Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 31, 2016

T. Zhu
F. Wang
D. Feng
Q. Shi
Y. Xie
Huazhong University of Science and Technology

September 28, 2015

Congestion-Aware and Robust MultiCast TCP in Software-Defined Networks
draft-tingwei-mctcp-00

Abstract

Reliable group communication is required in distributed applications, such as distributed file systems (HDFS, GFS and Ceph), where such group communication is defined by the sender and the group members are small (e.g. three). However, existing standards for reliable multicast transport are receiver-initiated and suffer from inefficiency in either host-side protocols or multicast routing.

This draft proposes a sender-initiated, efficient, congestion-aware and robust reliable multicast solution in Software-Defined Networks (SDN), called MCTCP (MultiCast TCP). The main idea behind MCTCP is to manage the multicast groups in a centralized manner, and reactively schedule multicast flows to active and low-utilized links, and it is implemented by extending TCP as the host-side protocol and managing multicast groups in SDN-controller.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 31, 2016.

Table of Contents

1. Introduction

Traditional reliable multicast schemes are mainly designed for very large multicast groups, such as PGM [RFC3208], NORM [RFC5740]. They are not suitable for the sender-defined small group multicast scenarios mainly for the following reasons.

1. Traditional reliable multicast schemes are receiver-initiated application-layer protocols (based on UDP), which suffer from high software overhead on end hosts and mismatch to the sender-initiated mode.

2. Traditional IP multicast routing algorithms, such as PIM-SM [RFC4601], are not designed to build optimal routing trees. They are not aware of link congestion, and thus apt to cause significant performance degradation in burst and unpredictable traffic environment.

3. Traditional multicast group management protocols, such as IGMP [RFC3376],MLD [RFC2710], are not aware of link failures. Any failure in multicast spanning trees can suspend transmission and lead to significant performance loss or business interruption.

The emergence of SDN (Software-Defined Networking), brings new ideas for solving routing efficiency issues of reliable multicast in data centers. A centralized control plane called SDN-controller provides global visibility of the network, rather than localized switch level visibility in traditional IP networks. Therefore, multicast routing algorithms can leverage topology information and link utilization to build optimal (near-optimal) routing trees, and be robust against link congestion and failures.

This memo proposes an SDN-based sender-initiated, efficient, congestion-aware and robust reliable multicast scheme, called MCTCP, which mainly designed for small groups. The main idea behind MCTCP is to manage the multicast groups in a centralized manner, and reactively schedule multicast flows to active and low-utilized links. Therefore, the multicast routing can be efficient and robust. To eliminate the high overhead on end hosts and achieve reliability, MCTCP extends TCP as the host-side protocol, which is a transport-layer protocol.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Software Defined Networking (SDN): defined in RFC 7426 [RFC7426]

Sender: A sender is a node which can start up a multicast communication and send data to several other nodes.

Receiver: A receiver can only wait for connection from a sender and receive data. It does not need to subscribe a multicast group in advance, but just keep listening for connection.

Multicast Session: A multicast session contains a sender and several receivers.

MST: Multicast Spanning Tree.

HSP: Host Side Protocol.

MGM: Multicast Group Manager.

3. Motivation

MCTCP is designed for the sender-defined small group scenarios, which are very common in distributed systems like distributed file systems. MCTCP can make full use of the advantage of the SDN technology, and provide a framework for other intelligent or user-defined functions.

Compared to traditional reliable multicast schemes, MCTCP has the following advantages:

1. Easier programming in upper-level applications. MCTCP provides common socket APIs, so that a programmer can use MCTCP easily.

2. More efficient. MCTCP can process the packets more efficient in host-side protocol (Transport-Layer), and can forward the packets via more efficient multicast spanning trees.

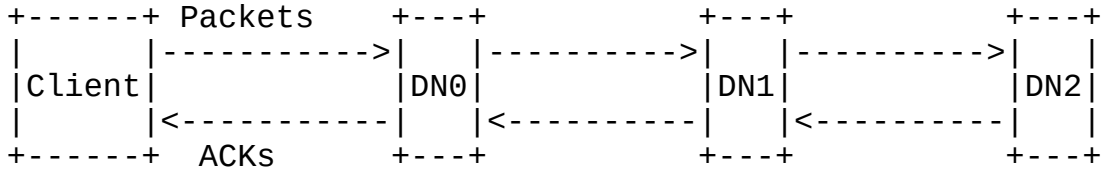
3. More robust. MCTCP can be aware of link failures, so the loss caused by a link failure decreases greatly.

4. More secure. MCTCP is inherent secure as the sender keeps the information of all receivers. Moreover, the centralized admission control in MGM helps achieve security.

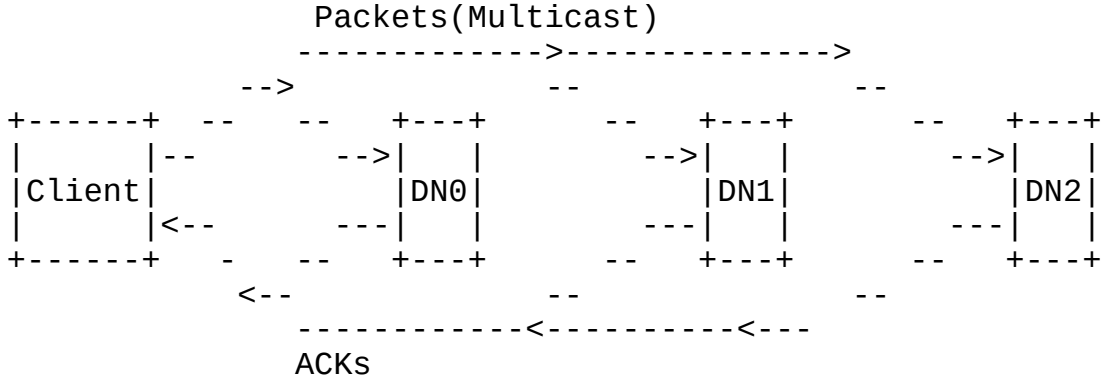
5. Require no assistance from network devices. Different from the network-equipment scheme Xcast [RFC5058], which supports a very large number of small multicast sessions by explicitly encoding the list of destinations in the data packets, MCTCP can deploy on common SDN-enabled network devices and need no assistance from network devices.

4. An Example Application

The HDFS data replication process is a typical one-to-many data transmission, during which the client gets the list of DNs (Data Nodes) from a NN (Name Node), and then delivers the data chunks to them. We assume the replication factor is three.



(a) Pipeline-based data replication



(b) Multicast-based data replication

Illustration of Pipeline-based and Multicast-based data replication.

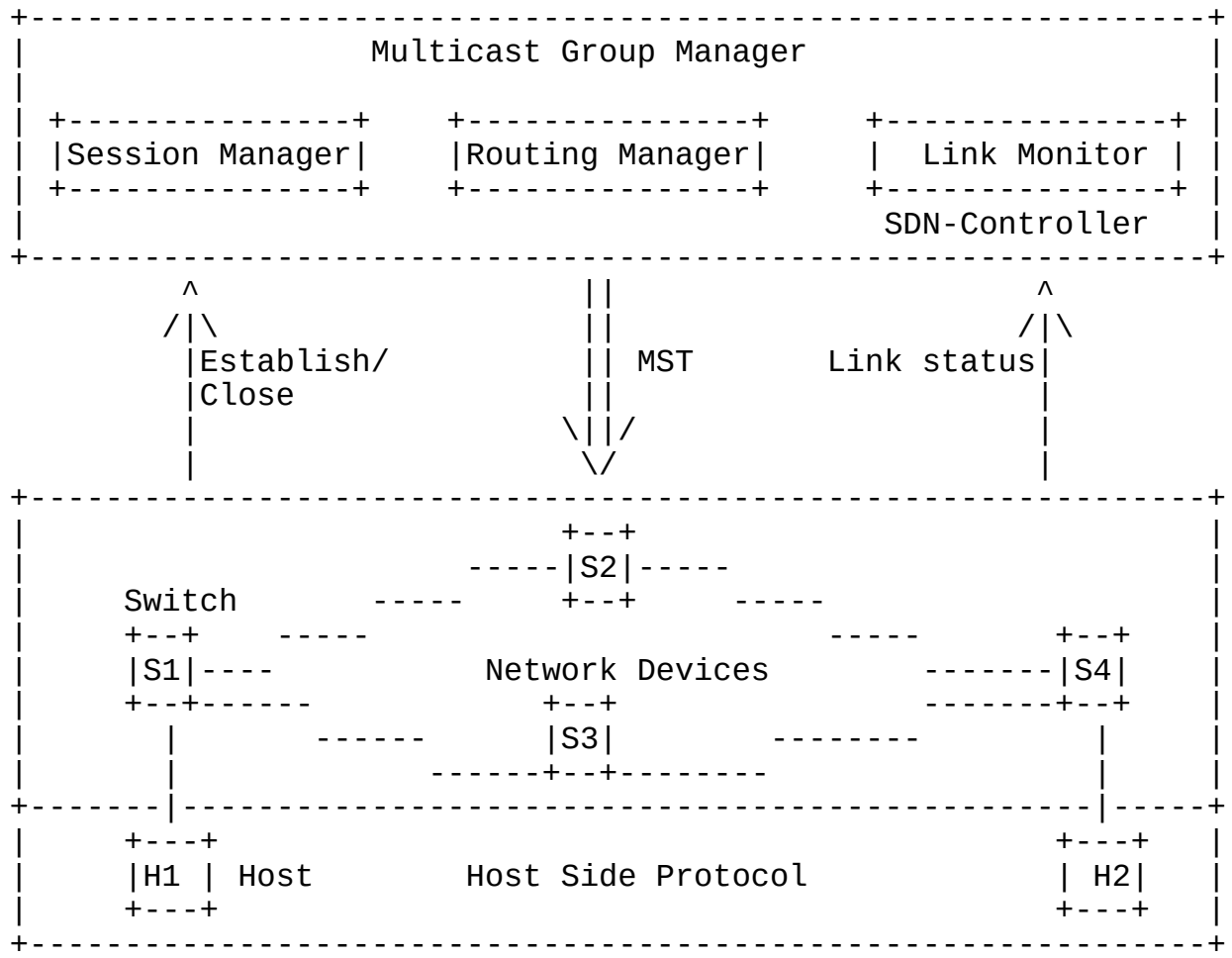
Figure 1

As shown in Figure 1(a), the original HDFS employs a pipeline-based replication method. The data transmission unit is a packet, which is usually 64KB. For each packet, the client first transfers it to DN0; then the DN0 stores and passes it to DN1; finally the DN1 stores and transfers it to DN2. After the DN2 receives the packet, it returns an acknowledgment to DN1; then the DN1 returns an acknowledgment to DN0; finally the DN0 returns an acknowledgment to the Client. Therefore, the whole process can be regarded as a six-stage pipeline. Let O-HDFS denote the original HDFS. O-HDFS has 2*n stages when configured as n replicas, resulting in long delay in packet transmission. In addition, O-HDFS delivers data in unicast, which will generate a large number of duplicated packets into the network and reduce the overall transmission performance.

Let M-HDFS denote the multicast-based HDFS, which using MCTCP for data replication. As shown in Figure 1(b), the client divides the data into packets, and then delivers them to three data nodes DN0, DN1, DN2 in multicast. For each packet, the client transfers it to DN0, DN1, DN2 simultaneously using MCTCP, and then all the data nodes return acknowledgements to the client directly. Therefore, M-HDFS's data replication procedure can be regarded as a two-stage pipeline. Compared with O-HDFS, M-HDFS has less stages (two stages to six stages), thus resulting in lower latency. Meanwhile, M-HDFS delivers data in multicast, so redundant packets in network are reduced greatly.

5. MCTCP Architecture

MCTCP consists of two modules, the HSP (Host-Side Protocol) and the MGM (Multicast Group Manager). The HSP is an extension of TCP, leveraging the three-way handshake connection mechanism, cumulative acknowledge mechanism, data retransmission mechanism and congestion control mechanism to achieve reliable multipoint data delivery. The MGM, located in SDN-controller, is responsible for calculating, adjusting and maintaining the MSTs for each multicast session. It keeps monitoring the network status (e.g. link congestion and link failures) and creates maximal possibility to avoid network congestion and to be robust against link failures.



The architecture of MCTCP.

Figure 2

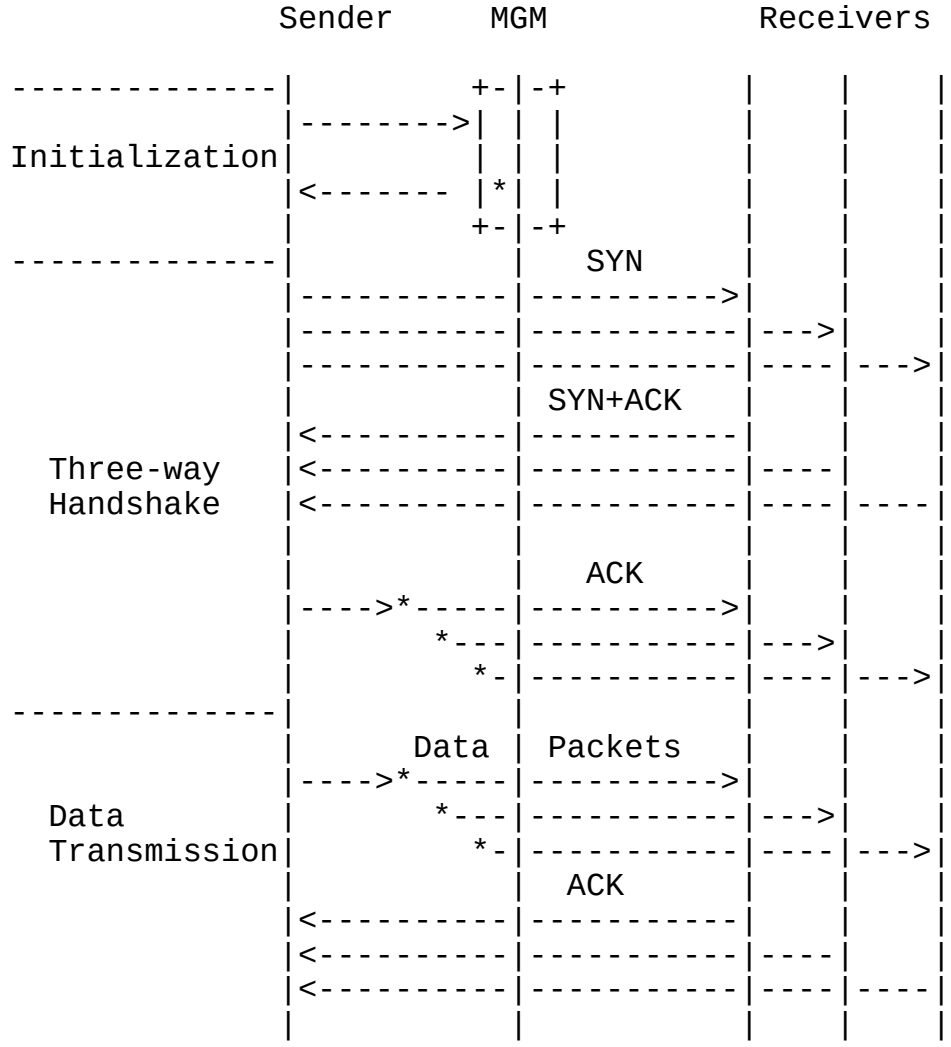
The sender establishes connection with multiple receivers explicitly before data transmission. First, the sender requests to the MGM for calculating the MST. Second, the MGM calculates and installs the MST. Third, the sender starts three-way handshake with receivers, and begins data transmission after that. Fourth, the MGM adjusts the MST once link congestion or failure detected. Fifth, the sender notifies the MGM after data transmission finishes.

5.1. Host Side Protocol

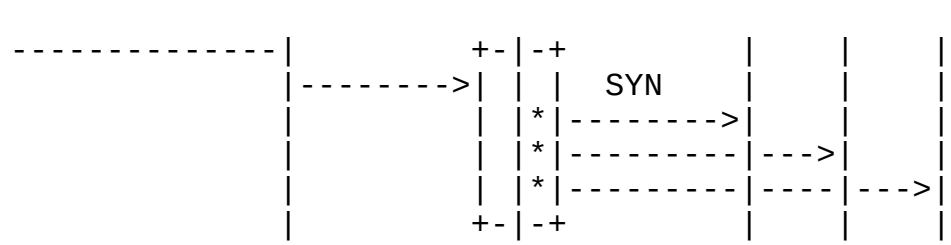
5.1.1. Session Establishment

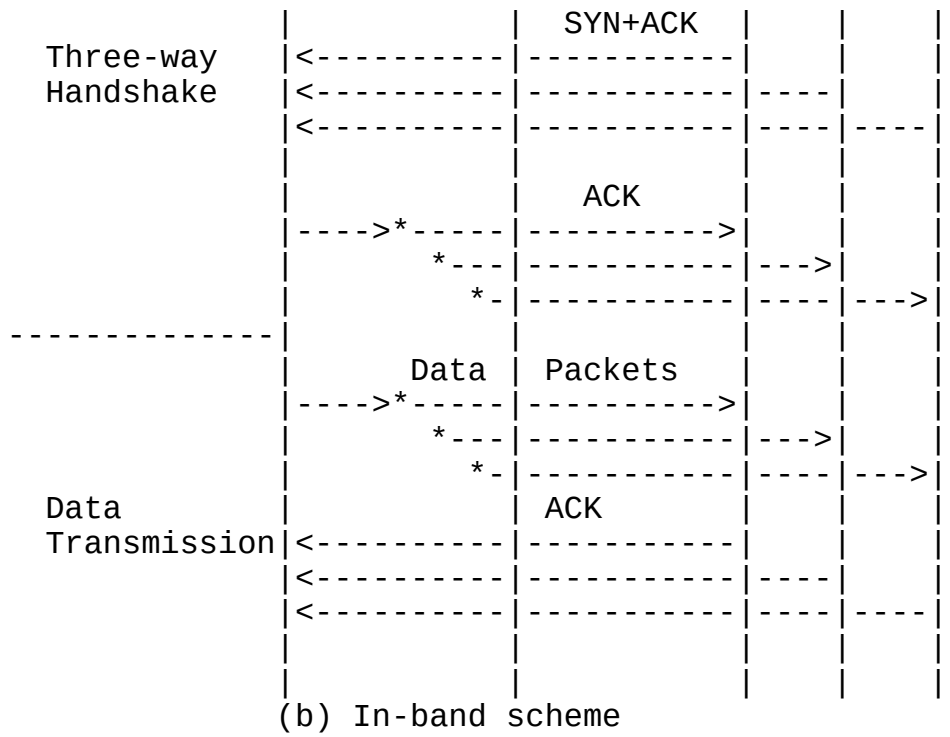
The sender requests to MGM for calculating MST when establishing a new session. Since the receivers do not obtain the multicast address in advance, the first handshake must be realized by using unicast

address. The multicast address is placed in the SYN packet. After receiving the SYN packet, the receivers get the specific multicast address, and join the group (just put the multicast address into the interested list, but not send IGMP messages), so that they can receive the multicast messages.



(a) Out-band scheme





The Procedure of Session Establishment and Data transmission. There are three receivers.

Figure 3

There are two alternative schemes, the out-band and the in-band schemes. For the out-band scheme, the sender requests to the MGM before three-way handshake. After calculating the MST, the MGM notifies the sender to start three-way handshake. For the in-band scheme, the SYN packet is used to request MST for calculation, and redirected to the MGM. After receiving the SYN packet and calculating MST, the MGM dispatches the SYN packet to all the receivers in unicast. Figure 3 illustrates the procedure of connection establishment and data transmission.

The out-band scheme suffers from time overhead of an extra RTT to the SDN controller. Hence, this scheme is suitable for the large amount data transmission scenes, in which the overhead of session establishment is negligible. The in-band scheme has no extra time overhead, but brings much pressure on the SDN controller. This scheme is more suitable for extremely small membership and delay-sensitive scenes.

5.1.2. Data Transmission

When a session is established, data transmission begins.

Packet Acknowledgement. The sender maintains a sliding window and processes the acknowledgement from receivers. The send window advancement is decided by the slowest receiver. As MCTCP is mainly designed for small group scenarios, the ACK-implosion problem in traditional large member reliable multicast is negligible.

Packet Retransmission. The sender does multicast retransmission when the timer expires or a packet loss is detected. Since the efficient and robust multicast forwarding achieved by MGM can greatly reduce the packet loss, the emergence of retransmission in MCTCP will be largely decreased.

Congestion Control. A large amount of congestion control algorithms can be used in MCTCP, such as TFMCC [RFC4654], pgmcc [PGMCC].

Node failure. A receiver is considered as failed if the sender does not receive any acknowledgement from it within a threshold time. The failed receiver, which may encounter crash or network failure, should be cleaned out from the multicast session in order to ensure the transmission of the rest receivers. Therefore, the upper-level applications should be responsible for fault recovery.

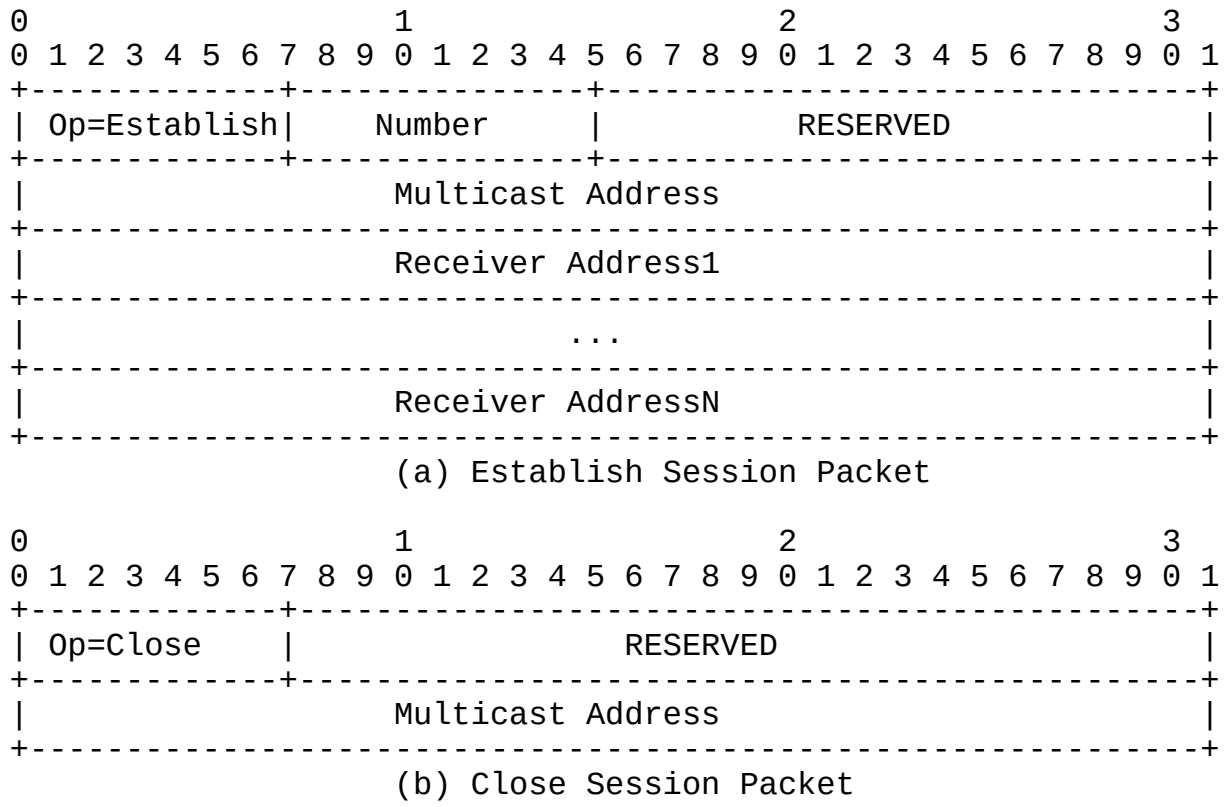
5.1.3. Session Close

After data transmission is completed, the sender closes the multicast session initiatively, and then notifies the MGM.

5.1.4. Packet Format

There are two kinds of packets in MCTCP, the control packets and the data packets. The Control Packets are used to maintain the session states. The Data Packets are the regular packets.

5.1.4.1. Control Packet



Control packet

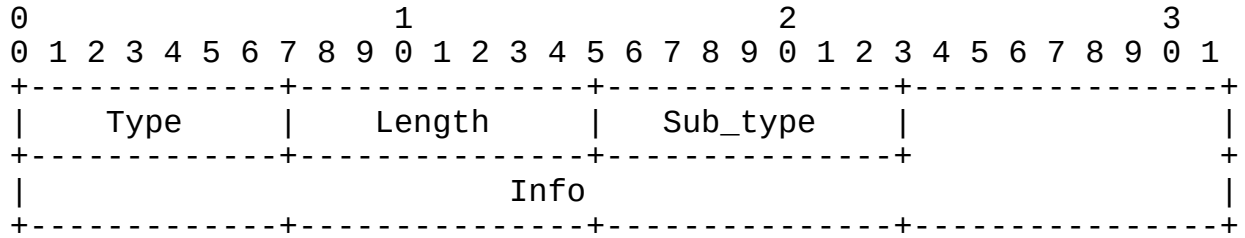
Figure 4

There are at least two control packets:

- SessionEstablish packet: If a sender wants to start a multicast session, it MUST assign a multicast address and a set of receivers, then send them to the MGM for MST calculation. As shown in Figure 4, the packet MUST contain the multicast address, the receiver number and the address of each receivers.

- SessionClose packet: When a multicast session is closed, the sender MUST tell the MGM.

5.1.4.2. Data Packet



Options Field in MCTCP Data Packets

Figure 5

MCTCP is a new transport-layer protocol. To simplify the complexity of implementation and ensure compatibility of the protocol, the packet format of MCTCP is the same as TCP, and the related features for MCTCP are implemented in TCP options. The options field of MCTCP are depicted in Figure 5, where the 'type' is the option type, namely defined by TCPOPT_MCTCP, the 'sub_type' is the sub-options of MCTCP, INCLUDING OPTION_MCTCP_XID, OPTION_MCTCP_MCADDR, OPTION_MCTCP_SENDER, etc. The 'info' is the contents for corresponding sub-types.

- OPTION_MCTCP_XID, used to identify a unique group, with length of 7 bytes, 4 bytes for XID. The XID is generated by the sender, delivered to receivers during connection establishment for identifying the multicast session, defined as the shared initial sequence number of all receivers as well.

- OPTION_MCTCP_MCADDR, used to deliver the current multicast address, with length of 8 bytes, 1 byte for receiver ID and 4 bytes for multicast address. This option is used in the SYN packet for delivering the multicast address to receivers, and in all packets which the receivers send to the sender for identifying which group the packets belong to. The receiver ID identifies which receiver the packet comes from, and is set NULL in the SYN packet.

- OPTION_MCTCP_SENDER, used to identify whether the packet is sent out by the sender, with the length of 3 bytes. In the sender, the five-tuples <multicast address, source port, destination address, destination port, protocol> is used to identify a session instead of <source address, source port, destination address, destination port, protocol>. Therefore it is different from the receivers in processing a receiving packet in the sender. The host has to know whether a packet is coming from a sender or a receiver before processing it.

5.1.5. Programming APIs

The HSP uses the common socket APIs for programming. When programming using MCTCP, the receivers call the listen() system call for listening, just the same as TCP. At the sender, the user can specify a multicast address for the multicast session, otherwise a random multicast address will be allocated automatically by the HSP. Then the sender should call setsockopt() function to specify the address list of the receivers before connect(), as shown below.

```
#define PEER_NUM 3
struct sockaddr_mc{
  uint16_t sin_port;
  struct in addr_sin addr;
}
struct sockaddr_mc mc_addr[PEER_NUM];
setsockopt(fd, IPPROTO_MCTCP, MCTCP_ADDR, mc_addr, sizeof(mc_addr));
```

5.2. Multicast Group Manager

MCTCP uses a logically centralized approach to manage multicast groups. The MGM, located in SDN controller, manages the multicast sessions and MSTs. By keeping the global view of the network topology and monitoring the link status in real-time, the MGM can adjust the MSTs in case of link congestion or failures. Specifically, the MGM consists of three sub-modules, the session manager, the link monitor and the routing manager, as shown in Figure 2.

5.2.1. Session Manager

The session manager is responsible for maintaining the states of all groups. When establishing or closing a multicast session, the sender informs the session manager. Hence, the session manager can keep track of all the active multicast sessions. If a multicast session is closed, the MST will not be cleared immediately, but just be marked inactive. Therefore, a session with the same sender and receivers can reuse the MST. The session manager periodically cleans up the inactive MSTs.

5.2.2. Link Monitor

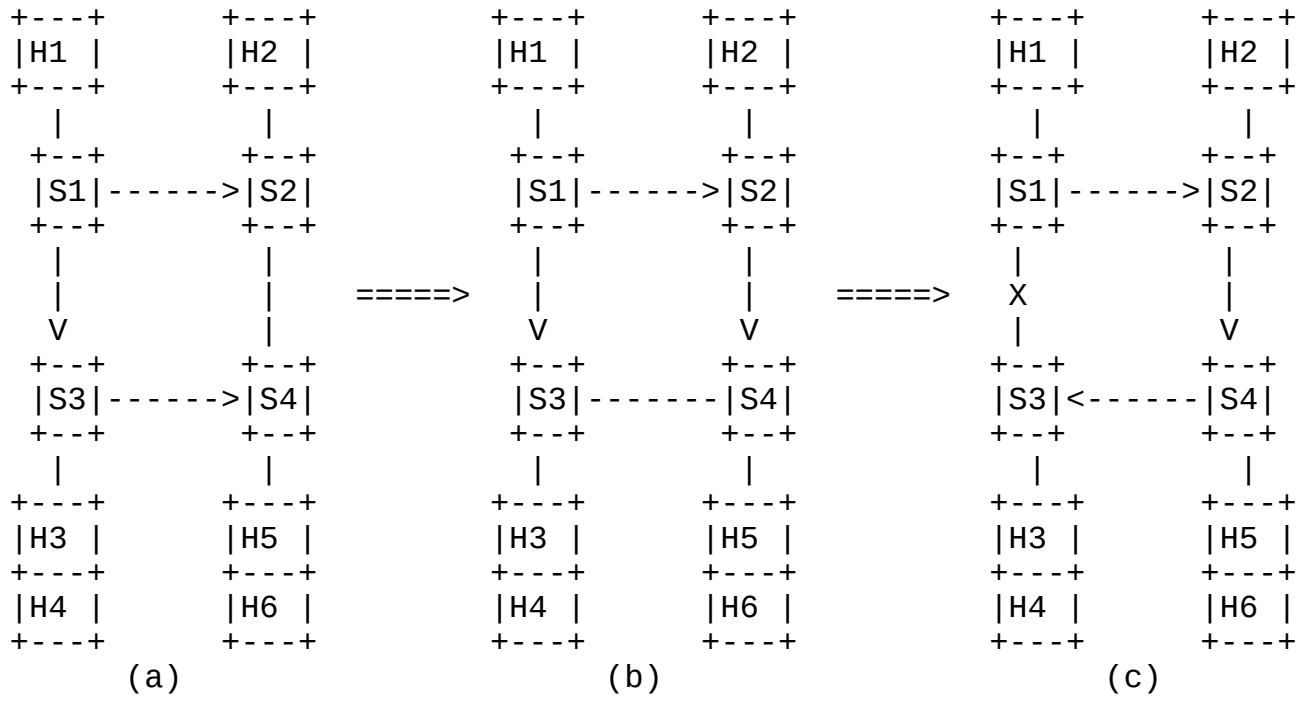
Link Monitor is responsible for monitoring network link status, and estimating the weight of each link periodically. This can be achieved easily by sFlow, NetFlow or "port-status" interface in OpenFlow [OpenFlow] protocol.

5.2.3. Routing Manager

The routing manager is responsible for calculating and adjusting MSTs. When establishing a new multicast session, the routing manager calculates the minimum cost MST based on the current link utilization. When a link overloads or failure occurs, the adjustment for all MSTs over the link will be triggered. The routing manager is divided into two parts, the routing calculation and the routing adjustment. The MST should be calculated quickly during session establishment. In the case of link congestion, the MST should be adjusted in the best-effort way. When a link fails, all the relevant MST should be quickly updated.

- Routing calculation. The members of a group are assigned by the sender, and no dynamically join/leave is allowed in MCTCP once the session begins. So a lot of static multicast routing algorithm can be used, the minimum-cost path heuristic algorithm (MPH)[MPH], for example. The MPH algorithm inputs a set of sender/receiver nodes and all-pairs shortest paths which are calculated by Floyd-Warshall algorithm, and outputs a minimum cost MST.

- Routing adjustment. When the link monitor detects link overloading, i.e. the link weight is larger than a preset threshold, the routing adjustment will be triggered. In routing adjustment, the relevant MSTs should be recalculated.



An example for MST adjustment. A multicast group G1:H1-->{H2,H3,H5} in (a). Then H4 starts sending data to H6 with TCP in (b). A link down between S1 and S3 happens in (c).

Figure 6

For example, consider a simple network topology which consists of four switches, as shown in Figure 6.

At time T0, there is one group G1:H1->{H2,H3,H5}, and the current MST is MST1:{S1->S2, S1->S3, S3->S4}.

At time T1, H4 start to send data to H6, causing plenty of TCP traffic on link S3->S4, resulting in confliction with group G1 at link S3->S4. Once the link monitor detects the congestion, the routing manager will start to adjust the MST of G1. So the new MST will be MST2:{S1->S2, S1->S3, S2->S4}.

At time T2, link S1->S3 fails. Then the routing manager will adjust the MST of G1 to MST3:{S1->S2, S2->S4, S4->S3}.

6. Security Considerations

MCTCP is more secure than traditional reliable multicast schemes, mainly for the following two reasons.

First, MCTCP is a sender-defined scheme, all the receivers are specified by the sender. Therefore, eavesdroppers can not join or leave a multicast session freely. It is hard to steal data from a multicast session.

Second, all the multicast sessions are under control of the MGM, so it is easy to enable admission control and policy enforcement. For example, the MGM can enable authentication for each senders and receivers, so that a malicious sender is hard to start up a multicast session. Some forms of denial-of-service attack which wants to enlarge by using multicast can be prevent.

7. IANA Considerations

TBD

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, DOI 10.17487/RFC2710, October 1999, <http://www.rfc-editor.org/info/rfc2710>.

[RFC3208] Speakman, T., Crowcroft, J., Gemmell, J., Farinacci, D., Lin, S., Leshchiner, D., Luby, M., Montgomery, T., Rizzo, L., Tweedly, A., Bhaskar, N., Edmonstone, R., Sumanasekera, R., and L. Vicisano, "PGM Reliable Transport Protocol Specification", RFC 3208, DOI 10.17487/RFC3208, December 2001, <http://www.rfc-editor.org/info/rfc3208>.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <http://www.rfc-editor.org/info/rfc3376>.

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, DOI 10.17487/RFC4601, August 2006, <http://www.rfc-editor.org/info/rfc4601>.

[RFC4654] Widmer, J. and M. Handley, "TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification", RFC 4654, DOI 10.17487/RFC4654, August 2006, <http://www.rfc-editor.org/info/rfc4654>.

[RFC5058] Boivie, R., Feldman, N., Imai, Y., Livens, W., and D. Ooms, "Explicit Multicast (Xcast) Concepts and Options", RFC 5058, DOI 10.17487/RFC5058, November 2007, <http://www.rfc-editor.org/info/rfc5058>.

[RFC5740] Adamson, B., Bormann, C., Handley, M., and J. Macker, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol", RFC 5740, DOI 10.17487/RFC5740, November 2009, <http://www.rfc-editor.org/info/rfc5740>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <http://www.rfc-editor.org/info/rfc7426>.

8.2. Informative references

[MPH] Takahashi, H. and A. Matsuyama, "An approximate solution for the steiner problem in graphs", Math. Japonica, vol. 24, no. 6, pp. 575-577, April 1980.

[OpenFlow] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., and L. Peterson, "OpenFlow: Enabling Innovation in Campus Networks", ACM SIGCOMM, vol. 38, no. 2, pp. 69-74, April 2008.

[PGMCC] Rizzo, L., "Pgmcc: A TCP-friendly Single-rate Multicast Congestion Control Scheme", ACM SIGCOMM, p: 17--28, October 2000, <http://doi.acm.org/10.1145/347057.347390>.

Authors' Addresses

Tingwei Zhu
Huazhong University of Science and Technology
WuHan 430074
P.R.China

Email: twzh@hust.edu.cn

Fang Wang
Huazhong University of Science and Technology
WuHan 430074
P.R.China

Email: wangfang@mail.hust.edu.cn

Dan Feng
Huazhong University of Science and Technology
WuHan 430074
P.R.China

Email: dfeng@hust.edu.cn

Qingyu Shi
Huazhong University of Science and Technology
WuHan 430074
P.R.China

Email: qingyushi@hust.edu.cn

Yanwen Xie
Huazhong University of Science and Technology
WuHan 430074
P.R.China

Email: ywxie@hust.edu.cn