function to establish the inter-slice routing link to the JGN-X network virtualization platform in order to implement our novel traffic-control scheme—the inter-slice

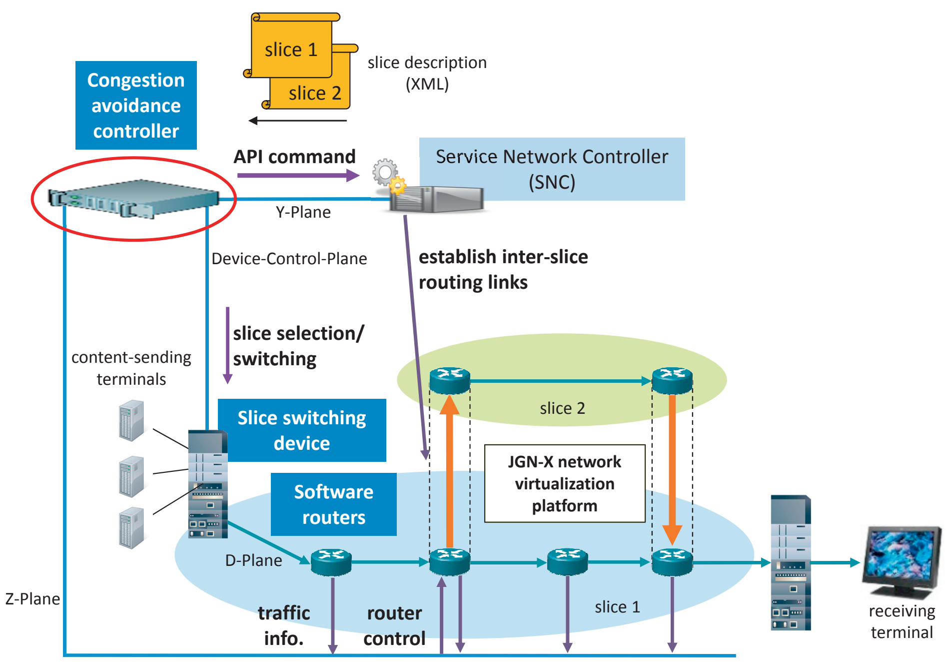


Fig. 3 System architecture

routing.

The congestion avoidance controller uses the Z-Plane, another L2 VLAN, to acquire traffic information from software routers and to control the packet forwarding behavior of them. In addition, the controller issues, by way of the Device-Control-Plane, commands to the slice switching device[7] regarding selecting (or switching to) a slice to which each content-sending terminal belongs. The slice switching device is a gateway device that provides a connection between a user site and the virtualization platform, and is capable of dynamically changing the slices that contain user-side terminals. The slice selection/switching function described in Subsection **2.2** is realized using this device.

2.3.2 Traffic control

A conceptual overview of the traffic control is shown in Fig. 4. The congestion avoidance controller checks, referring to the traffic information obtained from software routers running on the VNodes' NSs, if there is sufficient bandwidth in the transmission route in the slice currently in use. In case a bandwidth shortage is detected at any of the Link Slivers (LSs) along the data transmission route, such an attempt is made as to determine the range that contains all such LSs and to divert all or a part of the traffic—incoming to the NS that is located in the origin (upper most point) of the region—to other slices.

The first step of the traffic diversion is searching for a candidate slice for the diversion, i.e. a slice that contains two relevant VNodes: one that contains the origin NS of the target range, and the other that contains the terminal NS of the range (note that there may be cases where the terminal point is a slice switching device—such cases lie beyond the scope of this paper). Only such slices are qualified because,

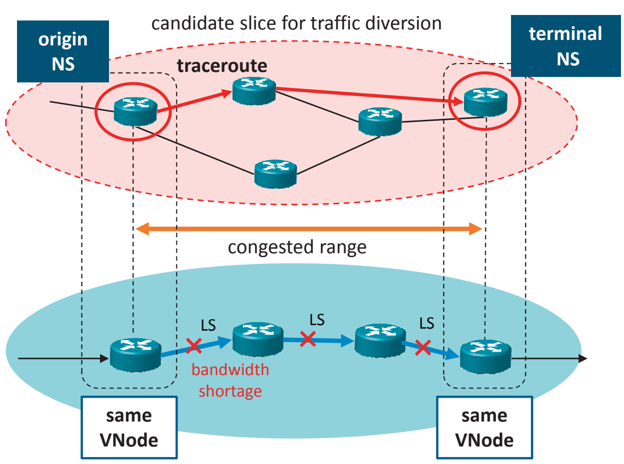


Fig. 4 Traffic control

in the current implementation, inter-slice routing links can be established only between the NSs that belong to the same VNode.

If a slice with the qualification is found, then a check is made to measure the possible traffic volume that can be diverted to the candidate slice. As the inter-slice routing proposed in this paper assumes that the diverted packets are transferred along the IP routing path of the new slice, the volume of traffic that can be diverted is determined by the end-to-end amount of available bandwidth along the IP routing path. There are various methods for the available bandwidth measurement, however, we implemented a function in the software router: the traceroute-based function that carries information on available bandwidth of LSs, enabling calculation of the end-to-end amount of available bandwidth.

According to this information, the congestion avoidance controller determines the traffic volume to be diverted to the slice, and establishes an inter-slice routing link on an as-needed basis. Following the establishment of the link, it issues commands to instruct the relevant software routers to change their packet transfer behavior. The command include such elements as: routing of a part of the packets to the specified slice, change of the destination address, routing back of the packets to the original slice, and recovery of the destination address. These procedures—search of a qualified slices followed by diversion of packets—are repeated until the bandwidth shortage in the original slice is resolved. In addition, attempts may also be made, on an as needed-basis, at widening the target range of LSs.

When the slices qualified for the inter-slice routing cannot be found or their bandwidth has already been exhausted, the congestion avoidance controller performs the slice switching—from the one that contains the content-sending terminal to another. If this does not resolve the bandwidth shortage, the controller tries to create new slices by way of the SNC's API and uses them (NOTE: the slice creation capability is not implemented in the current system).

2.3.3 Experiments

A measurement was made, using the API provided by the SNC, to evaluate the time required to generate an inter-slice routing link, and the result was approximately 5.8 seconds. This indicates that the state of bandwidth shortage persists at least for that length of time and that the degradation of content transmission quality (e.g. loss of packets) can hardly be prevented, even if the system activates congestion avoidance control at the moment it detects

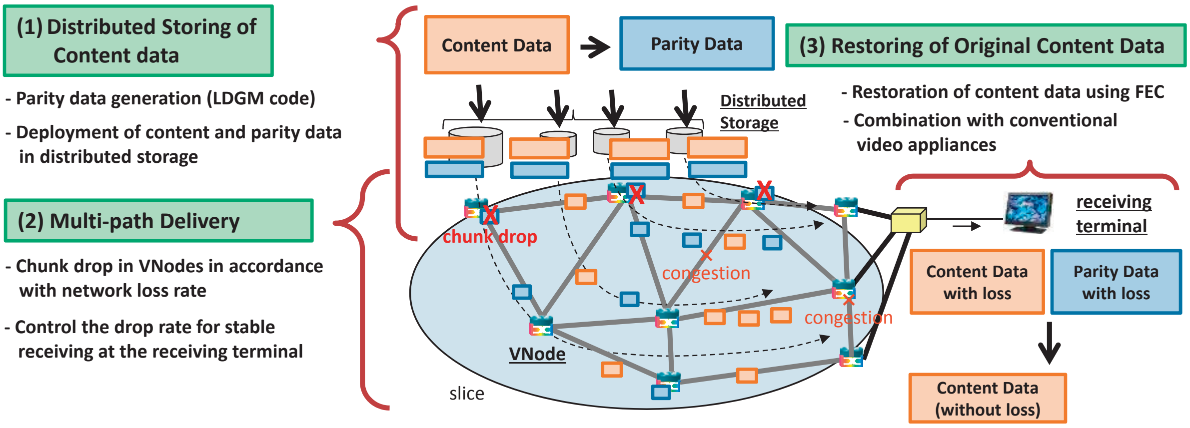


Fig. 5 System overview of distributed storing and delivery by employing forward error correction scheme with LDGM code

the bandwidth shortage. Therefore, as long as we assume the use of the current JGN-X virtualization platform, early prevision and preemptive activation of diverting control (inter-slice routing) are essential to obviate the occurrence of bandwidth shortage.

So far, the multi-slice based traffic control technology has proved the viability of stable delivery of compressed 4K images (approx. 170 Mbps). In this experiment, we used our JPEG2000 codec device[8] and the software router previously described.

3 Redundant encoding and transmission technology for globally distributed content

3.1 Distributed storage using LDGM code

This section describes an outline of distributed storage that uses error correction code, especially Low Density Generator Matrix (LDGM) code. Distributed storage represents a storage method in which the source file (original information) is processed with error correction code to make it redundant, and the resultant data is divided into multiple data units (chunks) for storage in distributed locations. Advantages of this approach include better fault-tolerance (in terms of host and site), distributed load, and others. Among these features, we focus on its affinity to distributed delivery—i.e. this configuration – the content-sending terminals (storage hosts) are distributed within the network—has potential for use in multi-network-path based transmission.

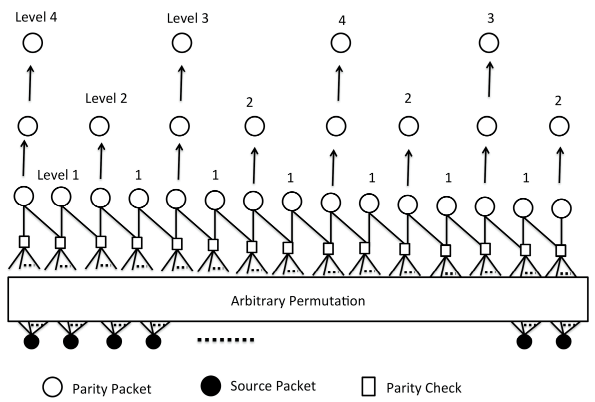
Simple parity check (as in RAID) and Reed-Solomon (RS) code have often been used as the error correction code for distributed storage[9]. In this research, we employ LDGM code, whose encoding/decoding time is linearly related (O(n)) to the content's data size (n). We consider this property is most suited to real-time broadband applications such as video delivery. In addition to high speed encoding, the LDGM code allows flexible settings in terms of code length and redundancy (code rate)[10]. These characteristics make it easier to accommodate VBR (variable bit rate) image applications. Because coding/decoding of LDGM requires a packet-by-packet (e.g. UDP packet) calculation, we set the chunk data size equal to the packet size in order to enable faster processing. This feature enables low-load transmission for the content-sending terminal: it has only to add a packet header (such as an IP/UDP header) before sending out the chunk.

In the distributed storage scheme using LDGM code, the receiving terminal receives chunks of data from each of the content-sending terminals, and restores an original source file by applying LDGM decoding. Namely, each chunk of data (or packet) as a whole, stored in distributed locations, is equivalent to a code block. Therefore, the number of chunk packets contained in a code block, or code length n, tends to become very large. The high speed performance of LDGM code is essential in view of this as well[11].

Focusing attention on the programmable nature of VNodes and the properties of LDGM, we are studying a method to dynamically control redundancy of chunk data in response to the variations of traffic volume within the network and trying to develop a system that allows users to receive a stable stream of video data (Fig. 5)[12][13].

3.2 Redundancy control using VNode's programmability

When error correction code is used to compensate packet losses on the network, the content-sending terminal normally determines the redundancy, with consideration of the current situation of the network traffic, before it starts sending the packets. However, it is difficult to determine optimum and fixed redundancy, because the traffic volume always changes at any time and place. To address this problem, we have proposed a redundancy control method that can take the real-time network situation into consideration by exploiting the programmability of VNode. In this method, the content-sending terminals send out chunk packets after assigning them with enough redundancy, and each VNode discards some of them based on the degree of congestion taking place on the network. This enables dynamic redundancy adjustment. The discard rate is determined by calculating an adequate redundancy value to compensate packet losses within the network (packet loss ratio is notified by the measurement system). In the system that uses LDGM code, the number of received packets required to decode correctly can be approximated statistically as $(1+\delta)nr$, where n represents the code length and r represents redundancy. The discard rate can be determined from it.



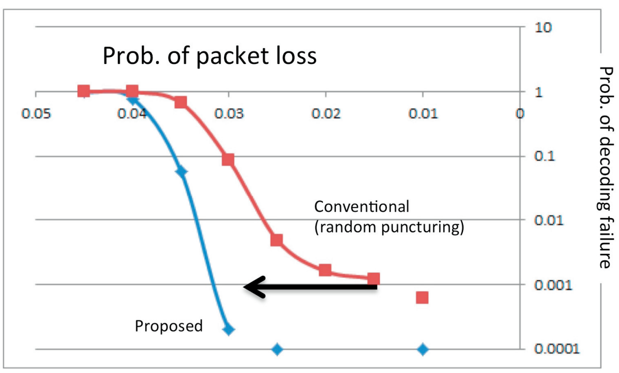


Fig. 6 The performnce of stable restoration by classed chunk dropping operation based on the graph structure of LDGM code

3.3 Improving delivery service quality by introducing discard priority

Arbitrary packet discard on a VNode is, from the viewpoint of using error correction code, equivalent to naturally-arising random packet losses. Examination of the decoding process of LDGM coded packets reveals that the decoding performance (decode success ratio against the same number of received packets) can be improved if the positions of lost packets can be estimated beforehand. Therefore, by giving additional information (i.e. discard priority) to a chunk packet when it is generated using LDGM coding, and allowing the VNode to selectively discard packets, decoding performance can be improved significantly. The method of discard priority assignment to a chunk packet is equivalent to the approach shown in the code diagram of LDGM coding: it is classified using a packet combination so that the minimum loop between each coded node does not decrease if the designated packets in the code block are lost. Introduction of this scheme into the distributed-storage based video delivery system generated a significant performance improvement—successful reception, or decoding, was achieved even when there is a larger number (2.5 times) of packet losses compared to the case where conventional arbitrary packet discard was used—leading to a higher level of service quality (Fig. 6).

3.4 System performance

We constructed a distributed-storage based video delivery system using the methods described above. We used 16 distributed content-sending terminals and a multi-pass network configuration with a total bandwidth of 20 Gbps. Performance evaluation carried out on this system showed that a total of 16 Gbps (equivalent to 10 uncompressed HD videos) video streams can be transmitted with no less than 90% decode success ratio.

4 Transmission quality evaluation technology in a multi-slice environment

In this section, we further discuss the effects of the technologies described in Sections **2** and **3** on large-volume content delivery from the aspect of communication quality. Specifically, we discuss the following: the effect of packet-by-packet traffic diverting, and the effect of packet discard priority. In this research we used an experimental network that simulates the network virtualization platform.

4.1 Communication quality in many-to-one delivery system

The many-to-one type delivery scheme is so different from the conventional TCP-based content delivery that the existing communication quality indices (file transfer throughput, network delay, jitter, and others) cannot be applicable in a straightforward manner. In this section, we discuss communication quality indices essential to the many-to-one delivery scheme from two different standpoints: one from content receivers, and one from network operators.

From the viewpoint of content receivers, communication quality should be evaluated from the success ratio of content reconstruction (file restoration probability) and the burden placed on a receiving terminal. The file restoration probability indicates the ratio of successful restoring of playable content through decoding. For example, if the receiving terminal receives 100 files and it successfully restores—regardless of executing error correction or not—all of them to a replayable state every time, then the file restoration probability is 100%. If it successfully restores 50 files but failed in 50 trials, then the probability becomes 50%. Generally, a high file restoration probability entails a high packet arrival rate. The burden placed on a receiving terminal represents the computational or operational load it must bear to receive and restore (error correction) the content. Where LDGM is used for error correction, the computational complexity tends to increase as the packet loss ratio increases, meaning larger computational load on the receiving terminal. On the other hand, the operational load to receive packets tends to increase as the packet loss ratio decreases. In this paper, we focus our discussion on the file restoration probability.

From the viewpoint of the network carriers and providers, reducing of total traffic volume, while maintaining communication quality for content receivers, could enable more efficient network resource utilization. In addition, at the time of unexpected traffic increase, if traffic can be diverted to another transmission path or a virtual network that has enough bandwidth, instead of introducing new communication circuits, higher line utilization rate can also be achieved.

4.2 Effect verification of packet-by-packet diversion scheme

In this section, we discuss the effect of the packet-by-packet route diversion scheme described in Section **2**. Performance evaluation was carried out using the Order Insensitive Flow (OIF) router, which is under research and development at Keio University. The OIF router is a software router constructed on the assumption that it runs on a VNode. Here, Order Insensitive Flow (OIF) is defined to be a type of packet flow in which packet reorders exert no effect on the application program, and Order Sensitive Flow (OSF) is defined as the type of packet flow in which packet reordering has an effect on the application program. The target of this research—large volume content delivery traffic of distributed storage data using UDP—is classified as OIF. Let us start with a simple explanation of an OIF router's operations. The OIF router discriminates between OSF and OIF from the destination port number in a packet. Then, it checks if congestion is taking place in its output interface. If no congestion is detected in the output interface, the router passes both OIF and OSF packets to normal routes. If congestion is taking place in the output interface, it passes OIF packets to a diversion route and OSF packets to a normal route, to an effect of avoiding short term traffic congestion that may result in packet losses. The routers are assumed to have diversion route information beforehand (they share the information

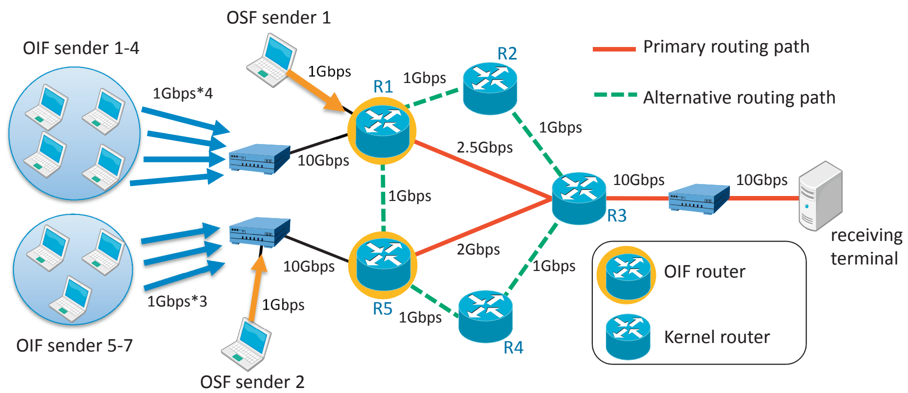


Fig. 7 Evaluation environment for packet-based detour with OIF router

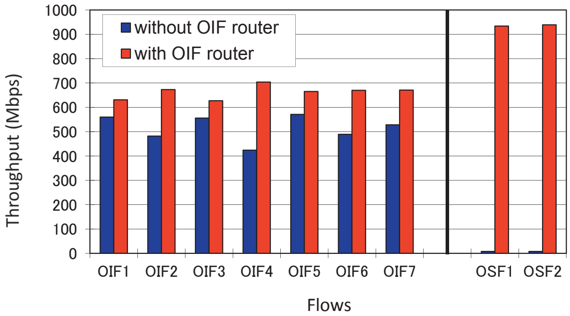


Fig. 8 Throughput comparison between conventional routing scheme and packet-based detouring scheme with OIF router

through mutual communication).

In this research, we evaluated packet arrival rate to a receiving terminal using the OIF routers. The OIF routers are PC-based software routers running on Ubuntu 12.10. The OIF routers, and the kernel routers that execute normal transfer processing, are configured as illustrated in Fig. 7. In Figure 7, R1 and R2 are allocated to the virtual network A, and R4 and R5 to the virtual network B. R3 represents the gateway on the receiving terminal, connected to both of the virtual networks (A and B). In this environment, the 2 Gbps links directly connecting R1/R5 to R3 represent the normal routes, and the routes connecting R1/R5 to R3 via R2/R4 represent the diversion routes. In this evaluation experiment, the sending terminals (OIF senders and OSF senders) are configured to generate congestion on the normal route output interface of R1 and R5. Comparison of effective throughputs for OSF and OIF are shown in Fig. 8: one set of data is obtained by activating OIF diversion on R1 and R5, and the other uses only a conventional router (kernel router) function on them. This figure shows that OIF does not interfere with activities of OSF, and that the throughput of OIF itself is improved with the aid of alternative diverting routes. These results clearly indicate that the use of the packet-by-packet route diversion enables efficient large volume content delivery through effective exploitation of network resources.

4.3 Stability evaluation in terms of communication quality

In this section, in confirmation of what we explained in Section **3** (redundancy encoding technology), we discuss how the stability of communication quality is affected by introduction of the prioritized packet discard scheme. Transmitting UDP packets with redundant data has an effect of not only eliminating retransmission processing time but also alleviating the effect of variation of packet loss ratio within a range of redundancy. On the other hand, as the redundant encoding technology assigns an error correction code before the content is divided for distributed storage, it is difficult to assign an increased code in the middle of the transmission process to meet the changes in network conditions. Therefore, redundancy of the error correction code should be set high assuming the worst-case situation in the network quality. This entails an excessive increase in network traffic.

To improve this situation, we employed the technology explained in Section **3** and evaluated a method that discards some of the packets selectively at the content-sending terminal according to the network traffic condition. This method aims at minimizing the amount of redundant packet transmission while maintaining transmission quality on the user side. This method classifies redundant data into a set of groups using the method described in Subsection **3.3**, and controls the number of groups (i.e. transmission volume) which are actually sent out according to the network traffic condition.

Using this method, we evaluated how the amount of redundant packet transmission affects the file restoration probability at the receiver terminal while the packet loss ratio in the network is deliberately fluctuated.

Specifically, we assumed the packet loss rate within the network falls within the range from 0.001% to 5%, and redundant packets are generated assuming 20% redundancy. Then, by varying the number of redundant packets actually sent out from the content-sending terminals within the range from 1/32 to 1/4, each corresponding file restoration probability was measured at the receiver terminal. The original file used in this experiment has a size of 10.24 MB, and 72 distributed content-sending terminals were used for sending data. Figure 9 shows the results of evaluation. Although the amount of deletable redundant data varies depending on the packet loss ratio within the network, the rate of successful file restoration can be 100% despite the fact that some of the redundant packets had been selectively discarded before transmission. This result clearly indicates the viability of total traffic reduction while maintaining communication quality for the receiving terminal. The file restoration ratio did not show significant fluctuation as long as the variation of the packet loss ratio within the network remained within a certain small range, indicating good stability of the scheme. From the experiment, it became apparent that we can determine the amount of discard once the upper limit of packet loss ratio is known. In practice, both reasonable traffic reduction and stable

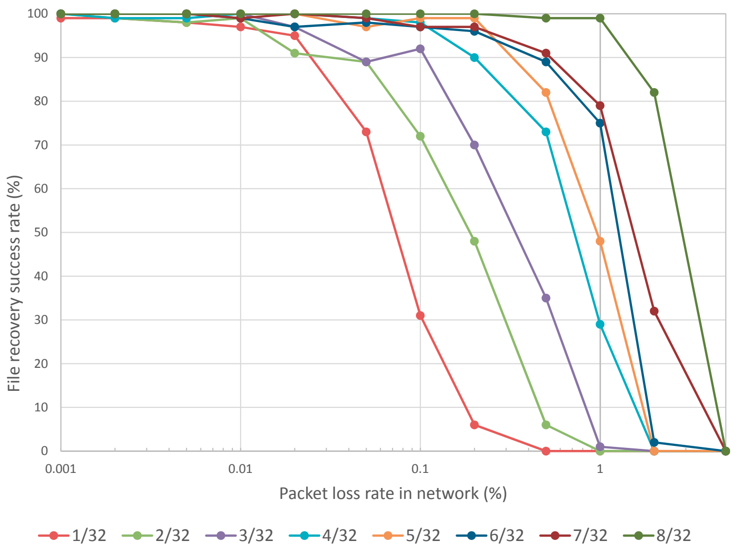


Fig. 9 File recovery success rate for 8 types of redundant data transmission size reduction (10 MB file with 20% redundant data)

communication quality can be achieved by referring to the packet loss ratio prescribed by the Service Level Agreement (SLA) as well as measured packet loss ratio.

As for the error correction at the receiving terminal, it took 14.0 ms in the case of 1/8 (12.5% redundant code transmission) where packet loss ratio within the network is 0.1%, while it took 0.4 ms when all the redundant packets were sent (100% transmission) (same packet loss ratio within the network: 0.1%). This clearly shows that the computational load on the receiving terminal increased. Therefore, stable communication quality with lower traffic requires additional computational resources at receiving terminals.

5 Conclusions

Toward development of a platform for efficient large volume content delivery, we have studied three technologies: multi-slice based traffic control technology, redundant encoding and transmission technology for globally distributed content, and transmission quality evaluation technology in a multi-slice environment. In this paper we presented detailed discussions on them.

The first (multi-slice based traffic control technology) aims at achieving flexible route modifications and traffic dispersion without resorting to complex route searching and path setup signaling procedures. Unlike conventional route control methods that perform routing and path control within a monolithic network, this technology employs a packet-by-packet route diversion exploiting multiple virtual networks with their own routing planes. We constructed the control mechanism on JGN-X network virtualization platform, and verified stable delivery of compressed 4K video data (approx. 170 Mbps).

The second (redundant encoding and transmission technology for globally distributed content) is an attempt to reduce traffic volume without compromising decoding characteristics on the receiving terminal. It uses LDGM coding—known to have a good affinity with video transmission applications—and takes advantage of the programmable nature of the VNode. In this technology, discard priority information is added to each chunk packet, and VNodes perform selective discard based on this information. Performance evaluation indicated that this selective discard scheme significantly improves decoding characteristics compared with conventional random drop even in the environment where higher occurrence (2.5 times) of packet losses is observed. We confirmed stable delivery of huge image data (16 Gbps: equivalent to 10 uncompressed HD videos) using 16 distributed sending terminals.

The third (transmission quality evaluation technology in a multi-slice environment) was an attempt to establish an effect/stability evaluation technique targeted at: a communication network that employs packet-by-packet route diversion, and, priority-based packet discard. For the former, this technology confirmed that OIF traffic does not interfere with OSF traffic, and OIF is capable of improving its throughput by utilizing alternative diverting routes. This enables highly effective use of network resources in content delivery. For the latter, we verified that the prioritized packet discard can attain a 100% file restoration ratio on the receiving terminal by performing selective discard when the packets are sent out from the sending terminals. This approach has a traffic reduction effect.

In the future, we will continue our research aiming at constructing an improved control strategy by combining the two technologies: the multi-slice based traffic control technology and the redundant encoding and transmission technology. Specifically, the inter-slice routing and the slice select/switching will not be necessary until the degradation of the received image quality becomes unable to be recovered even by using the error correcting function in a slice. Considering that the restrictions on bandwidth needed for the packet transmission should possibly be less when redundancy coding is used, it may be possible to reduce the frequency of slice control activation and to loosen the restriction on the bandwidth needed for detour slices. From the viewpoint of the slice control, considering these, we will study the way to determine the detailed timing of such control activation and selection criteria of alternative slices, and evaluate its outcome when used with redundant encoding.

From the viewpoint of the redundant encoding and transmission, further review will be needed in cases where the inter-slice routing and the slice select/switching uses a variety of slices that have different transmission characteristics such as packet loss ratio.

After achieving the integration of the element technologies, we then proceed to propose the outcome as a standard architecture for large-volume content delivery platforms, and to start discussions on its commercial viability. In the latter discussion, we expect to start our work with a thorough review of network virtualization infrastructures that meet the prerequisites for implementing our technologies, taking requirements for content delivery applications and levels of dissemination of related network virtualization technologies into consideration.

References

1 Kaneko Kunitake, et al. "Content Espresso: A Global Secure File Sharing System for Large Files," IEICE Technical Report, Vol.111, No.469, IN2011-192, pp.329–334, March 2011.

2 OPEN NETWORKING FOUNDATION, "OpenFlow Switch Specification Version 1.4.0 (Wire Protocol 0x05)," Oct. 14, 2013.

3 E. Rosen, et al., "Multiprotocol Label Switching Architecture," RFC3031, January 2001.

4 Nakao Akihiro, "Research on Network Virtualization for New Generation Networks," The Journal of the Institute of Electronics, Information and Communication Engineers,Vol.94, No.5, pp.385–390, 2011.

5 Network Testbed JGN-X, http://www.jgn.nict.go.jp

6 The Click Modular Router Project. http://www.read.cs.ucla.edu/click/

7 Ogura Tsuyoshi, et al. "Virtual network switching technology for flexible remote collaboration environment," IEICE Technical Report, NS2013-249, pp.419–426, March 2014.

8 Shirai, Kitamura, and Fujii, "4K Super High Definition Video Streaming System Using JPEG 2000 Codec," IEICE Technical Report, Vol.107, No.244, CS2007-29, pp.43–48, Sept. 2007.

9 Plank, James S., and Lihao Xu, "Optimizing Cauchy Reed-Solomon codes for fault-tolerant network storage applications," Network Computing and Applications, 2006. NCA 2006. Fifth IEEE International Symposium on Network Computing and Applications, IEEE, 2006.

10 Jin, Hui, Aamod Khandekar, and Robert McEliece, "Irregular repeat-accumulate codes," Proc. 2nd Int. Symp. Turbo codes and related topics. 2000.

11 Kitamura, Ogura, Kimiyama, and Fujii, "Practical Modeling on LDPC Decoding Duration," Proceeding of IEICE Society Conference, Vol.2013, No.2, p.151, 2013.

12 Ando Daisuke, et al. "Content espresso: A system for large file sharing using globally dispersed storage," Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on Cloud Computing and Technology and Science, Vol.2, IEEE, 2013.

13 Kitamura, Ogura, Kimiyama, and Fujii, "Basic Study on Distributed Streaming System with Multiple Sender using Foward Error Correction Codes for Large-sized Video Streaming," Technical Report of IEICE, CS, Vol.113, pp.69–74, 2013.

Tsuyoshi OGURA

Senior Research Engineer, Media Processing Systems Research Group, Media Innovation Laboratory, NTT Network Innovation Laboratories
Network Protocol, High-speed Storage Server, Network

Kunitake KANEKO, Ph.D.

Assistant Professor, Faculty of Science and Technology, Keio University
Application Oriented Networking, Network Architecture

Masahiko KITAMURA

Researcher, Media Processing Systems Research Group, Media Innovation Laboratory, NTT Network Innovation Laboratories
Distributed Parallel Processing, High-speed Real-time Processing, High-speed Real-time Transmission

Hiroyuki KIMIYAMA, Ph.D.

Senior Research Engineer, Media Processing Systems Research Group, Media Innovation Laboratory, NTT Network Innovation Laboratories
Distributed Parallel Processing, High-speed Real-time Processing, High-speed Real-time Transmission

Tatsuya FUJII, Ph.D.

Senior Research Engineer, Super Visor,
Group Leader, Media Processing Systems
Research Group, Media Innovation Laboratory,
NTT Network Innovation Laboratories
Video Communication, Video Codiing,
Digital Image Processing,
Media Tranport System

Atsushi TAKAHARA, Dr. Eng.

Director, NTT Network Innovation
Laboratories
Network Virutalizations, System Design
Method, Resillient System