Realization of a Scalable and Reliable Multicast Transport Protocol for Many-to-Many Sessions

Seungik Lee, Yangwoo Ko, and Dongman Lee

In this paper, we present the Enhanced Communication Transport Protocol–Part 5 (ECTP-5), which provides scalable and reliable multicast communication service for many-to-many applications by constructing high quality recovery trees from two-layer logical trees and repairing the losses via unicast automatic repeat request–based error control. In order to realize the protocol, we developed feasible protocol architectures and building blocks including additional functions which deal with engineering details, such as membership dynamics and sender coordination. Experimental results show that ECTP-5 scales well with various session sizes and packet loss rates in terms of control overhead and recovery latency.

Keywords: Reliable multicast transport protocol, many-to-many session, ECTP-5.

I. Introduction

The proliferation of the Internet has removed geographical barriers and enabled various types of multi-user interactions on a global scale. With the increasing number of participants in a session, multicast is becoming essential in multi-party applications. Reliability is one of the key issues in multicast communication. In order to support reliable transport in multicast communication, several approaches have been introduced [1]-[7]. These approaches have been standardized by the IETF, ITU-T, and ISO/IEC [7]-[10], where detailed architectures have been specified and implemented.

While there have been successful studies on reliable multicast, they focus only on one-to-many applications, such as file transfer and TV broadcast. With the high interest in collaborative work among distributed users, however, many-to-many applications, in which a participant receives data from others and sends data to others, are widely used on the Internet. Examples of many-to-many applications include video conferencing, networked multiplayer games, and distributed data caches. Some previous studies and standardized protocol suites argue that they also can deal with the case of multiple senders. However, it was shown in [11] that they suffer from scalability problems in terms of session throughput and tree maintenance overhead when applied to a many-to-many session. Group-Aided Multicast (GAM) [12] was developed to address the scalability problem. By using group concepts, GAM provides two different types of logical trees for inter- and intra-group members to balance the quality of the recovery tree with its maintenance overhead. Control packets, such as negative acknowledgment (NACK) and REPAIR, are delivered along the recovery tree via unicast to avoid throughput degradation due to high processing loads at receivers. While this approach has been proven by numerical analyses and simulations

Manuscript received May 21, 2007; revised Sept. 19, 2007.

This work was supported in part by the MIC & IITA, Rep. of Korea under the ITRC support program (IITA-2007-C1090-0014) and IT Leading R&D Support Project; by ETRI under research project (4010-2007-0028).

Seungik Lee (phone: +82 42 866 6220, email: silee@icu.ac.kr), Yangwoo Ko (email: newcat@icu.ac.kr), and Dongman Lee (phone: +82 42 866 6113, email: dlee@icu.ac.kr) are with School of Engineering, Information and Communications University, Daejeon, Rep. of Korea.

to be scalable in many-to-many sessions, there has not been any feasibility test deploying it on the Internet. It does not consider engineering details, such as dynamic membership, quality of recovery trees, and coordination of senders, which are essential for a successful realization of a reliable multicast protocol.

In this paper, we present Enhanced Communication Transport Protocol–part 5 (ECTP-5), which is a new standard reliable multicast protocol for many-to-many sessions [13]. We demonstrate how it addresses the scalability problems previously mentioned by describing its detailed rationale, a design overview, and experimental results. While the basic rationale of its design originates from that of [12], ECTP-5 is newly designed as a complete protocol suite which includes refined error recovery and secure support of dynamic changes in a session, a recovery tree, and membership. It uses a tree-based automatic repeat request (ARQ) approach with the following salient features: high quality recovery tree construction while keeping the maintenance overhead of the trees reasonably low using a two-layer hierarchy of the participants; scalable error control, which addresses the feedback implosion and retransmission exposure problem; session and tree maintenance mechanisms, which are robust against membership dynamics and host failure; sender identification, which is used to coordinate the multicast senders according to specific applications; and logical tree adaptation, which adjusts the recovery tree to improve its quality.

To demonstrate its feasibility as a standard protocol, we developed a prototype system which consists of modular and reusable protocol building blocks and compared its performance with those of NACK-Oriented Reliable Multicast (NORM) [8] and Asynchronous Layered Coding (ALC) [7], which are well known reliable multicast protocol standards from the IETF. From intensive experiments over Emulab [14] with different session sizes and packet loss rates, we can see that ECTP-5 can be used on the Internet in a scalable manner with comparable performance to the existing standards in terms of control overhead and recovery latency.

The remainder of this paper is organized as follows: In section II, other standardization activities for reliable multicast transport are summarized. In section III, we outline the design considerations in building a many-to-many reliable multicast protocol architecture and its procedures. The protocol architecture and procedures are briefly described in section IV. We discuss ECTP-5 implementation details in section V and the evaluation results in section VI. Finally, the conclusion follows in section VII.

II. Related Work

There are currently several standardization activities for reliable multicast transport. In this area, the IETF RMT WG [15], ITU-T Q.1/17, and ISO/IEC JTC1/SC6/WG7 are known to be the most active groups. The IETF RMT WG focuses on the standardization tracks for NORM [8] and ALC [7]. The NORM protocol is a NACK-based reliable multicast transport protocol instantiation. It provides reliable delivery by repairing lost packets with forward error correction (FEC) or by retransmitting the data from the sender. The ALC protocol uses layer-coded data delivery and FEC [16] to provide a reliable and rate-controlled stream service, though it does not guarantee completely reliable delivery because it does not use any feedback from receivers. Both NORM and ALC have been published as IETF experimental requests for comment (RFCs).

The ITU-T Q.1/17 and JTC1/SC6/WG7 have jointly developed the ECTP series [9], [10], [17]-[19]. The ECTP series covers the distributions of multicast data to multiple receivers. It is categorized into six parts according to the communication type of target applications. ECTP-1, ECTP-3, and ECTP-5 focus on one-to-many, some-to-many, and many-to-many communications, respectively. They commonly provide reliable data delivery with a tree-based ARQ approach, though they function differently when requesting or repairing lost packets and when constructing recovery trees. The ECTP-1 protocol has been published as an ISO/IEC International Standard (IS) and an ITU-T Recommendation, while ECTP-3 and ECTP-5 have been approved as ITU-T Recommendations. The ECTP-2, ECTP-4, and ECTP-6 protocols focus on quality of service (QoS) management of ECTP-1, ECTP-3, and ECTP-5 respectively.

The NORM protocol provides reliable multicast transport of bulk data or streams by selective NACKs and retransmission of data between one or more senders (even though it was originally designed for single-sender sessions) and the receivers. If a receiver detects one or more packet losses, it multicasts (or unicasts in some cases) a NACK packet to the entire multicast group. Since this operation can cause feedback implosion, NORM uses a probabilistic NACK suppression mechanism with a random back-off timer. Even with this suppression mechanism, however, the scalability of NORM is limited to small or medium-sized groups [20].

The ALC content delivery protocol deals with heterogeneous receivers. It divides content into multiple layers and transmits them via several multicast group addresses. Each layer is encoded by FEC so that packet losses can be repaired on the receiver side without feedback. Therefore, ALC scales well and can be used in a unidirectional communication environment, such as with satellite communications. Since it has no feedback from receivers, however, it does not guarantee complete reliability of data when the redundant data for FEC coding is insufficient. The FEC coding/decoding overheads are another drawback [21].

The ECTP-1 protocol targets one-to-many multicast services.

It provides session management and membership management by a sender and tree-based reliable delivery. The recovery tree along which the control packets, such as ACK and REPAIR, are sent is constructed at session initialization. Periodically, ACK packets are sent to the parent node with piggybacked loss information of a receiver. Upon reception of ACK packets from child nodes, a parent node sends the aggregated ACK information to its parent node. It also retransmits the requested lost packets to its children via a local multicast channel. The ECTP-3 protocol is very similar to ECTP-1 in terms of reliability support, but it permits multiple senders in a session. It additionally provides backward channels from each sender to the root of the recovery tree. Using a backward channel, the root of the recovery tree forwards the control packets to corresponding senders. Both of them use a tree-based ARQ approach and scale better than NORM [8]. However, because they use periodic ACK packets to request retransmission of lost packets, the recovery latency depends on the acknowledgement interval. The use of a local multicast channel at each parent-child relationship also limits usage for large-scale sessions due to difficulties in managing the number of multicast channels.

III. Design Considerations

1. Scalable Reliability Support

As the scale of the network increases in terms of the number of participants and geographic span, reliable multicast protocols face two intrinsic challenges: *feedback implosion* and *exposure to retransmission* [22]. These challenges have been previously considered in many studies, and tree-based approaches with local recovery along a recovery tree have been proposed. However, these approaches do not consider the case in which there are many sources, which may result in throughput degradation due to the substantially larger processing load of control packets on receivers [11].

As well as the processing overhead of the participants, the quality of a recovery tree is another important factor to consider. Tree-based reliable multicast protocols perform best when a recovery tree is congruent to its corresponding multicast routing tree [23]. A separate recovery tree for each sender as in RMTP [2] could be roughly congruent with its corresponding routing tree so that the best protocol behavior can be achieved. However, the tree maintenance overheads, which increase linearly with the number of senders, would not be acceptable when there are many senders. In order to eliminate the need to maintain one recovery tree per source, Lorax [5] constructs and maintains a single recovery tree for a many-to-many group. However, it suffers from poor approximation if the underlying multicast routing protocols provide per-source routing trees, such as DVMRP, MOSPF, and PIM-DM. These two examples show that there is a fundamental trade-off between the quality of recovery trees and the tree maintenance overheads unless multicast routing uses a globally shared routing tree.

In order to address these scalability problems, ECTP-5 exploits a combined group/tree approach using unicast feedbacks [12]. The per-source tree approach and the shared tree approach are only two extreme ends, and there is a spectrum of policies that subsumes the two ends. By introducing a group concept, ECTP-5 finds the sweet spot in which the quality of recovery trees can be balanced with tree maintenance overheads. Participants in an ECTP-5 session form multiple groups, namely local groups, according to their topological locality. A representative node of a local group is called the local owner (LO), which is responsible for membership management of the group. Using this hierarchy of members, ECTP-5 builds a two-layer logical tree which consists of shared intra-group logical trees and inter-group logical trees. The recovery tree for a sender can be derived by grafting these trees.

2. Dynamic Membership and Fault Tolerance Support

In a group communication session, the participants can dynamically join or leave during the session. Dynamic membership is one of the key considerations in tree-based reliable multicast transport protocols because it affects recovery-tree maintenance and session management. When a participant leaves the session without reconstruction of the recovery tree, its child nodes cannot recover from packet losses; thus, the protocol fails to support reliability. Moreover, the protocol should be designed to survive any type of failure due to dynamism in distributed end hosts and networks. For example, the protocol must work even when the control packets are lost or an end host does not respond properly due to a host failure.

For dynamic membership support in ECTP-5, a participant or a dedicated server, namely, the transport-connection owner (TCN), maintains session membership through explicit join and leave control messages and validates it with periodic probe messages. When leaving a session, a participant should also leave the intra-group logical tree by informing its parent to keep the logical tree valid. If the participant has one or more child nodes, it tells the child nodes to attach to the participant's parent node prior to leaving the tree. Node failure can be detected by the lack of response to control packets, such as repair and repair request packets.

3. Sender Identification

A participant may need to get explicit permission from a coordinator to speak to other participants for floor control in conferencing tools. Thus, the number of senders needs to be limited, or only the authorized senders should be allowed to send multicast data to the session depending on applications.

In order to manage the authorized senders, a participant sends a request to become a sender to a centralized server; and the server accepts or declines the request according to application configuration. The other participants should also be able to verify whether the received data packets are from the authorized senders by using information obtained from the centralized server. The ECTP-5 protocol uses *tokens* for sender identification. A sender can send multicast data to the session members with its own token given by the TCN, and the receivers validate incoming data using the token list, which is periodically updated by the TCN. Note that ECTP-5's sender identification is not designed as a security measure against security threats such as fraudulent senders.

4. Logical Tree Adaptation

To improve loss recovery efficiency, the recovery trees should be able to adapt as closely as possible to a multicast routing tree, which can be estimated by comparing the packet loss patterns of receivers. Through comparison of packet loss patterns, a parent and its child in a recovery tree can determine whether their parent-child relation is appropriate.

In [12], the delivery status of data packets is recorded by local group members and the representative nodes of local-groups (namely, the *cores*) for adapting intra-group and inter-group logical trees, respectively. However, if a core node is not a sender or its data sending rate is not high enough to be used for the comparisons, intra-group logical tree adaptation does not work. Moreover, if data packets from all the sources are used to adapt inter-group logical trees, the overhead from recording packet delivery status becomes too huge to make the protocol feasible.

In order to solve these problems, ECTP-5 adjusts only intra-group logical trees. For intra-group logical tree adaptation, a representative node of each local group, namely the LO, periodically generates and multicasts pseudo-data for tree adaptation, and the local group members then record the loss patterns of the packets for comparisons.

The other issue to be considered is secure buffer management after tree changes. The repair buffer of a parent node is maintained for its child nodes, and a data packet stored in the buffer is released when the data packet is acknowledged to have been received by all the children. If a new child node that changed its attachment according to the logical tree adaptation mechanisms requests its new parent node to repair the released data, the data cannot be repaired, and the protocol results in failure. To resolve this problem, all senders should keep a set of data in the buffer even if all of their child nodes acknowledged their receipt so that a receiver who failed to be recovered locally can directly request the senders to repair the lost packets.

IV. Protocol Design

The ECTP-5 transport protocol is designed to support Internet multicast applications. It operates over IPv4/IPv6 networks that have IP multicast forwarding capability with the help of IGMP and IP multicast routing protocols, as shown in Fig. 1.

In an ECTP-5 session, there is a single TCN, which is either one of the participants or a dedicated server. The TCN is responsible for session management, including session creation/termination, late join, session maintenance, and token management. For example, in teleconferencing applications, the TCN may act as the conference server, which may be used for control of the conferencing without sending multicast data. For multi-user on-line games, the TCN may act as the game-control server.

An ECTP-5 participant is called a transport service (TS) user. It can send multicast data packets to the group or receive multicast data packets over the multicast data channel. A TS-user who is sending multicast data in the session is called a sending TS-user (SU). Every SU must have a token for multicast data transmission. In other words, a TS-user who gets a token from the TCN is called an SU.

A TS-user can become an LO or a leaf entity (LE) depending on its role in logical trees. An LO is a representative node of a local group and is designated statically.[1] It is responsible for maintaining an intra-group logical tree of the group and the recovery trees for all SUs in its local group. Each LO is also connected to other LOs via inter-group logical trees. It also periodically generates test traffic for logical tree adaptation. An LE is a member of a local group whose representative is an LO. It should join an intra-group logical tree of the group and is responsible for exchanging control packets with its parent or child LEs along the recovery tree.

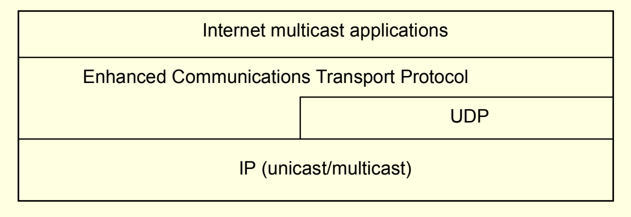
Internet multicast applications
Enhanced Communications Transport Protocol
UDP
IP (unicast/multicast)

Fig. 1. ECTP-5 network model.

[1] In ECTP-5, it is assumed that an LO is located near the egress point of a network, and a prospective TS-user knows which local group (or LO) it should participate in before it joins a session. Since it has been shown that the location of an LO affects the protocol performance significantly [24], [25], further investigation into LO placement and selection mechanisms is required.

There are two phases in the ECTP-5 protocol, a session creation phase and a data transmission phase. In the session creation phase, the session or membership information, which is needed to establish an ECTP-5 session, is exchanged between the TCN- and TS-users prior to sending or receiving data. This phase is required only if the session data should be guaranteed to be delivered to all the members right from the start. In the data transmission phase, the participants send or receive the data.

Figure 2 illustrates the multicast data transport channel in a session. As shown in the figure, the TCN and an SU can transmit multicast data to the other session members over an IP multicast (group) address.

The TCN or an SU will generate data (DT) packets using a segmentation procedure. To do this, the sender splits a multicast data stream of an application into multiple DT packets. Each TS-user delivers all the data packets received to the application in the order sent by the SUs. When reassembling the received data packets, the corrupted and lost packets are detected using a checksum and a sequence number. The lost DT packets are recovered in the error control function.

For reliable transport of multicast data, ECTP-5 builds a two-layer logical tree as shown in Fig. 3. At the lower layer, each LE in a local group joins an LO-rooted shared logical tree (intra-group logical tree; dashed line in Fig. 3). At the upper layer, LOs constitute logical trees for each SU (inter-group logical tree; solid line in Fig. 3). It should be noted that the recovery tree for each SU is derived by grafting these inter-group and intra-group logical trees.

Error control is performed by exchanging control packets between parent and child nodes along the recovery tree. If a packet loss is detected by a gap in the packet sequence numbers, a child node sends a NACK packet to its parent immediately via unicast. The parent LO or SU that receives the NACK packet retransmits the data packet (RD or REPAIR) to the requestor via unicast. Each child generates an ACK packet after receiving a specific number of data packets to make the buffer management of parent nodes secure.

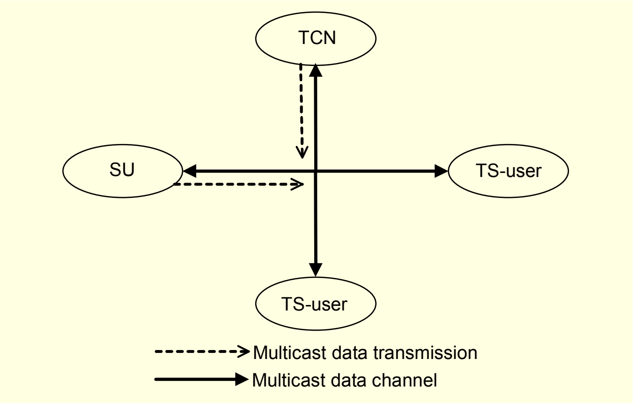

Fig. 2. Multicast data transport in an ECTP-5 session.

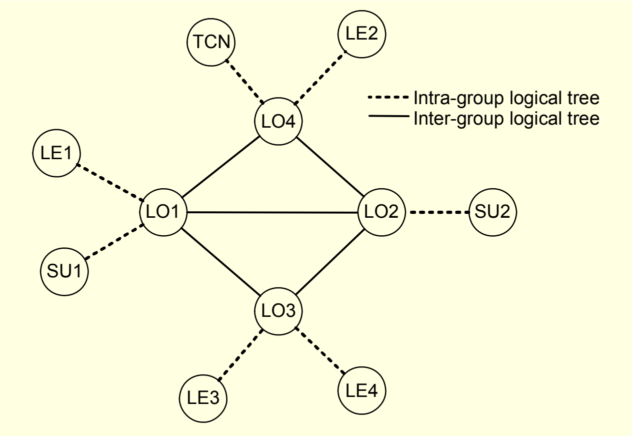

Fig. 3. Two-layer logical trees for an ECTP-5 session.

To make reliability support more efficient, the intra-group logical trees should be as close as possible to the underlying multicast routing tree. The ECTP-5 protocol adopts a logical tree adaptation mechanism using multicast routing tree approximation with error bitmaps of TS-users. An error bitmap represents packet delivery status, which indicates the loss pattern of multicast packets. Each TS-user sends its error bitmap to its parent node with respect to multicast data from the root node of its intra-group logical tree, LO, via periodic ACK messages. By comparing the error bitmaps of itself and those of its children, a node decides whether each child is likely to be its actual child in the underlying multicast routing tree or not. If the child node is determined not to be its actual child, a node starts to change the logical tree by delegating the child node to its parent or one of its other children. After recursive tree changes, the intra-group logical tree converges to another one that is closer to the underlying multicast routing tree.

V. Implementation

1. Protocol Entities

As previously described, ECTP-5 provides several essential mechanisms for reliable multicast transport. In order to realize these, we implemented a prototype system, which consists of a protocol core and 4 control building blocks, including session management, tree-based negative acknowledgement and polling (T-NAPP), generic group/tree (GGT), and logical tree adaptation (LTA). Each of the building blocks provides different functions and performs independently. This enables the building blocks to be reused in other protocol implementations with only small changes in the interfaces. The

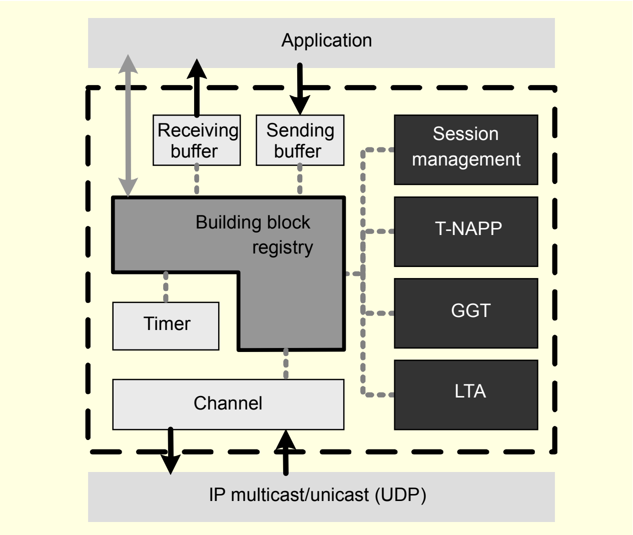

Fig. 4. ECTP-5 protocol entities.

protocol entities are shown in Fig. 4.

The building block registry (BBR) is the heart of the ECTP-5 protocol suite. It deals with all the other protocol entities and coordinates all protocol events to inform the corresponding protocol entities of the events. Each control building block should register itself to the BBR to interact with the other building blocks or to deal with several protocol events, such as incoming/outgoing protocol messages and timer expiration.

The sending buffer (SB) and the receiving buffer (RB) are the intermediate interfaces for sending and receiving data between the application and ECTP-5. The channel sends the outgoing packets handed over from the BBR and also receives the incoming packets from the network to deliver them up to the BBR. The timer manages timed schedules for control building blocks. If a scheduled timer expires, it informs the BBR of the timeout.

In the case of sending data, for example, a control building block for flow/congestion control (CCB) schedules a timer and registers itself to the BBR as a handler for the timer expiry. At the expiration of the timer, the BBR informs the CCB of the event, then the CCB signals for the BBR to pop a set of data out of the SB and to send the data via the channel. In the case of receiving data, the channel delivers the incoming packet to the BBR. Then the T-NAPP, which is triggered by the BBR, inserts the payload of the packet into its repair buffer and performs further actions, such as error detection, error recovery, and data delivery to the application via the RB.

2. Control Building Blocks

The control building blocks perform the main functions of ECTP-5 for reliable multicast data delivery. The functional diagram of the control building blocks is presented in Fig. 5.

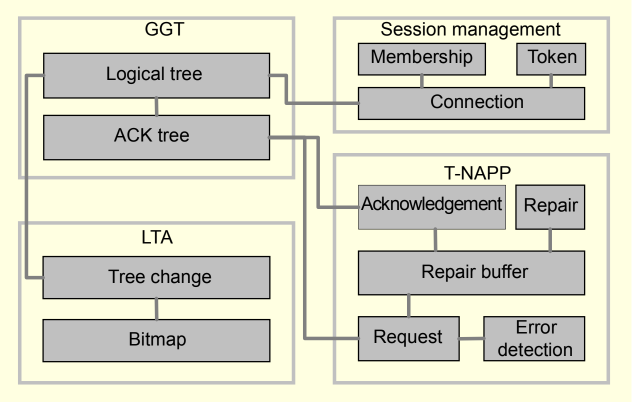

Fig. 5. Functional components of the control building blocks.

The details of each control building block are given in the following.

- Session Management

Connection manages creation and termination of an ECTP-5 session.

Membership manages join/leave of users and maintains membership of the users with probe messages.

Token manages assignment, recall, and status of tokens.

- T-NAPP

Error Detection detects packet losses with examination of the checksum or a gap in the sequence numbers.

Request requests that the parent node repair packet losses along the recovery tree.

Repair sends repair packets in reply to repair requests from child nodes.

Repair Buffer manages packet buffers for each sender to locally recover packet losses of child nodes.

Acknowledgement handles incoming acknowledgements from child nodes to aggregate them and periodically acknowledges successful packet delivery of the node itself and its child nodes to a parent node.

- GGT

Logical Tree manages an intra-group logical tree at a local group for recovery tree construction and its adaptation.

ACK Tree manages recovery trees for each sender that can be derived from logical trees.

- LTA

Tree Change handles the procedures of logical tree change and delegation for tree adaptation.

Bitmap manages and reports the error bitmap of a set of subsequent packets *to* a parent node for logical tree adaptation.

Although the building blocks differ in their features, some of them need to interact with each other. However, there is no way

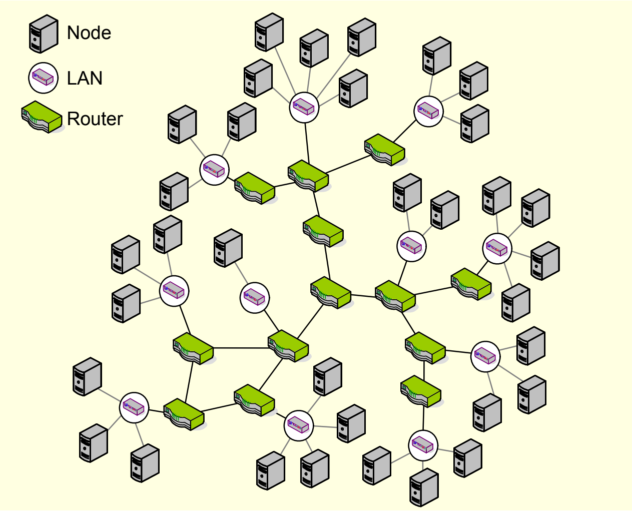

Fig. 6. Test topology (S=30) generated by GT-ITM.

for them to interact directly. They can only interact through the BBR. For example, the request protocol component in the T-NAPP can obtain information about a parent node from the ACK tree in the GGT using a pre-defined interface between the BBR and itself.

VI. Evaluation Results[2)]

We compared the performance of our implementation with that of MCLv3 [27], which is an open-source GNU/GPL implementation of NORM and ALC. We measured control overhead and recovery latency while varying the session size (S) to 30, 60, 90, and 120 nodes and varying the packet loss rates (PLR) of each link to 0.01 (i.e., 1%), 0.001, and 0.0001. We used Emulab [13] as a network testbed to observe their performance in large-scale networks. Emulab provides a realistic and large-scale network testbed with an emulated network and node control system. It allowed us to conduct experiments in networks with hundreds of nodes rather than with 20 or less nodes as in previous work. While varying the number of participants, several test topologies were generated using GT-ITM [28], but they were slightly modified to work around the limitations of Emulab, such as only allowing a maximum of four network interfaces in a node. One example for 30 participants (S=30) is shown in Fig. 6.

A sample application which transfers a 1 MB file was tested on top of ECTP-5. In the experiments, a local group consisted of 10 participants that were close to each other in the topology. In the case of S=30, for example, the test topology consists of 1 TCN, 3 LOs, and 30 LEs. Since we do not cover any congestion control scheme in this study, the data sending rate was fixed at 256 kbps with a payload size of 512 bytes to avoid any side effect from network congestion. The bandwidth of each link was set to 100 Mbps, and the link delays were randomly selected between 6 ms and 24 ms.

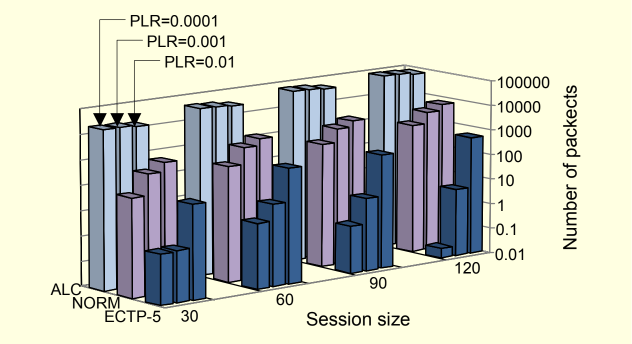

Fig. 7. Repair redundancy.

Figure 7 shows the experimental results for ECTP-5, NORM, and ALC in terms of repair redundancy.

The average number of redundant data or repair packets received by a node was measured. In a reliable multicast protocol, a packet loss can be repaired by a retransmission of the packet (as in NORM and ECTP-5) or using an extra received FEC packet (as in ALC). Any other repair packets which are ignored at the receiver-side incur unnecessary processing overhead for the protocol as well as bandwidth waste. As shown in the figure, the repair redundancy of ECTP-5 was much smaller than that of the others (that is, 5, 4,000, and 80,000 packets in ECTP-5, NORM, and ALC, respectively, at S=120, PLR=0.001) because it uses unicast-based local recovery along the recovery tree. Since an LE repeats the request for retransmission if the retransmission of the repair packet is delayed due to other losses as the PLR becomes higher, the redundancy increases but still remains comparably low.

In contrast, NORM and ALC use multicast-based error recovery so that they suffer from high repair redundancy. A NORM sender multicasts FEC packets in reply to NACK packets from the receivers. If a receiver loses a smaller number of packets than that of FEC packets received, the remainder are redundant. The redundancy of ALC increases with the session size because an ALC sender multicasts a specific number of FEC packets with no consideration of how many packet losses the receivers experience. For the same reason, the total number of received FEC packets decreases as the packet loss rate increases. Note that the decrement is not proportional to the increment of the packet loss rates because receivers should remain in the session longer to receive more FEC packets as the PLR increases.

Figure 8 shows the evaluation results in terms of recovery latency, which is the time taken to receive a repair packet (or an

[2)] Note that the performance evaluation in this paper is to show the feasibility of ECTP-5 compared to the other existing standard protocols. Refer to [12] and [26] to see the validation and evaluation results of the basic architecture.

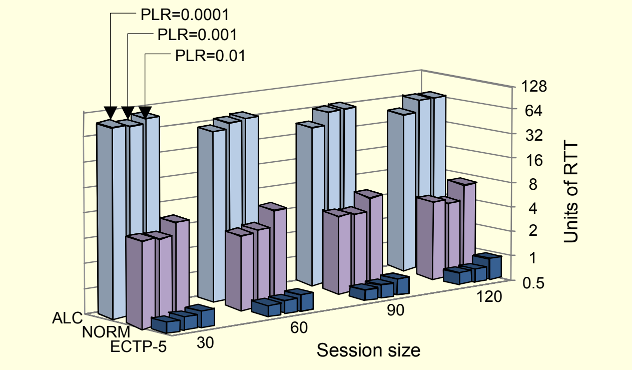

Fig. 8. Recovery latency per RTT.

FEC packet) after detecting the corresponding loss.

The latency is given in units of round-trip time (RTT) from a receiver to the corresponding sender of a lost packet. In the case of a naive unicast-based error recovery mechanism, a receiver directly requests a sender to repair lost packets, and it takes at least one RTT. In ECTP-5, however, the average ratio of recovery latency to RTT is about 0.8 because a packet loss of a participant can be repaired immediately and locally by a parent node, which has better spatial locality than the sender. In NORM, lost packets are repaired by their senders and the NACK packets for them are aggregated at the end of receiving a fragment (indicated by a FLUSH packet). Thus, the recovery latency becomes much larger (6 on average). Using ALC makes it worse (78 on average) because a receiver does not send any feedback for a loss and just waits for another FEC packet for the corresponding fragment. This high recovery latency may hinder NORM and ALC from being used for real-time or interactive applications.

Figure 9 shows the control overhead for error recovery in the protocols. The control overhead is measured as the average number of control packets sent or received by a receiver (namely, NACK and REPAIR packets in ECTP-5; NACK, FEC, and FLUSH packets in NORM; and FEC packets in ALC). The ACK or CMD packets are not included in the measurement because they are related to congestion control rather than error recovery. As shown in the figure, we can conclude that ECTP-5 largely outperforms the others in terms of control overhead. Although the control overhead of each protocol increases with session size, the increase in the overhead of ECTP-5 remains reasonably low. However, the rate of increase in the overhead as the packet loss rate grows is higher than that of other approaches. The reason is that LEs repeatedly send NACK packets to their parents in a short interval (200 ms in this experiment) until the packet losses are repaired. Nevertheless, we can see that the control overhead of ECTP-5 is not very different from that of NORM at PLR=0.01.

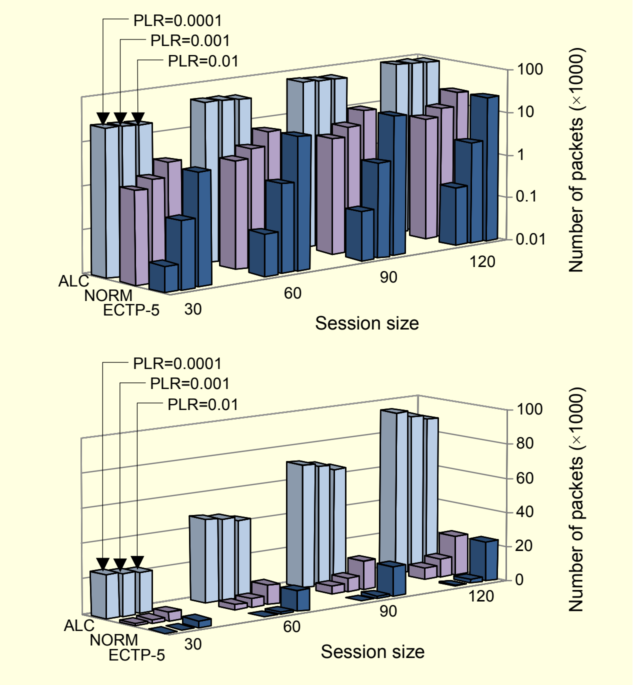

Fig. 9. Control overhead.

VII. Conclusion

To provide reliable multicast transport for multi-party applications, we introduced ECTP-5, which deals with many-to-many applications. It is a scalable many-to-many reliable transport protocol using a 2-layer tree-based error control mechanism and an adaptive logical tree construction algorithm. We provided the detailed protocol architecture and secure protocol procedures in addition to further engineering details, such as session management, dynamic membership support, and fault-tolerant tree maintenance. After developing a prototype system and conducting intensive experiments over Emulab, a realistic and large-scale network testbed, we evaluated the performance of ECTP-5 in terms of scalability while varying the session size and packet loss rates. The experimental results show that ECTP-5 scales well with reasonably small control overhead and low recovery latency compared to NORM and ALC.

As mentioned in section IV, an LO should be carefully placed and selected to maximize the protocol performance in terms of control overhead and recovery latency of its local group members [24], [25]. This can be done using various metrics, such as RTT (as in [12]), topological information, or a combination of them. This topic was already considered in [12], [24] and [25], and will be dealt with further in future work. While this paper does not cover any congestion control mechanism, such a mechanism is essential for a reliable multicast transport protocol to be actually deployed on the

Internet. There are many conventional approaches [29]-[31] which provide congestion control schemes for tree-based reliable multicast transport protocols dealing with scalability and TCP-friendliness. While all of them are applicable to ECTP-5, the approach proposed in [31] seems to be the best candidate for use with many-to-many applications. An integration of this congestion control scheme with ECTP-5, which would result in a complete protocol instantiation, is our future goal.

References

[1] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, Dec. 1997, pp. 784-803.

[2] S. Paul, K. Sabnani, J.C. Lin, and S. Bhattacharyya, "Reliable Multicast Transport Protocol," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, Apr. 1997, pp. 407-421.

[3] B. Whetten and G. Taskale, "An Overview of Reliable Multicast Transport Protocol II," *IEEE Network*, vol. 14, no. 1, Jan. 2000, pp. 37-47.

[4] R. Yavatkar, J. Griffioen, and M. Sudan, "A Reliable Dissemination Protocol for Interactive Collaborative Applications," *ACM Multimedia*, Nov. 1995, pp. 333-344.

[5] B.N. Levine, D.B. Lavo, and J.J. Garcia-Luna-Aceves, "The Case for Reliable Concurrent Multicasting Using Shared Ack Trees," *ACM Multimedia*, Nov. 1996, pp. 365-376.

[6] D.M. Chiu, S. Hurst, M. Kadansky, and J. Wesley, *TRAM: A Tree-Based Reliable Multicast Protocol*, Sun Microsystems Laboratories Technical Report (TR-98-66), July 1998.

[7] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, and J. Crowcroft, *Asynchronous Layered Coding (ALC) Protocol Instantiation*, IETF RFC 3450, Dec. 2002.

[8] B. Adamson, C. Bormann, M. Handley, and J. Macker, *Negative-Acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol*, IETF RFC 3940, Nov. 2004.

[9] S.J. Koh (project editor), *Enhanced Communications Transport Protocol – Part 1 (ECTP-1)*, ITU-T Recommendation X.606 | ISO/IEC IS 14476-1.

[10] S.J. Koh (project editor), *Enhanced Communications Transport Protocol – Part 3 (ECTP-3)*, ITU-T Recommendation X.607 | ISO/IEC FDIS 14476-3.

[11] W. Yoon, D. Lee, and H.Y. Yoon, "Comparison of Tree-Based Reliable Multicast Protocols for Many-to-Many Sessions," *IEE Proceedings-Communications*, vol. 152, no. 6, Dec. 2005.

[12] W. Yoon, D. Lee, H.Y. Youn, and S. Lee, "A Combined Group/Tree Approach for Scalable Many-to-Many Reliable Multicast," *Elsevier Computer Communications Journal*, vol. 29, no. 18, Nov. 2006, pp. 3863-3876.

[13] D. Lee (project editor), *Enhanced Communications Transport Protocol – Part 5 (ECTP-5)*, ITU-T Recommendation X.608 | ISO/IEC FDIS 14476-5.

[14] Emulab - Network Emulation Testbed, http://www.emulab.net/.

[15] IETF Reliable Multicast Transport (rmt) Working Group, http://www.ietf.org/html.charters/rmt-charter.html.

[16] B. Watson, M. Luby, and L. Vicisano, *Forward Error Correction (FEC) Building Block*, IETF RFC 5052, Aug. 2007.

[17] H.K. Kahng (project editor), *Enhanced Communications Transport Protocol – Part 2 (ECTP-2)*, ITU-T Recommendation X.606.1 | ISO/IEC IS 14476-2.

[18] H.K. Kahng (project editor), *Enhanced Communications Transport Protocol – Part 4 (ECTP-4)* ITU-T Draft Recommendation X.607.1 | ISO/IEC CD 14476-4.

[19] H.K. Kahng, *Enhanced Communications Transport Protocol – Part 6 (ECTP-6),* ITU-T Draft Recommendation X.608.1 | ISO/IEC CD 14476-6.

[20] R.B. Adamson, and J. Macker, "Quantitative Prediction of NACK-Oriented Reliable Multicast (Norm) Feedback," *IEEE MILCOM*, Oct. 2002.

[21] C. Neumann, V. Roca, and R. Walsh, "Large Scale Content Distribution Protocols," *ACM Computer Communication Review* (CCR), vol. 35 no. 5, Oct. 2005.

[22] C. Papadopoulos, G. Parulkar, and G. Varghese, "An Error Control Scheme for Large-Scale Multicast Applications," *IEEE INFOCOM*, Apr. 1998.

[23] B.N. Levine, S. Paul, and J.J. Garcia-Luna-Aceves, "Organizing Multicast Receivers Deterministically by Packet-Loss Correlation," *ACM Multimedia*, Sept. 1998, (LPG98), pp. 201-210.

[24] K.R. Kang, S.H. Kim, and D.M. Lee, "An Analysis of the End System Heterogeneity in Many-to-Many Application Layer Multicast," *Lecture Notes in Computer Science*, vol. 3090, Aug. 2004, pp. 1025-1034.

[25] D.U. Lee, S.I. Lee, Y.W. Ko, and D.M. Lee, "Dynamic Core Election in GAM, 2-Layered Many-to-Many Reliable Multicast Protocol," *Korean Information Science Society Conference*, Oct. 2006.

[26] S.I. Lee, S.H. Kim, K.M. Lee, K.R. Kang, and D.M. Lee, "An Analysis of Recovery Mechanism Performance for Many-to-Many Reliable Multicast Protocols," *Korean Information Science Society Conference*, Oct. 2002.

[27] V. Roca et al., "MCLv3: an Open Source GNU/GPL Implementation of the ALC and NORM Reliable Multicast Protocols," http://www.inrialpes.fr/planete/people/roca/mcl/.

[28] K. Calvert and E. Zegura, GT-ITM: Georgia Tech Internetwork Topology Models, http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html.

[29] D.M. Chiu, M. Kadansky, J. Provino, J. Wesley, H.P. Bischof, and H. Zhu, "A Congestion Control Algorithm for Tree-Based Reliable Multicast Protocols," *INFOCOM*, June 2002, pp. 1209-

1217.

[30] I. Rhee, N. Balaguru, and G.N. Rouskas, "MTCP: Scalable TCP-like Congestion Control for Reliable Multicast," *IEEE INFOCOM*, vol. 3, Mar. 1999, pp. 1265-1273.

[31] K.R. Kang, D.M. Lee, and J.Y. Yu, "MTRMCC: A Congestion Control Mechanism for Many-to-Many Tree-Based Reliable Multicast Protocols," *IEICE Transactions on Communications*, vol. 87, no. 6, June 2004, pp. 1601-1609.

Seungik Lee received the BS degree in computer science and engineering from Handong University, Korea, in 2000, and the MS degree in engineering from Information and Communications University (ICU), Daejeon, Korea, in 2002. He is currently a PhD candidate at ICU. His research interests include multimedia multicast, IPTV, and next generation networks.

Yangwoo Ko received the BS degree in computer science and statistics from Seoul National University, Korea, in 1989. He is currently a PhD candidate at Information and Communications University (ICU), Daejeon, Korea. His research interests include future Internet and urban computing.

Dongman Lee received the BS degree in computer engineering from Seoul National University, Korea in 1982, and the MS degree and PhD degrees in computer science from KAIST, Korea in 1984 and 1987, respectively. From 1988 to 1997, he worked as a technical contributor at Hewlett-Packard. He joined Information and Communications University (ICU) at Daejon, Korea as an associate professor in 1997. He became a professor in 2003. He has actively participated in the Korean Internet Address and Name Committee since 1998. He received the Prime Minister Award in recognition for his work in the advancement of the Korean Internet in 2000. His laboratory, Collaborative Distributed Systems and Networks Lab, was appointed as a National Research Laboratory in 2001. He received the Technical Achievement Award from KRNet07, the largest Korean Internet conference. He has published more than 100 papers in international journals and conference proceedings. His research interests include distributed systems, computer networks, mobile computing, and pervasive computing. He is a member of KISS, ACM, and IEEE.