Configuration of ACK Trees for Multicast Transport Protocols

Seok Joo Koh, Eunsook Kim, Juyoung Park, Shin Gak Kang, Ki Shik Park, and Chee Hang Park

For scalable multicast transport, one of the promising approaches is to employ a control tree known as acknowledgement (ACK) tree which can be used to convey information on reliability and session status from receivers to a root sender. The existing tree configuration has focused on a 'bottom-up' scheme in which ACK trees grow from leaf receivers toward a root sender. This paper proposes an alternative 'top-down' configuration where an ACK tree begins at the root sender and gradually expands by including non-tree nodes into the tree in a stepwise manner. The proposed scheme is simple and practical to implement along with multicast transport protocols. It is also employed as a tree configuration in the Enhanced Communications Transport Protocol, which has been standardized in the ITU-T and ISO/IEC JTC1. From experimental simulations, we see that the top-down scheme provides advantages over the existing bottom-up one in terms of the number of control messages required for tree configuration and the number of tree levels.

I. INTRODUCTION

The end-to-end multicast transport is one of the important issues in the next generation Internet research areas. Multicast transport issue includes the reliable multicast such as the error control and congestion avoidance, and the control of multicast session such as status monitoring and session management on top of IP multicast in the networks [1].

Numerous researches on the multicast transport have been made over the past 10-20 years [2], [3]. Based on those works, the IETF Reliable Multicast Transport (RMT) Working Group [4] has been standardizing three RMT protocols: Layered Coding Transport, NACK Oriented Reliable Multicast, and Tree based ACK (TRACK) protocols. On the other hand, ITU-T and ISO/IEC JTC 1 have also made efforts to develop a standard protocol for end-to-end multicast transport under the project of Enhanced Communications Transport Protocol (ECTP) [5]-[7]. ECTP has been designed to support a tight control of multicast sessions and Quality of Service (QoS) management as well as the reliability provisioning for multicast transport [8].

In the ECTP and TRACK protocols, the error and congestion controls are provided over a logical control tree. Such a control tree is called an acknowledgement (ACK) tree in that it specifies the logical paths for ACK packets containing the reliability and session status information from receivers to a root sender. Along the tree hierarchy, ACK packets are forwarded from the receivers to the sender via one or more intermediate repair head (RH) nodes, and the sender and RH nodes will perform retransmission of lost data packets for their children receivers.

While a session is created, an ACK tree is configured to connect the sender and receivers into a hierarchical tree structure. The IETF is developing a tree configuration scheme [9], in

---

Manuscript received February 7, 2001; revised July 31, 2001.

Seok Joo Koh (phone: +82 42 860 6218, e-mail: sjkoh@pec.etri.re.kr), Eunsook Kim (e-mail: eunah@etri.re.kr), Juyoung Park (e-mail: jypark@etri.re.kr), Shin Gak Kang (e-mail: sgkang@etri.re.kr), and Ki Shik Park (e-mail: kipark@etri.re.kr) are with Protocol Engineering Center, Information Support Division, ETRI, Daejeon, Korea.

Chee Hang Park (e-mail: chpark@etri.re.kr) is with Information Security Technology Division, ETRI, Daejeon, Korea

which an ACK tree is automatically constructed by interactions between sender and receivers. This scheme is called a 'bottom-up' approach taking the viewpoint that the ACK tree construction process starts from the leaf receivers, and the tree gradually grows toward the root sender. The related work or its variants can be referred to in the literatures [10], [11].

The bottom-up approach for tree configuration may give some benefits in terms of the tree optimization, but it also involves the following drawbacks:

- *a deeper tree with many tree levels*

The bottom-up scheme is inclined to generate a deep tree with many tree levels since all the nodes try to find the nearest parent nodes without any consideration of the configured tree levels. Note that many tree levels are not desirable for the stable operation of the tree-based reliability mechanisms. For example, the delivery of the error and congestion status information from the leaf receivers to the root sender will be more delayed in a deeper tree due to the frequent ACK aggregations at the intermediate RH nodes.

- *more control messages required for tree configuration*

The bottom-up scheme based on the Expanding Ring Search (ERS) may generate more optimized trees in terms of the time-to-live (TTL) hop distances. It is noted, however, that such optimal trees are obtained at the cost of more control messages for the tree configuration. More control messages require more network bandwidths.

In addition to these drawbacks, the bottom-up scheme cannot guarantee a loop-free tree configuration. To provide a loop-free ACK tree, an additional metric such as distance from the sender must be enforced along with an appropriate activation rule [11]. In particular, the ERS-based scheme depends on the multicast transmission capability of end receivers, while the recently focused multicast routing protocol such as Source Specific Multicast [12] excludes such assumption.

In this paper, we propose an alternative top-down tree configuration [13], in which, only the on-tree RH nodes are allowed to accept children, but an off-tree RH is prohibited to have its children. Thus, an ACK tree will grow from the root sender toward the leaf receivers by including the non-tree nodes in a stepwise manner. The proposed top-down scheme generates a lower-level tree with less control messages. The proposed scheme also provides a loop-free tree intrinsically.

The underlying idea for tree configuration is simple but practical in the viewpoint of the design of a realistic multicast transport protocol. In particular, the proposed tree configuration will be preferred by multicast sessions that require the sender-initiated tight control of overall tree for billing and security purposes, as described in ECTP.

This paper is organized as follows. Section II introduces the ECTP protocol as an example standard protocol for multicast transport, along with tree-based reliability mechanisms. In Section III, the existing ERS-based bottom-up scheme is summarized more in detail. The proposed top-down scheme is described and compared with the bottom-up one in Section IV. Section V shows the experimental results for the evaluation of the proposed scheme under a variety of test environments. In Section VI, we conclude this paper.

II. A MULTICAST TRANSPORT PROTOCOL: ECTP

This section introduces the ECTP protocol, which is a standard protocol for multicast transport based on ACK tree [5], [6]. The ECTP is a transport protocol designed to support Internet multicast applications. ECTP operates over IPv4/IPv6 networks that have IP multicast forwarding capability. Figure 1 shows an overview of the ECTP operations.

Before an ECTP transport connection is created, the prospective receivers are enrolled into the multicast group. Such a group is called an enrolled group. During enrollment, authentication processes may be performed together with group key distribution.

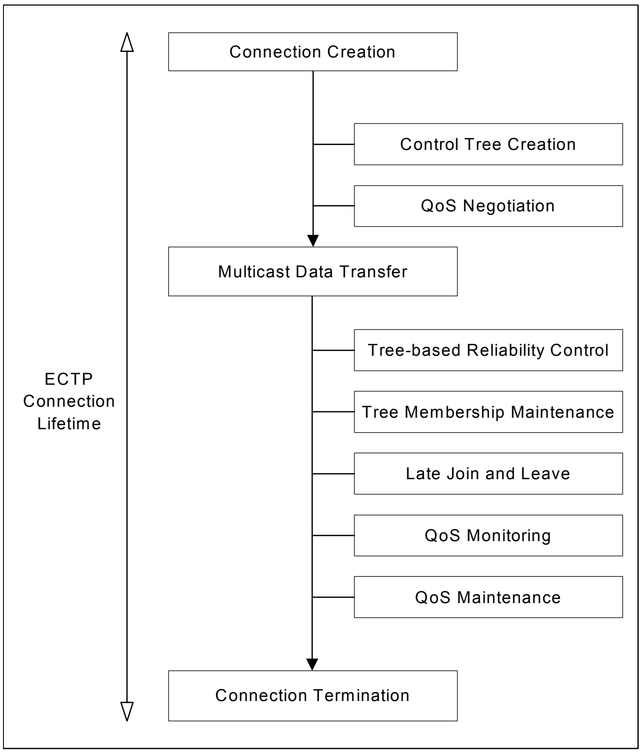


Fig. 1. ECTP protocol operations.

The IP multicast addresses and port numbers must be announced to the receivers. These enrollment operations may rely on the web page announcement.

An enrolled receiver will be connected to the multicast-capable network with the help of the IP multicast routing protocols. Those multicast routing protocols will refer to the announced multicast addresses. An ECTP transport connection will be created for the enrolled receivers.

ECTP is designed to support tightly controlled multicast connections. ECTP sender is at the heart of the multicast group communication. The sender is responsible for the overall management of the connection by governing connection creation and termination, and the late join and leave operations.

The ECTP sender triggers the connection creation process by sending a connection creation message. Each enrolled receiver responds with a confirmation message to the sender. The connection creation is completed when the sender receives the confirmation messages from all of the active receivers, or when the pre-specified timer expires. QoS negotiation may be performed in the connection creation [7].

Throughout the connection creation, some or all of the enrolled group receivers will join the connection. The receivers that have joined the connection are called active receivers. An enrolled receiver that is not active can participate in the connection as a late-joiner. The late-joiner sends a join request to the sender. In response to the join request, the sender transmits a join confirm message, which indicates whether the join request is accepted or not. An active receiver can leave the connection by sending a leave request to the sender.

After a connection is created, the sender begins to transmit multicast data. For data transmission, an application data stream is sequentially segmented and transmitted by means of data packets to the receivers. The receivers will deliver the received data packets to the applications in the order transmitted.

To make the protocol scalable to large multicast groups, ECTP employs the tree-based reliability control mechanisms. A hierarchical tree is configured during connection creation. A control tree defines a parent-child relationship between any two tree nodes comprising a pair. The sender is the root of the control tree. Each local group is defined along the tree hierarchy. A local group consists of a parent and zero or more children. The error, flow and congestion controls are performed for each local group defined by the control tree. Figure 2 illustrates an ACK tree hierarchy for reliability control, in which each node represents a sender (S), a receiver (R), or repair head (RH). In the tree creation, a control tree is gradually expanded from the sender to the receivers. Therefore, it is called a top-down configuration. ECTP has also adopted a bottom-up approach standardized in IETF [9].

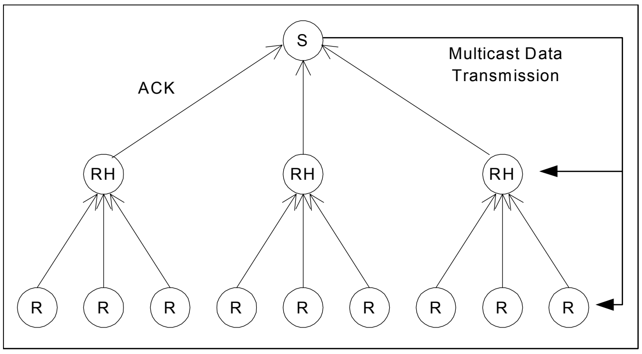


Fig. 2. ACK tree for reliability control.

In ECTP, error control is performed for each local group defined by a control tree. If a child detects a data loss, it sends a retransmission request to its parent via ACK packets. An ACK message contains the information that identifies the data packets successfully received. Each child can send an ACK message to its parent using one of the two ACK generation rules: ACK number and ACK timer. If data traffic is high, an ACK is generated for the ACK number of data packets. If the traffic is low, an ACK message will be transmitted after the ACK timer expires. After retransmission of data, the parent activates a retransmission back-off timer. During the time interval, the retransmission request(s) for the same data will be ignored. Each parent can remove the data acknowledged by all of its children from its buffer memory.

Tree-membership is maintained during the connection. A late-joiner is allowed to join the control tree. The late-joiner listens to the heartbeat messages from one or more on-tree parents and then joins the best parent. When a child leaves the connection, the parent removes the departing child from the children-list. Node failures are detected by using periodic control messages such as null data, heartbeat and acknowledgement. The sender transmits periodic null data messages to indicate that it is alive, even if it has no data to transmit. Each parent periodically sends heartbeat messages to its children. On the other hand, each child transmits periodic acknowledgement messages to its parent.

During the data transmission, if severe network congestion is indicated, the sender suspends the multicast data transmission temporarily. In this period, no new data is delivered, while the sender transmits periodic null data messages to indicate that it is alive. After a pre-specified time, the sender resumes the multicast data transmission.

After an ECTP connection is created, QoS monitoring and maintenance operations are performed for the multicast data transmission [7]. For QoS monitoring, each receiver is required to measure the parameter values experienced. Based on the measured values, a receiver determines a parameter status value for each parameter. These status values will be delivered to the sender via ACK packets. Sender aggregates the parameter status values reported from receivers. If a control tree is employed, each parent RH node aggregates the measured values reported from its children and forwards the aggregated values to its parent via its ACK packets.

The sender takes QoS maintenance actions necessary to maintain the connection status at a desired QoS level, based on the monitored status values. Specific rules are pre-configured to trigger QoS maintenance actions such as data transmission rate adjustment, connection pause and resume, and connection termination. Those rules are based on observation that how many receivers are in the abnormal or possibly abnormal status.

The sender terminates the connection by sending a termination message to all the receivers after all the multicast data are transmitted. The connection may also terminate due to a fatal protocol error such as connection failure.

III. BOTTOM-UP TREE CONFIGURATION

The tree configuration currently proposed in IETF [9] can be considered as a bottom-up approach from the viewpoint that all the leaf receivers try to find their parent RH nodes in parallel and the tree is expanded from the leaf receivers toward the root sender.

A generic tree configuration proceeds in the order of session advertisement, RH discovery, and binding to the best RH. In the session advertisement, each receiver realizes the existence of a session and the sender by using an out-of-band mechanism such as Web page announcement. In this way, the receiver will obtain the multicast group address and the sender's address, and the other information necessary for the ACK tree construction. When the sender indicates the creation of an ACK tree, each receiver begins the RH discovery process to find one or more candidate RH nodes that are active in the session. Among the candidate RH nodes discovered, a receiver selects and binds to the best RH node as its parent by using a pre-configured rule such as TTL distance or IP address.

Depending on the RH discovery mechanism employed, the detailed tree configuration procedures are slightly different from each other. In [9], three different RH discovery mechanisms are provided: point of contact (POC) based, generic router assist (GRA) based and expanding ring search (ERS) based.

A POC is a dedicated server that is established for the ACK tree construction in the network. Each RH node enrolls itself to the POC which is located in its domain. Each receiver asks the POC "which RH nodes are active in the domain." The POC then responds to the receiver with a list of active RH nodes. The GRA approach tries to construct an ACK tree that keeps congruency with the underlying multicast routing tree topology. To do this, each GRA router collects the routing tree information and delivers it to the downstream receivers for the associated multicast group.

In the ERS-based scheme, a receiver discovers the RH nodes by using multicast query messages. Each receiver first asks if there is a neighboring RH node by multicast. One or more RH nodes may respond to the query message. If there is no response from RH nodes, the receiver will re-send the query messages with an increased TTL value until at least one RH node is discovered.

We note that both POC and GRA schemes depend on the dedicated facilities such as POC servers or GRA routers that have been pre-configured in the network. The corresponding performance for the tree configuration also depends on how much those facilities are deployed. This paper thus focuses on the ERS-based scheme as a promising bottom-up tree configuration scheme.

The ERS-based bottom-up tree configuration scheme is illustrated in Fig. 3, and the control messages employed for the tree configuration are summarized in Table 1.

For tree configuration, the sender first indicates the tree creation by sending a session BEACON message to all the receivers and RH nodes.

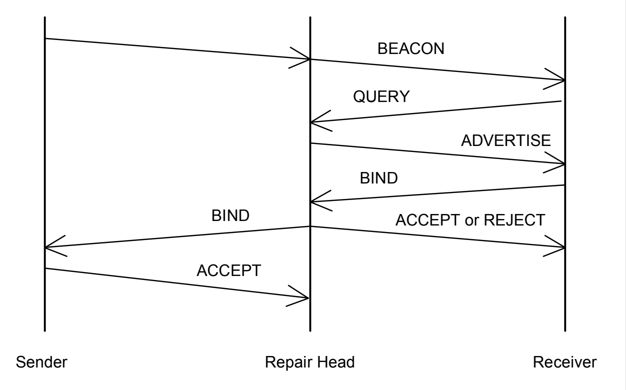


Fig. 3. ERS-based bottom-up configuration scheme.

Table 1. Control messages employed for the ERS-based bottom-up configuration.

Message Types	From	To	Transport	Description
BEACON	Sender	Receivers	Multicast	Indication of tree creation
QUERY	Receiver	RHs	Multicast	Finding candidate RHs
ADVERTISE	RH	Receivers	Multicast	Response from RH
BIND	Receiver	RH	Unicast	Bind request
ACCEPT or REJECT	RH	Receiver	Unicast	Response to the bind request

The BEACON message contains a metric, by which each RH or receiver can measure the TTL distance from the sender, which will be used to generate a loop-free ACK tree, as described later.

When the tree creation is indicated, each receiver begins to find the candidate RH nodes by sending a multicast QUERY message. Some of the RH nodes may respond with a multicast ADVERTISE message. The ADVERTISE message may contain the TTL distance from the sender to the responding RH node. If there is no response from RH nodes, the receiver retransmits the QUERY message by increasing the TTL value of the corresponding IP packet.

Once one or more candidate RH nodes are discovered, the receiver selects the best RH node. The specific rule for the selection typically depends on the implementations. In the ERS-based scheme, the RH node with the shortest TTL distance from the sender is preferred. If a tie occurs in the TTL distance, the RH node with the lowest IP address will be chosen. It is noted that the TTL distance and the IP address can be used to guarantee the generation of a loop-free ACK tree since all the RH nodes can be distinctively arranged in the order of those two metrics.

The receiver binds itself to the chosen RH by sending a BIND message, and the RH node will respond with an ACCEPT or REJECT message. The decision of ACCEPT or REJECT may be made based on the tree parameters such as the maximum number of children. If a receiver receives a REJECT message, it will retry to join another RH node.

When an RH node which is not on the tree has its own child, it begins the process to find its parent node. The procedural steps for parent discovery are the same as those for RH discovery. If the sender is located in the neighborhood, the RH node will bind itself to the sender without using a QUERY message. If some of the receivers have failed to attach to the tree, they may join the tree as late-joiners, which is outside the scope of this paper.

IV. TOP-DOWN TREE CONFIGURATION

The bottom-up tree configuration seems to be effective in terms of tree optimization. However, such a scheme is inclined to generate more control messages for the tree configuration and a deeper tree with more tree levels than necessary. In this section, a top-down tree configuration scheme is proposed which aims to overcome these drawbacks of the existing approach.

1. Preliminaries

Depending on multicast deployment in the network, the sender and RHs may share a single IP multicast address or use different IP multicast addresses. In the networks where sender and RHs use different multicast addresses or channels, all of the multicast addresses employed must be announced to the group members before the ACK tree is created. In this case, one of them is used for the multicast data transmission by sender, and the others are for the multicast control by RHs such as the multicast data retransmission and the multicast transmission of control messages. If a single multicast address is shared by sender and RHs, the TTL-scoped multicasting is required for each multicast traffic to restrict the multicast traffic into its local scope. In this paper, the TTL-based traffic scoping is assumed for fair comparison of the bottom-up and top-down schemes.

Before a tree is created, every prospective group member must be attached to the network interface with the help of the IP multicast routing protocols. This ensures that the enrolled member listens to the multicast control packets from sender and RHs for tree configuration.

2. Tree Configuration Mechanisms

The proposed top-down scheme is similar to the existing bottom-up approach from the viewpoint that the same control messages as those used in the existing scheme are employed for the tree configuration. However, the QUERY messages from the receivers to the RH nodes are not used in the proposed scheme. The procedural steps for the tree configuration are also different from those in the bottom-up scheme.

In the proposed scheme, an ACK tree gradually grows from the root sender to the receivers via one or more RH nodes. When an RH node is on the tree, it begins to find its children nodes by sending a multicast ADVERTISE messages to non-tree nodes.

More specifically, the top-down tree configuration steps can be summarized as follows:

(A) Sender transmits a BEACON message for indication of the tree creation.
(B) Some of the neighboring receivers or RH nodes join the sender by sending a BIND message. The sender responds with the corresponding ACCEPT or REJECT message.
(C) When an RH node is on the tree, it begins to transmit ADVERTISE messages to invite non-tree nodes. Each non-tree receiver will collect the information on the list of on-tree RH nodes by receiving the ADVERTISE messages from one or more RH nodes.
(D) When a non-tree RH or receiver receives one or more ADVERTISE messages, it selects the best on-tree RH node and sends a BIND message to the selected RH node.
(E) The on-tree RH node determines whether the BIND request must be accepted or rejected. In the REJECT case,

the receiver will try to join another on-tree RH node.

(F) These procedures are repeated until all the nodes are included into the ACK tree.

Figure 4 illustrates an instance of the top-down tree configurations. As shown in the figure, the QUERY message is not used in the top-down configuration. Instead, each on-tree RH node triggers the invitation of the non-tree RH or receiver nodes by sending a multicast ADVERTISE message.

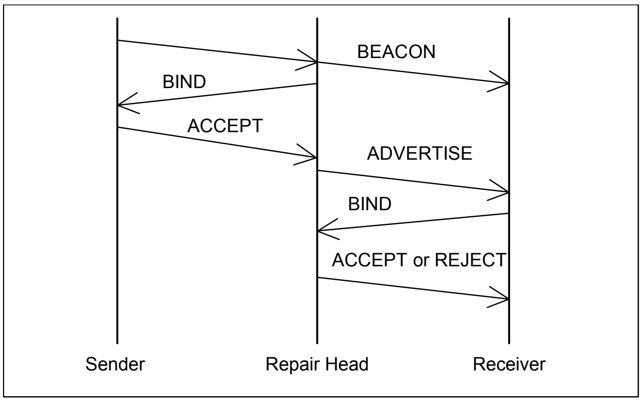


Fig. 4. Top-down tree configuration scheme.

3. Comparison of the top-down and bottom-up schemes

In the top-down configuration scheme, the on-tree RH nodes invite the non-tree nodes by sending multicast ADVERTISE messages, and thus the QUERY messages from the receivers are not used. This ensures that the ACK tree gradually grows from the root sender to the leaf receivers by including non-tree nodes in a step-wise manner. Figure 5 shows the difference between the two schemes.

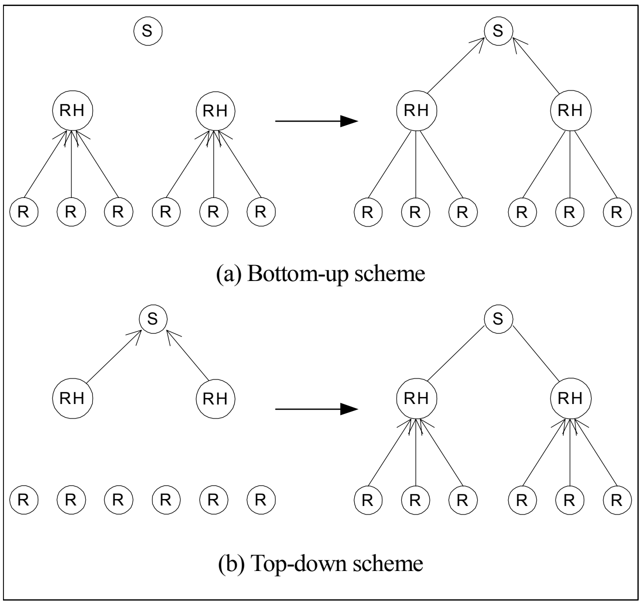
The top-down configuration gives some advantages over the bottom-up scheme, which include:

(A) Reduced Control Messages for Tree Configuration

The top-down scheme basically gets rid of the QUERY messages used in the bottom-up approach. In addition, it does not depend on the TTL increases unlike the ERS-based scheme. Owing to these features, the control messages required for the tree configuration are dramatically reduced.

(B) Lower Number of Levels in Tree

Since the tree gradually expands from the root sender to the leaf receivers, the top-down scheme is inclined to generate a tree with as few tree levels as possible. Fewer levels are preferred to reduce the feedback time from the leaf receivers to the sender in the TRACK protocol operations.



(a) Bottom-up scheme

(b) Top-down scheme

Fig. 5. Comparison of the bottom-up and the top-down schemes.

(C) Simpler to Implement

The bottom-up scheme needs to use an additional metric such as the TTL distance from the sender for generating a loop-free tree, while the top-down scheme can guarantee no loops since only the on-tree RH nodes are allowed to invite their children.

(D) Independency of the Underlying Multicast Routing

The ERS-based scheme requires the multicast transmission capability of end receivers so as to find a parent node by using Query message (see Table 1 of Section III), while in the top-down scheme such an assumption is not required, since the multicast transmissions are allowed for only RH nodes.

V. EXPERIMENTAL RESULTS

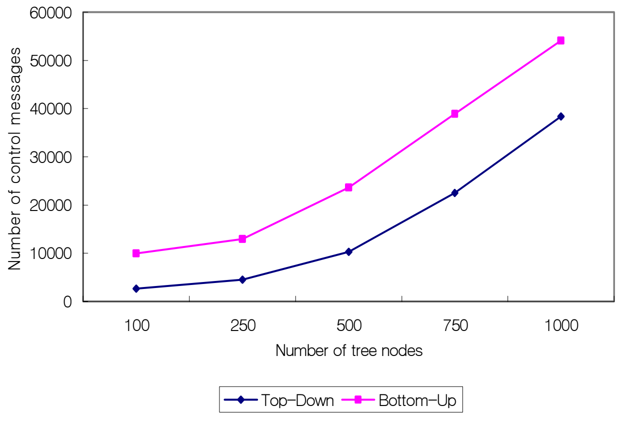
In this section, we compare the performance of the existing ERS-based bottom-up with the proposed top-down tree configuration schemes through experimental simulation results.

1. Simulation Environments

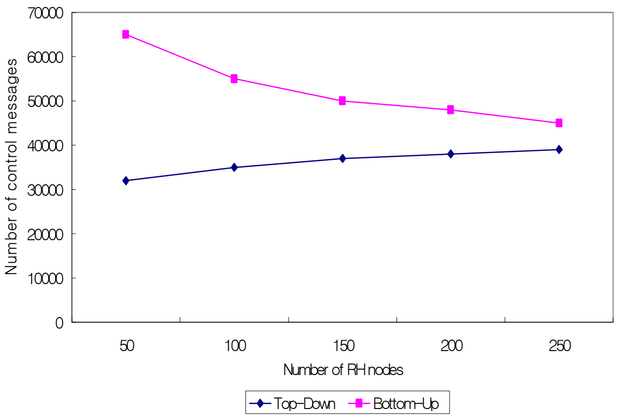
All the simulation experiments are performed for 100, 250, 500, 750 and 1000 tree nodes. Each node in the network is randomly plotted in the plane of 35 × 35 pixels, where one integer pixel unit represents one TTL hop distance. The link distance between any pair of nodes is given by calculating the TTL link distance between those two nodes on the plane. For each test network, 10% of the overall tree nodes are assigned as the RH nodes, and the maximum number of children allowed for an RH is set to 25 nodes.

For the test networks with 1000 nodes, we also performed the experimental simulations to evaluate the impact of the number of RH nodes on the performance. In these experiments, the number of RH nodes is increased from 50 to 250 by 50.

In the experiments, each multicast message has its own TTL value for the 'TTL-scoping'. First, the TTL value of the BEACON message is set to 60, which provides a whole coverage of the network. In the ERS-based scheme, the TTL values of the QUERY messages are increased from 5 to 10 by increments of one. On the other hand, the ADVERTISE messages have the TTL value of 10 in both the bottom-up and top-down schemes.



(a) Comparison for different number of tree nodes



(a) Comparison for different number of RH nodes

Fig. 6. Performance comparison based on the number of control messages generated.

2. Performance Metrics

To evaluate the performance of the tree configuration, we employ the following metrics:

(A) Number of control messages generated during the tree creation: we counted all the control messages including the BEACON, QUERY, ADVERTISE, BIND, and ACCEPT/REJECT messages.

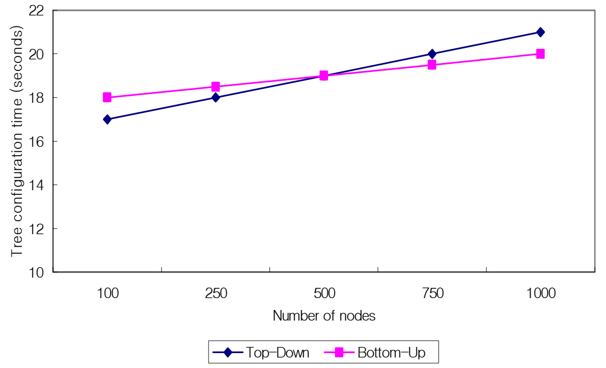
(B) Tree configuration time: this represents the elapsed time until all the nodes are connected to the tree.

(C) Number of tree levels for the configured ACK tree.

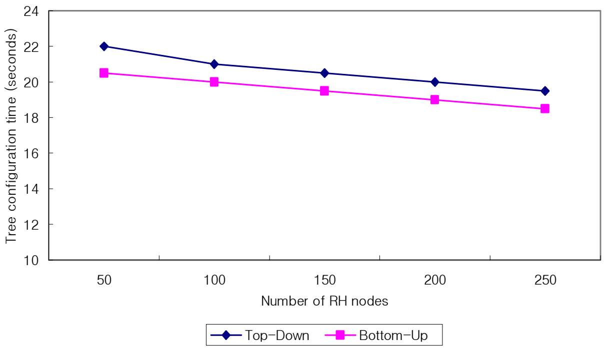
(D) Sum of the link costs for the configured tree in the TTL hop distances; this is measured by summing up the TTL hop distances of the tree links on the configured tree.

3. Simulation Results

Figure 6 shows the performance comparison in terms of the number of the control messages generated for the tree configuration. From Fig. 6(a), it is shown that the top-down scheme reduces the number of the control messages required, compared with the bottom-up scheme. Figure 6(b) gives the simulation results for the different number of RH nodes in the networks with 1000 nodes. From the figure, we can see that as the number of RH nodes increases, the performance of the bottom-up scheme gets better and the performance gap between those two approaches becomes smaller. This implies that the bottom-up scheme operates more efficiently in the groups with more RH nodes in terms of the number of control messages.



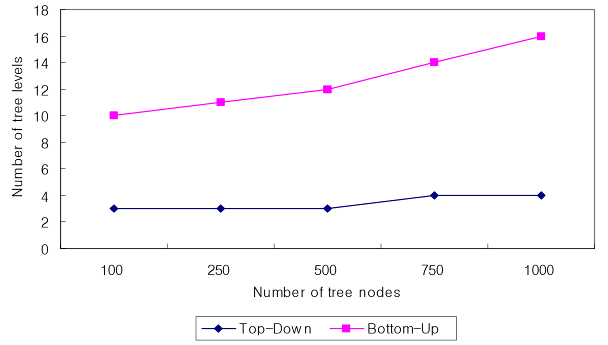
(a) Comparison for different number of tree nodes



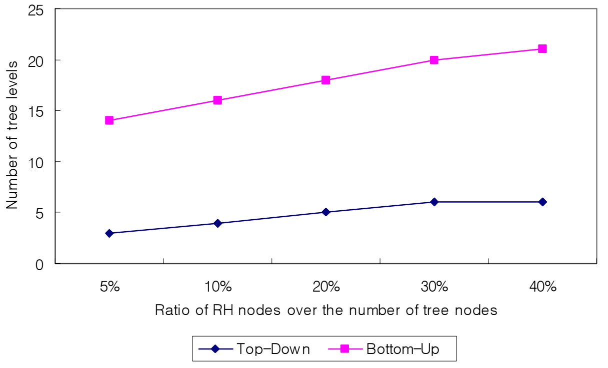
(b) Comparison for different number of RH nodes

Fig. 7. Performance comparison based on the tree configuration time.

Figure 7 compares the performance of the top-down and bottom-up schemes in terms of the tree creation time. In Fig. 7(a), the measured tree creation times are plotted for different numbers of tree nodes. From the figure, it is shown that the bottom-up scheme provides better performance for the tree groups with more than 500 nodes, but the top-down scheme is better in the relatively small-sized groups. Figure 7(b) shows the simulation results for different numbers of RH nodes in the groups with 1000 nodes. As the number of RH nodes increases, the tree creation time slightly decreases.



(a) Comparison for the different number of tree nodes

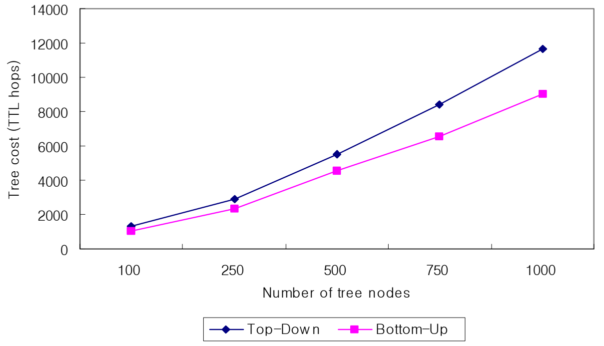


(b) Comparison for the different number of RH nodes

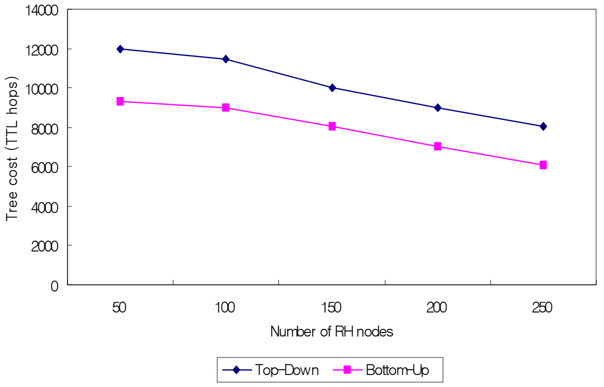
Fig. 8. Performance comparison based on the tree depth of the configured trees.

Figure 8 shows the performance in terms of tree levels of the configured tree. Figure 8(a) plots the results for different numbers of tree nodes, and Fig. 8(b) shows the performance for different numbers of RH nodes in the network with 1000 tree nodes. From the figures, it is clear that the top-down mechanism generates fewer tree levels. The number of tree levels generated by the bottom-up scheme is nearly three or four times that by the top-down scheme. As shown in Fig. 8(b), as the number of RH nodes increase, deeper trees are generated containing more tree levels.

Figure 9 shows the simulation results for the tree cost in units of TTL hop distances. First, Fig. 9(a) shows the performance for different numbers of tree nodes, and Fig. 9(b) shows the performance for different numbers of RH nodes in the groups with 1000 tree nodes. From the figures, we can see that the bottom-up scheme generates more optimized trees in TTL hops than the top-down scheme. As shown in Fig. 9(b), the tree link cost gets much smaller as the number of RH nodes increases.



(a) Comparison for the different number of tree nodes



(b) Comparison for the different number of RH nodes

Fig. 9. Performance comparison based on of the tree link cost of the configured trees.

4. Summary

From the various experimental results to date, we can make the following observations:

(A) In terms of the number of the control messages required for tree configuration, the top-down scheme is preferred to the bottom-up scheme. This implies that the top-down approach is more suitable to employ in networks where the bandwidth resources are scarce. It is also noted that the bottom-up scheme generates less control messages in the groups with more RH nodes.

(B) In terms of the tree configuration time, the bottom-up mechanism is better in the large-sized groups, while the top-down scheme provides better performance in the relatively small-sized groups.

(C) In terms of the depth of the configured tree, we see that the top-down mechanism is inclined to generate a tree with fewer levels, compared with the bottom-up mechanism. It is noted that fewer tree levels make it simple to keep the tree-based reliable control mechanisms stable. The tree depth does not depend on the number of tree nodes and RH nodes in the network.

(D) In terms of the tree cost optimization, the bottom-up scheme is better than the top-down scheme. This benefit comes at the cost of more control messages and more tree levels. The tree cost optimization can be much more easily achieved in groups with more RH nodes.

VI. CONCLUSIONS

In this paper, the two tree configuration schemes, top-down and bottom-up, are discussed and compared for the tree-based multicast transport protocols. The top-down scheme is preferred to the existing bottom-up approach, in terms of the number of control messages required for the tree configuration and the depth of the configured trees. The top-down configuration is also simple to implement from the viewpoint that the tree configuration scheme itself guarantees a loop-free ACK tree.

But, the existing bottom-up mechanism still provides some advantages in terms of tree optimization. Based on the comparison and observations made in this paper, it is recommended that these two schemes be selectively used according to the requirements of applications and the underlying networks.

REFERENCES

[1] S. Koh *et al.*, “Minimizing Cost and Delay in Shared Multicast Trees,” *ETRI Journal*, vol. 22, no. 1, Mar. 2000, pp. 30-37.

[2] K. Obraczka, “Multicast Transport Protocols: A Survey and Taxonomy,” *IEEE Communication Magazine*, vol. 36, Issue1, Jan. 1998, pp. 94-102.

[3] Reliable Multicast Links, http:// research.ivv.nasa.gov/ RMP/ links. html.

[4] IETF Reliable Multicast Transport WG, http://www.ietf.org/html.charters/rmt-charter.html.

[5] ITU-T SG7 Q.8 | ISO/IEC JTC 1/SC 6/ WG 7, *Enhanced Communications Transport Protocol: Specification of Simplex Multicast Transport*, ITU-T draft Recommendation X.ectp-1 and ISO/IEC FCD 14476-1, Sept. 2001.

[6] ITU-T SG7 Q.8 | ISO/IEC JTC 1/SC 6/ WG 7, *Enhanced Communications Transport Protocol: Specification of QoS Management for Simplex Multicast Transport*, ITU-T draft Recommendation X.ectp-2 and ISO/IEC CD 14476-2, Sept. 2001.

[7] *Enhanced Communications Transport Protocol,* http://ectp.etri.re.kr/.

[8] S. Koh *et al.*, "Framework of Connection Management for Multicast Transport," *Proc. of the APCC2000 Conf.*, 30 Oct. - 2 Nov., Seoul, Korea, 2000.

[9] M. Kadansky *et al.*, *Reliable Multicast Transport Building Block: Tree Auto-Configuration*, IETF Internet Draft, draft-ietf-rmt-bb-tree-config-01.txt, Nov. 2000.

[10] B. Levine, D. Lavo, and J. Garcia-Luna-Aceves, “The Case for Reliable Concurrent Multicasting Using Shared ACK Trees,” *Proceeding of the fourth ACM Int’l Conf. on Multimedia*, 1996, pp. 365-376.

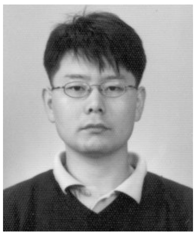
[11] C. Maihofer, “Scalable and Reliable Multicast ACK Tree Construction with the Token Repository Service,” *Proceeding of the IEEE Int’l Conf. on Networks 2000*, pp. 351-358.

[12] S. Koh *et al.*, “Enhanced Core Based Tree for Many-to-Many IP Multicasting,” *Telecomm. Review*, vol. 11, no. 3, June 2001, pp. 485-493.

[13] E. Kim, S. Koh *et al.*, “An ACK Tree Creation Mechanism using Top-Down Scheme for Tree-based Reliable Multicast Transport Protocols,” *Proceeding of the IC2001 Conf.*, July 2001, Las Vegas, USA.

Seok Joo Koh received B.S. and M.S. degrees in management science from KAIST in 1992 and 1994 respectively. He also received Ph.D. degree in industrial engineering from KAIST in 1998. His research topic of dotoral dissertation was the communication network design and planning in the optimization perspective. Since 1998, he has been with Protocol Engineering Center, ETRI. He has actively participated in ECTP (Enhanced Communications Transport Protocol), project of ITU-T SG7 and ISO/IEC JTC1/SC6, as a project editor since 2000. He has also been involved in the IETF RMT WG. His current research interests include internet multicasting and mobile internet.

Eunsook Kim received B.S., M.S., and Ph.D. degrees in computer science from Sookmyung Women’s University, Korea in 1996, 1998, and 2001, respectively. Since March 2001, she has been with Protocol Engineering Center, ETRI. Her current research interests include IP multicasting, reliable multicast and session-group management.



Juyoung Park received B.S. and M.S. degrees in electronics engineering from Chungnam Nat'l Univ. in 1995 and 1997, respectively. He also received Ph.D. degree in electronics engineering from Chungnam Nat'l Univ. in 2001. Since 2001, he has been with Protocol Engineering Center in ETRI. His current research interests include internet multicasting, internet QoS, and mobile internet.

Shin Gak Kang received B.S., M.S., and Ph.D. degrees in electronics engineering from Chungnam National University, Daejeon, Korea in 1984, 1987, 1998, respectively. He joined ETRI in 1984 and is currently a team leader of Communication Protocol Standardization Research Team in Protocol Engineering Center (PEC). He has actively participated in ITU-T Study Group 7 as Rapporteur for end-to-end QoS multicast issue since 1997. He is also leading the VoIP Forum in Korea as vice-chairman. His current research interests include reliable multicast, voice over IP, and internet security.

Ki Shik Park was educated at Seoul National University, in the Rep. of Korea, where he obtained a first class honors degree of B.A. in 1982 and M.A. in 1984. And he got Ph.D. degree in the field of telecommunications policy in 1995. He joined ETRI in 1984, and he is currently in charge of the Protocol Engineering Center of ETRI as a principal researcher. He has 17-year-research experience in various divisions of ETRI including Telecommunication Systems Division, Information & Telecommunications Technology Division, etc. Currently, he is vice-chairman of ITU-T TSAG and chairman of WP3/ TSAG, and a member of ASTAP Advisory Board. He also serves as vice-chairman of Technical Assembly of Telecommunication Technology Association (TTA) of Korea.

Chee Hang Park received the B.S. degree in applied physics from Seoul National University, Korea in 1974, the M.S. degree from Korea Advanced Institute of Science and Technology, Korea in 1980, and the Ph.D. degree in computer science from University of Paris 6, France in 1987. During the last 10 years, he has been involved as project leader in several large projects such as Multimedia Computer System Development and High Speed Parallel Computer System Development. His research interests include multimedia systems, distributed system, middleware, group-ware, network virtual computing, and mobile agent architecture. He is currently executive director of Information Security Technology Divi-sion, ETRI.