Experiments of Multipath Multicast One-to-many Transfer with RS coding over Wide-Area OpenFlow Testbed Network

doi: https://doi.org/10.34385/proc.63.D4-1

Experiments of Multipath Multicast One-to-many Transfer with RS coding over Wide-Area OpenFlow Testbed Network

Masayuki KURATA[†a)], ***Nonmember***, **Masahiro SHIBATA**[†b)], ***and*** **Masato TSURU**[†c)], ***Members***

SUMMARY The importance of fast and efficient one-to-many transfers of a large file is increasing to replicate, move, or share bulk data not only intra-datacenter but also inter geographically distributed datacenters. We previously proposed the Coded Multipath Multicast (Coded-MPMC) one-to-many file transfer method in which the multicast transfer, the multipath transfer, and Reed-Solomon (RS) coding are integrated. This method aims to minimize the retrieval completion time of each recipient by simultaneously transmitting blocks of a file on multiple paths from a single sender to each recipient to maximize the aggregated flow value, i.e., to realize the max-flow to the recipient. We preliminarily implemented Coded-MPMC with OpenFlow protocol; however, we only tested its feasibility over a small homogeneous in-lab OpenFlow network. In this paper, through experiments on a wide-area OpenFlow testbed network, we show that Coded-MPMC correctly works in a heterogeneous and geographically-distributed network. The results suggest the practicability and potential benefits of Coded-MPMC in real networks.

key words: *OpenFlow, Multicast transfer, Multipath transfer, Max-Flow value, Reed-Solomon coding*

1. Introduction

The OpenFlow technology, as a practical realization of Software Defined Networking (SDN), provides the flexibility on controlling data flows in a centralized manner to deal with the increasing data traffic. The SDN is adopted to control traffic not only in a datacenter but also among geographically distributed datacenters across Wide Area Network (WAN), which refers to as SD-WAN, e.g., Google [1], Microsoft [2].

In transferring bulk data from a single sender to multiple recipients, i.e., one-to-many transfer, the use of unicast transfers will suffer from duplicate transmissions of the same data on specific links and/or the sender, resulting in inefficient use of network resources. On the other hand, the use of multicast transfers is more efficient because the data can be sent to multiple recipients in which the same data is transmitted on each link at most once. In response to the strong need for one-to-many file transfers using multicast transfers over SD-WAN, many efforts have been reported [3].

Based on our framework of the Multipath Multicast (**MPMC**) transfer model over a fully-controlled full-duplex network [4], we previously proposed **Coded-MPMC** which introduces Reed-Solomon (RS) coding at a sender. Through simulation on a wide range of large-scale network topologies, Coded-MPMC is shown to minimize the retrieval completion time (**RCT**) of a recipient, the duration from the time when the sender starts the file transfer until the recipient completes the file retrieval, for every recipient [5]. We have also implemented a prototype of Coded-MPMC; however, it is verified only on a small homogeneous in-lab OpenFlow network.

In this paper, we verify Coded-MPMC and its implementation in a more realistic network which is heterogeneous and geographically-distributed. We built a global OpenFlow testbed network consisting of our in-lab network and a wide-area OpenFlow testbed network "RISE" [6], which is extended to Seattle in the U.S. and conducted one-to-many file transfer experiments over that.

2. Overview of Coded-MPMC

Coded-MPMC aims to minimize RCT of each recipient R_X by retrieving the entire file using its own max-flow value M_X in one-to-many transfers. In this method, sender S divides a file into k equally-sized original blocks, where k is the least common multiple of all max-flow values and generates the necessary number of coded blocks using a systematic RS coding. In each phase, S transmits original or coded blocks to R_X in multiple paths whose aggregated link capacity is M_X (i.e., max-flow paths) using a multipath transfer, and at the same time, blocks duplicated by relay nodes are transmitted to other recipients to efficiently utilize link capacities using a multicast transfer. In an optimal transfer scheduling, each uncompleted recipient receives different blocks using its own max-flow value in each phase and retrieves the entire file in the lower-bound RCT as a result.

In Fig. 1, we illustrate an example of Coded-MPMC. Over a full-duplex network in which the link capacity between relay nodes is one and that between a relay node and a sender/recipient host is sufficiently large, S transfers a file to recipient R_C with the max-flow value of three and recipients R_A, R_B, and R_D with the max-flow value of two.

As shown in Fig. 1(a), S divides a file into six original blocks $\{1, 2, 3, 4, 5, 6\}$ because the least common multiple of all max-flow values is six, and generates one coded block $\{7\}$ using RS coding before block transmissions. Block sets $\{1, 2\}$, $\{3, 4\}$, and $\{5, 6\}$ are transmitted using three multicast flows each of whose flow value is one. Then, Coded-MPMC for the 1-st phase is optimal because each recipient R_X receives blocks using M_X multicast flows. Since R_A, R_B, and R_D do not receive six different blocks, Coded-MPMC proceeds to the 2-nd phase.

[†]The authors are Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, Kawazu, Iizuka-shi, Fukuoka, 680-4 Japan.
a) E-mail: kurata.masayuki318@mail.kyutech.jp
b) E-mail: shibata@cse.kyutech.ac.jp
c) E-mail: tsuru@cse.kyteuch.ac.jp

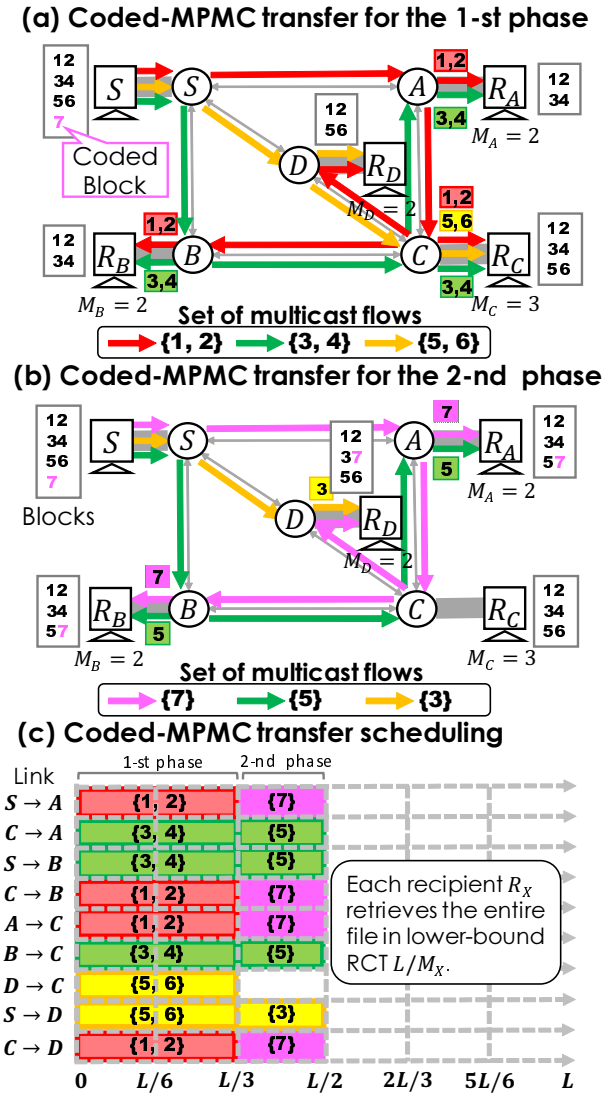

Fig. 1 Example of Coded-MPMC when sender S transfers original blocks "1", "2", "3", "4", "5", "6" and one coded block "7" to recipients R_A, R_B, R_C, and R_D

As shown in Fig. 1(b), block sets {7}, {5}, and {3} are transmitted using three multicast flows each of whose flow value is one. Then, Coded-MPMC for the 2-nd phase is also optimal because each uncompleted recipient R_X receives blocks using M_X multicast flows. Since all recipients receive six different blocks, Coded-MPMC is completed.

Figure 1(c) shows a time chart of Coded-MPMC scheduling. Each recipient R_X receives six different blocks to retrieve the file with size L in its lower-bound RCT L/M_X.

3. Implementation of Coded-MPMC

3.1 Procedure over OpenFlow Network

Our prototype of Coded-MPMC is implemented over an OpenFlow network consisting of Control Plane (C-Plane) and Data Plane (D-Plane). OpenFlow controller (OFC) application on C-Plane is developed using Ryu [7]. On the other hand, on D-plane, software switches (i.e., Open vSwitch [8] and Lagopus switch [9]) and hardware switches made by NEC Corporation are utilized as OpenFlow switches (OFSs). The applications including RS coding for sender/recipient hosts are realized by C++ programming language and the library [10] of RS coding.

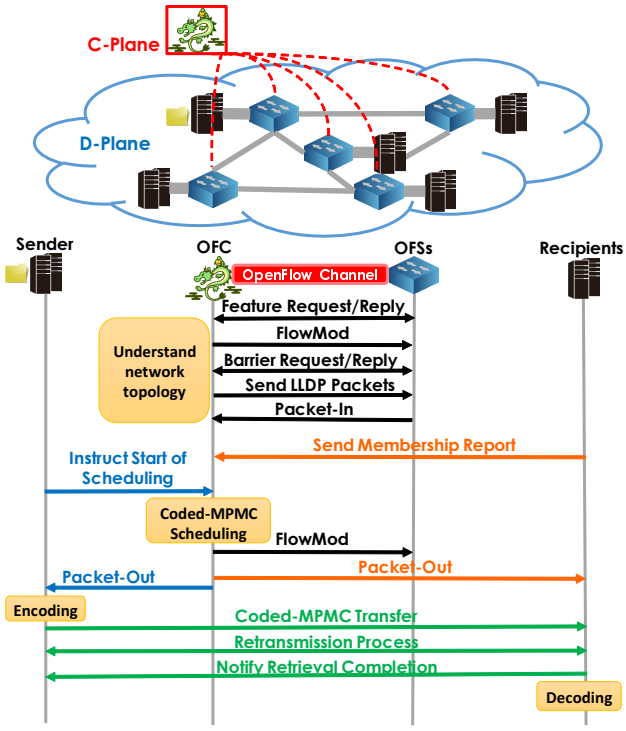

Fig. 2 Assumed network and procedures for Coded-MPMC with OpenFlow protocol

Figure 2 shows the procedures for Coded-MPMC over an OpenFlow network. First, OFC connects to all OFSs and creates an OpenFlow channel between OFC and each OFS. Next, to understand the network topology, OFC adds table-miss entries to OFSs and sends LLDP (Link Layer Discovery Protocol) packets after receiving Barrier Replies from OFSs. By receiving LLDP packets from OFS in Packet-In, OFC can obtain information on adjacent OFSs of its OFS.

Each recipient sends a membership report of Internet Group Management Protocol (IGMP) and notifies OFC of the recipient's network location. Then, the sender notifies OFC of the file size to transfer. With this notification as a trigger, OFC generates a schedule for Coded-MPMC. After scheduling, OFC adds flow entries to OFSs based on the generated schedule and notifies sender/recipient hosts of the file size and the information on the schedule.

The sender generates a necessary number of coded blocks and starts to transmit original or coded blocks. Finally, when the recipient completes receiving all packets of blocks to be retrieved, the recipient notifies the sender of its completion and starts decoding to retrieve the entire file.

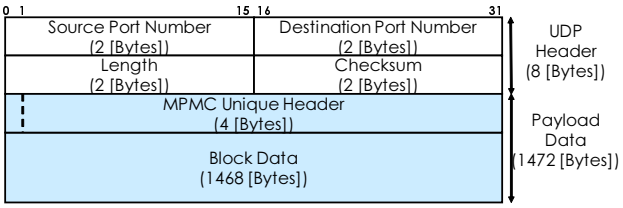

Fig. 3 UDP packet format for Coded-MPMC

3.2 UDP Packet Format

The transport layer protocol is User Datagram Protocol (UDP) to transfer the file using the multicast transfer and each equally-sized block is transmitted using UDP packets of format as shown in Fig. 3. Note that zero-padding is applied to create an equally-sized block if the file size is not a multiple of the number of original blocks.

A destination port number of the UDP header is determined from phase number α and block number β so that OFS can identify and route their packets. The port number is computed from "$5000+100\times\alpha+\beta$," e.g., its port number is "5207" when the block of "7" is transmitted for the 2-nd phase as shown in Fig. 1(b).

The payload data of the maximum size 1472 [Bytes] consists of MPMC unique header of size 4 [Bytes] and block data of size 1468 [Bytes]. The first 1 [bit] and the remaining 31 [bits] of its unique header indicate whether the UDP packet is the end of each block and where the block data is located in the entire file, respectively. In each phase, the sender transmits several kinds of blocks alternatively and starts to send in order from a packet with the first data in each block.

3.3 Retransmission Process

Since UDP does not guarantee the reachability of packet transfers, the application itself needs to deal with packet-loss. Therefore, in Coded-MPMC, OFC installs the unicast path for retransmission and the recipient performs a retransmission process with Negative ACKnowledgement (NACK) and packet retransmission timer.

After OFC completes scheduling for Coded-MPMC, OFC also computes the shortest path between a sender and each recipient. Then, OFC adds flow entries to realize retransmission processes using those paths. The packet retransmission timer of each recipient is started when receiving a packet for the first time and is reset every time a packet is newly received. When a packet of some block arrives in out-of-order within the block, the recipient recognizes that one or more packets are lost, and sends NACK to request retransmissions of those unreceived packets. When a timeout occurs (i.e., the retransmission timer exceeds a predefined timeout time), the recipient requests retransmissions of packets that have not been yet received. The retransmission process is repeated until the recipient completes to receive all packets of blocks to be retrieved.

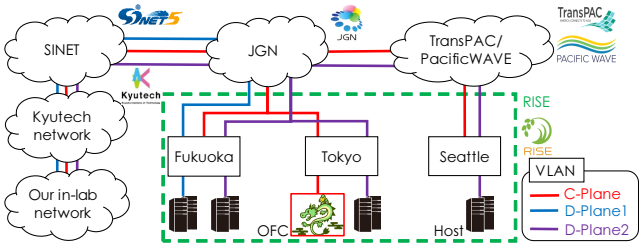

Fig. 4 Our implementing OpenFlow testbed network

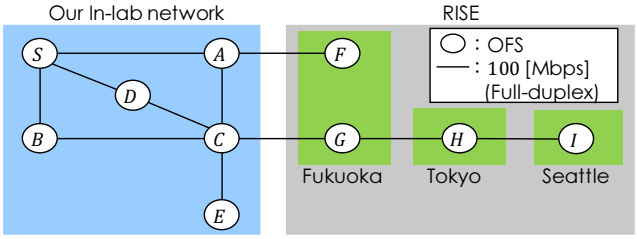

Fig. 5 Logical D-Plane network used in our experiments

4. Experiments on Testbed

4.1 Testbed Implementation

To verify Coded-MPMC from a sender to geographically distributed recipients, we have built an OpenFlow testbed network connecting our in-lab OpenFlow network and Fukuoka, Tokyo, and Seattle locations on a wide-area OpenFlow testbed "RISE" [6]. In our experiment environment as shown in Fig. 4, the in-lab network and RISE are connected by Kyutech campus LAN, Science Information NETwork (SINET), Japan Gigabit Network (JGN) domestic and internal network, and TransPAC/PacificWAVE transpacific line. To realize C-Plane and D-Plane networks over various substrates, L2-VPN using three types of VLAN is adopted.

Figure 5 shows the logical D-Plane network of Fig. 4. There are six OFSs in our in-lab network and four OFSs in RISE, with recipient host R_X connected to each OFS X, except for OFS S to which sender host S is connected; the link capacity between OFS and sender/recipient host sets to be 1000 [Mbps] to avoid the bottleneck. We assume that each link capacity between OFSs is 100 [Mbps]; therefore, there are recipients R_C with max-flow value of 300 [Mbps], R_A, R_B, and R_D with max-flow value of 200 [Mbps], and R_E, R_F, R_G, R_H, and R_I with max-flow value of 100 [Mbps]. Note that, by using the basic MPMC [4] without coding, at least one of recipients R_A, R_B, and R_D cannot retrieve in its lower-bound RCT; the coding in Coded-MPMC essentially benefits in this example.

4.2 Issues in Building Our Testbed

In building this testbed network, we met two issues on sending packets. First, IGMP snooping seems to be disabled in some L2-switch(es) between the in-lab network and JGN;

Table 1 Theoretical and experimental Retrieval Completion Time (RCT)

Recipient	R_A	R_B	R_C	R_D	R_E	R_F	R_G	R_H	R_I
Theoretical lower-bound RCT [s]	4.67	4.67	3.11	4.67	9.34	9.34	9.34	9.34	9.34
Experimental RCT [s] (**Without** Packet-loss)	4.71	4.71	3.14	4.71	9.50	9.50	9.50	9.50	9.50
Experimental RCT [s] (**With** Packet-loss)	4.71	4.71	3.14	4.71	9.50	10.21	10.21	10.22	10.32

therefore, L2 multicast packets cannot be transmitted between the in-lab network and RISE. We decided to use L2 broadcast packets to transfer block data instead of multicast packets. Second, LLDP packets also seem not to pass. We provide the OFC with the topology information between the in-lab network and RISE as static information. Note that those alternate measures do not affect the transfer performance of Coded-MPMC.

4.3 Installation of Proactive Flow Entries

RISE [6] is a layered structure consisting of an OpenFlow layer used by users (called U-land) and an OpenFlow layer used by operators (called K-land). OFS and OFC on K-land use reactive flow entries to authenticate packets transferred to OFS on U-land and to prevent some trouble from occurring. In our experiment, this specification makes data packets arriving at OFS on U-land to be sent to the OFS on K-land; then, their packets are sent to OFC on K-land by Packet-In. Therefore, for about one second between the start of Coded-MPMC and the installation of reactive flow entries, a lot of data packets are discarded due to Packet-In to OFC on K-land. Thus, we asked the operators of JGN to install proactive flow entries on OFS on K-land, and could confirm that such packet-loss does not occur.

4.4 Experiment Results

In this experiment, the sender transfers a file of size 104 [MBytes] in Coded-MPMC using each multicast flow with Constant Bit Rate (CBR) of 90 [Mbps]. As **the experimental RCT**, the duration from the time of receiving the first packet to the time of receiving the last packet (in all necessary packets) is measured by each recipient software for simplicity; therefore, the propagation delay from the sender to each recipient is excluded in the experimental RCT. In the worst case (R_I in Seattle), the difference is around 60 [ms] that is only 0.67 % from the results in Table 1. Note that the writing time of all data packets to the storage and the encoding/decoding time is excluded because they depend on the hardware and implementation.

Table 1 shows the experimental RCT of each recipient and its theoretical lower-bound RCT considering the file size, the CBR, and the MPMC unique header in the testbed of Fig. 4. Fortunately, two conditions can be seen in our testbed with and without packet-loss because it is constructed on shared testbed facilities in which the available link capacities may change. In the case with no packet-loss, the experimental RCTs are very close to the theoretical RCTs for all recipients. On the other hand, in the case that about thirty packets are discarded between our in-lab network and RISE, the experimental RCTs of R_F, R_G, R_H, and R_I become slightly larger because of retransmissions. Besides, we have confirmed that each recipient successfully retrieved the transferred file by decoding in all experiments. These results suggest that our implementation of Coded-MPMC including retransmission processes correctly works.

5. Conclusion

For fast and efficient one-to-many file transfers over a full-duplex OpenFlow network, Coded-MPMC can realize an optimal schedule in which each recipient retrieves the entire file in its lower-bound time duration. In this paper, we have presented an implementation of Coded-MPMC and the one-to-many file transfer experiments conducted over a wide-area OpenFlow testbed network connecting our in-lab OpenFlow network and RISE [6]. Through experimental results, we can conclude the practicability and potential benefits of Coded-MPMC and its implementation in real networks.

We are grateful to those who are involved in network operations in JGN/RISE by NICT, SINET by NII, and the Kyutech LAN, as well as Dr. Nagata and Mr. Okamoto in iD Corporation for their help in conducting the testbed experiment. This work was supported by the Resilient Edge Cloud Designed Network (19304), NICT.

References

[1] S. Jain, A. Kumar, *et al.*, "B4: experience with a globally-deployed software defined wan," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3—14, Aug. 2013.
[2] S. Kandula, I. Menache, *et al.*, "Calendaring for wide area networks," *Proceedings of the 2014 ACM conference on SIGCOMM*, vol. 44, no. 4, pp. 515–526, Aug. 2014.
[3] S. Islam, N. Muslim, and J. W. Atwood, "A Survey on Multicasting in Software-Defined Networking," *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, p. 355–387, Nov. 2017.
[4] A. Nagata, Y. Tsukiji, and M. Tsuru, "Delivering a File by Multipath-Multicast on OpenFlow Networks," *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, pp. 835–840, Sep. 2013.
[5] M. Kurata, K. Heira, M. Shibata, and M.Tsuru, "Minimizing One-to-Many File Transfer Times using Multipath-Multicast with Reed-Solomon Coding," *2019 31st International Teletraffic Congress (ITC 31)*, pp. 115–116, Aug. 2019.
[6] Y. Kanaumi, S. Saito *et al.*, "RISE: A wide-area hybrid OpenFlow network testbed," *IEICE transactions on communications*, vol. 96, no. 1, pp. 108–118, 2013.
[7] "Ryu SDN Framework," [Online]. Available: https://ryu-sdn.org/, Accessed on: Sep. 30, 2020.
[8] "Open vSwitch," [Online]. Available: https://www.openvswitch.org/, Accessed on: Sep. 30, 2020.
[9] "Lagopus switch and router," [Online]. Available: https://www.lagopus.org/, Accessed on: Sep. 30, 2020.
[10] "Longhair," [Online]. Available: https://github.com/catid/longhair, Accessed on: Sep. 30, 2020.