Protocol Behaviors for Multipeer Multimedia Communications

Yong-Woon Kim

Protocol Engineering Center, ETRI

E-mail : qkim@pec.etri.re.kr

Abstract

This paper proposes an enhanced transport protocol for multipeer communications. It is assumed that there exists a transport connection owner that takes the roles of the establishment, management and termination of a transport connection. The proposed protocol classifies the data transfer type into simplex, duplex and N-plex multicasts and provides several transport services to support various requirements in group communications. The general operations and reliability controls of each transfer type are different from one another and carried out by a shared control tree. The QoS negotiation is performed during the creation phase. The notification of negotiated values is followed by an acknowledgment procedure for confirmation. The four-way handshake is introduced. After negotiation, such a resource reservation protocol as RSVP can reserve system and network resources according to the arbitrated values. This paper suggests a conceptual model of the transport layer and its protocol behaviors over the IP multicast and RSVP network.

I. Introduction

Interactive/distributed multimedia applications require a reliable concurrent multicast service based on transmitting user data from a single or multiple sources to all members of a multicast group, such that: a) every packet from each source is delivered to each receiver within a finite time, free of errors, with no duplicates, and in the order sent by the source; and b) nodes responsible for retransmitting packets, can delete packets from memory within a finite time.[1] Such applications often involve a large number of members and interactive in nature with members dynamically joining and leaving the applications. These applications typically require a specific delivery form of multicast, that is, 1xN communication in which there are a single sender and multiple receivers.[2]

The goal is to exploit the highly efficient QoS supported delivery mechanisms over the IP Multicast, in order to construct a scalable and efficient multicast transport protocol called ECTP (Enhanced Communications Transport Protocol). It provides the following features:

- Required QoS values can be arbitrated during the connection creation phase. RSVP can use the negotiated values for resource reservation.
- Receiver-driven reliability control is carried out for scalability. Receivers do use a combination of periodic and eventual ACKs and restricted NAKs by the slotting and damping algorithm. In addition, the tree structure is exploited to restrict the scope of retransmissions to the region where packet loss occurs.
- Transient QoS or AGI violations may suspend the data transfer state. It can be resumed when the problems are recovered.

In section II, assumed network services and related works carried out before are explained. An extension of the domain name system, DNS, is introduced, which is a kind of a directory service to translate a name address to an IP address. In section III, the transport layer model of ECTP is described with a figure. In section IV, the transport services provided are presented and their protocol behaviors are described. Conclusions and directions for future work are offered in section V.

II. Background and Assumed Network Services

To meet the growing demand for reliable multicast, many reliable multicast protocols have been proposed. The issues raised by reliable multicast are considerably more complex than those related to reliable unicast. The following five items have to be considered seriously in reliable multicast protocols.

First, scalability should be provided. It means to be independent of the number of participants which may extend tens to hundreds, thousands, or more on a multipeer connection. Reliable multicast on the scale of tens of hundreds of participants scattered across the Internet requires carefully designed flow and error control algorithms that avoid the many potential bottlenecks. They include host processing capacity and network resources.[2,3]

Second, an efficient congestion avoidance algorithm should be applied. In general, multicast applications have the potential to cause more congestion-related damage to the network than do unicast applications.[5] The following algorithms have been exploited to cope with the congestion problem: window-based slow start, rate-based control, or both.

Third, error control has to be considered carefully. It is more complex in reliable multicast than reliable unicast. In RMP, a rotating token site only can receive NAKs from troubled member and can do selective repeat. A single ring can produce a scalability problem in a large group. Another problem is that the token is not passed to the next member of the ring of receivers until the new site has correctly received all packets that the former site has received.[1] In a tree structure, a parent and his children may be formed into a local group on which error control can be bounded, but some interactions with higher layers can be taken for control. Any other member in a local group can retransmit packets or the parent only can do.

Forth, QoS should be considered for multimedia applications. Different types of traffic have different QoS requirements and end nodes have different system and network resources for supporting the QoS requirements. In order to keep a specific level of quality, negotiation and reservation have to proceed among participants.

Fifth, multicast transport protocols have to consider a membership control. Group membership characteristics describe the way in which multicast groups are defined, established, and maintained. Two distinctions can be recognized: member identity & membership boundary.

Table 1 is for comparison about additional issues of proposed reliable multicast protocols so far.

III. The Transport Layer Model

Figure 1 depicts a transmission model between the transport layer and the network layer. It is assumed that a pair of a multicast address and a global port is known before beginning of multicast sessions. SDP and SAP can be used to announce session information including the pair. Application entities use a globally unique port and a multicast group address that may be expanded at routers.

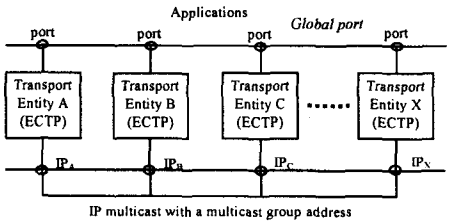

Figure 1 - Model of the transport layer over the network multicasting service

ECTP distinguishes a service data unit and a protocol data unit, so it can support the concept of ALF[6] by limiting reliability control within a service data unit.

Connection Owner

The basic transmission component in ECTP is a transport connection established among a pair of client users or more users. There is the only owner client user and its transport entity at a multicast group connection. The user with ownership can be able to trigger a connection establishment and the transport entity takes the role of the establishment, management, and termination of the connection.

Types of Transport Connection

A transport connection designed at ECTP may be classified into three transfer types. *Simplex*: the owner entity may send only and all others may receive only. This style of transmission can be used as the basic technology of push in WWW or multicast FTP.

Duplex: the owner entity can both send to and receive from all others whereas all other entities can receive only from and send only to the entity. *N-plex*: any entity is a sender as well as a receiver, that is, anyone can send something and all others may receive it, at any moment.

IV. ECTP Behaviors

An ECTP entity does provide various multipeer communication services: connection creation, data transfer, pause, resume, report, join, leave, and termination.

IV-1. Connection Establishment

Such key functions of a multicast transport protocol are carried out during the connection creation phase, as the AGI condition check, and QoS negotiation. The active group integrity herein called AGI specifies conditions on the member population, such as quorum number, minimum number of members, mandatory members, and so on.

Simplex Connection

The owner client requests a connection creation, i.e., a socket function call in other words, containing predefined or proposed values of connection characteristics, such as QoS parameters, AGI conditions, etc., which may be degraded by its transport entity in lack of available local resources. The owner entity sends a connection request packet in multicast to peer transport entities and these entities deliver an indication signal to their clients.

The client users may select the proposed values or adjust them higher or lower by negotiation rules and then respond to their entities. The selected or modified values may be degraded at each entity. Every participating entity sends a response packet in return to the source. Then the owner transport entity gets all information about participants in the end and check the AGI conditions. It may trigger the major arbitration of QoS only in the successful conditions and transmit a confirmation packet containing determined characteristics, which has to be followed by acknowledgments of peer entities. It may guarantee the explicit transmission and, therefore, the three-way handshake exchange should be expanded into the four-way handshake in the multipeer group communication.

Duplex Connection

The difference of duplex from simplex is to add another data transmission channel from each peer entity to the owner entity. Since a single QoS in a connection is allowed, every transmission channel has to have the same QoS values. The connection setup procedures are exactly the same with the simplex case because at the owner entity the negotiated parameter values are identical to all the return channels and can be sent in multicast.

N-plex Connection

The key feature of N-plex is to permit simultaneous multicast data transmissions among the whole active members. The difference of N-plex from duplex is that the returning unicast channel from a peer entity may be expanded into the multicast channel to other active entities. ECTP allows a single QoS in a single N-plex connection. The connection setup procedures are the same with the simplex or duplex case, but the QoS negotiation rules may be more complex than those of other types and the processing load may be increased.

IV-2. Data Transfer

The data transfer provides for an exchange of user data of an application. Flow and error controls take place for the reliable data transfer according to the required QoS values. A sending protocol entity splits a service data unit of an application user appropriately into multiple protocol data units and receiving entities reassemble them into the original service data unit, and then deliver it to application users. For efficiency ECTP uses the packet sequence number instead of the byte sequence number.

Simplex Data Transfer

The owner entity may send a single protocol data unit or multiple consecutive units to peer transport entities. Receivers respond with an

acknowledgment only to the last data packet containing the mark of end of service data unit, not to the intermediate ones. They send ACKs to their parent in unicast. Every parent can reply with ACKs even though ACKs of their childrens have not been arrived yet. If reliability control is limited within a block of consecutive packets segmented from a single service data unit, the ALF concept can be supported.

If any errors are detected, the recovery process starts. Prompt NAK has to be sent in multicast after a random delay that is controlled by the well-known slotting and damping algorithm. When a receiver obtains a NAK for a packet that it has not received and for which it has started a timer to send a NAK, the receiver sets a timer and behaves as if it had sent a NAK. With this scheme, it is hoped that only a single NAK is sent back to the source. So receivers use a unicast address to their parent in ACK transmissions and do a multicast address in NAK transmissions.

When peer transport entities have received a block of packets including the last packet containing the end mark, they reassemble collected packets into an original service data unit and deliver it to their client users.

As much as possible the window space is available, a sender may transmit consecutive packets segmented from a few service data units before it receives ACKs for the corresponding service data units. If the window comes empty, it stops to send data packets and waits for some credit advertised by receivers. This scheme is called the sliding window mechanism for avoidance of the end system saturation. The initial window size can be negotiated during the connection setup time to fix a problem of the slow start algorithm. Additionally the maximum transmission rate can be set for avoidance of the network congestion.

Duplex Data Transfer

The basic mechanism is similar to that of the simplex type. Each return path only to the owner entity is added to the multicast transmission path of the simplex type. So general operations are the same.

Since each return path is towards the owner entity, a system crash may be caused by packet concentration on it. MMCP limits the number of concurrent channels and uses the token-based control mechanism. In the duplex transfer type, the owner entity has a specific number of tokens and allocates them dynamically. Fairness and priority should be considered.

N-plex Data Transfer

In N-plex every packet has to be sent in multicast from a sending entity to all other receiving entities, in other words, it may be said that all the transport entities have their own simplex-style transmission path. So the basic control mechanisms are the same with the duplex and simplex cases. The token-based control also should be carried out to avoid excessive concurrent senders.

IV-3. Pause/Resume/Report

These services have to be invoked only by the owner entity that monitors the status of connection properties and calles appropriate protocol functions.

The pause is used to suspend the data transfer state temporarily in the case of QoS or AGI violations. A join or leave action will cause a membership change and then trigger the AGI condition check. If an AGI violation occurs in the soft AGI policy, the owner entity uses this protocol function. In order to suspend the transfer state, the entity sends a pause packet in multicast to peer entities and indicates it to its client user. Every other entity has to reply with an ACK and indicates the news to their client users. The ACKs are transmitted towards the source.

The resume service is used to resume the suspended connection when the violated properties are recovered or a certain problem is released. If the owner entity decides to resume the transfer state while monitoring, it sends a resume packet in multicast. Corresponding ACKs are required.

The report service is used to notify some warning messages or changes of connection characteristics to participating entities. When some characteristics are changed against the threshold but not below the level of minimum or the membership is changed by a join or leave, the owner entity gives the warning or the notification to peer entities. A report packet is sent in multicast and corresponding ACKs are required. Every client user will be notified by its transport entity.

IV-4. Join/Leave

The join service is used for a user to join an existing connection. A late-joining member has to accept the existing data transfer type and connection properties. A trying entity may use a unicast or multicast address. While joining a connection, it sends a join request packet in multicast to all other entities or in unicast to the owner entity if known. The packet contains the characteristic constraints under which the joining entity expects to join an existing connection. The owner entity decides whether the join request shall be accepted or not. Re-negotiation of the characteristics is not allowed. If it is accepted, the entity sends a join confirmation packet back to the join initiator and a report packet to active transport entities. These packets require corresponding ACKs.

The leave service is used to remove a user from an existing connection group. This service may be initiated:

- by a client user to leave a connection itself
- by a client user to deny to take part in a creating connection
- by the owner transport entity to exclude a troublesome entity
- by a transport entity to reject a join request of a participating user
- by a transport entity to reject a creation request of the owner

There are five cases related to the leave operations and they can be classified into three types. The first one is the case that a client user itself leaves an existing connection, the second one is the case that the owner entity kicks out other troublesome entities and the last one is the case that a transport entity rejects its user request. The last case does not require a packet transmission.

A leave initiator sends a leave request packet to the owner entity or to a target transport entity for exclusion. The initiator may be one of peer entities. Each leave request packet requires an ACK. Then the owner entity sends a report packet to notify the membership change.

IV-5. Termination

Only the owner client user or its transport entity may use this function to release an existing connection or to quit an ongoing operation of a connection creation. The entity sends a termination request packet in multicast. Each receiving entity should reply with an ACK and issue an indication message to its client user.

V. Conclusion

This paper has proposed a transport model for a reliable multicast transport protocol called ECTP and its supported transport services. The protocol requires some network facilities such as multicasting, resource reservation and directory service. The assumed network services have been presented in view of research and development.

The protocol has been designed for three types of data transfer: simplex, duplex and N-plex, which may be selective according to applications. Several transport services described beforehand and those transfer types can be very applicable to diverse applications.

The implementation is underway on FreeBSD 2.2.0 with IPv6 that has been almost completed. These results will be contributed to ISO/IEC JTC1/SC6 WG7 ECTP project having a goal of designing an enhanced transport protocol.

References

[1] Brian Neil Levine, David B. Lavo and J.J. Garcia- Luna-Aceves, "The Case For Reliable Concurrent Multicasting Using Shared Ack Trees", Proc. ACM Multimedia, 1996

[2] Rajendra Yavatkar, James Griffioen, and Madhu Sudan, "A Reliable Dissemination Protocol for Interactive Collaborative Applications", Proc. ACM Multimedia, 1995.

[3] Sridhar Pingali, Don Towsley, and James F. Kurose, "A comparison of sender-initiated and receiver-initiated reliable multicast protocols", Proceedings Of ACM SIGMETRICS 94, Vol. 14, pp. 221-230, 1994

[4] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", IEEE/ACM Transactions on Networking, Nov. 1996

[5] Allison Mankin and Allyn Romanow, "IETF Criteria For Evaluating Reliable Multicast

Transport and Application Protocols", Internet Draft

[6] D. Clark and D. Tennenhouse, D., "Architectural considerations for a new generation of protocols", Proc. of ACM SIGCOMM 90, Sept. 1990, pp. 201-208

	Control Model	Transfer Type	Join / Leave	Error Signal	NAK & ACK avoidance	Congestion avoidance / Flow control
MTP	token-based master	1xN, NxN	Yes	unicast NAK		rate-based
RMP	token-based ring		Yes	multicast NAK		window-based slow start
AFDP		1xN, 1xALL	Yes	multicast NAK via TCP	slotting/damping NAK	rate-based
RMTP-AT&T	group hierarchy		Yes	NAK	periodic ACK	window-based rate control & slow start
SRM	local recovery group		No	NAK	periodic ACK	
TMTP	tree-based	1xN, Nx1, MxN	Yes	multicast NAK	TTL-based NAK, a single combined & periodic ACK	window-based fixed rate control
MMCP	tree-based	1xN, NxN, duplex	Yes	multicast NAK	slotting/damping NAK, periodic/eventual ACK	window-based variable rate control

Table 1 - Comparison of Reliable Multicast Transport Protocols