A Router Assisting Control Tree Configuration Mechanism for Reliable Multicast

Eunsook Kim[1], Seok Joo Koh[1], Juyoung Park[1], Shin-Gak Kang[1], and Jongwon Choe[2]

[1] Electronics and Telecommunications Research Institute,
161 Kajeong-Dong, Yuseong-Gu, Deajeon, KOREA
{eunah, sjkoh, jypark, sgkang}@etri.re.kr
[2] Sookmyung Women's University,
53-12 Cheongpa-Dong 2Ka, Yongsan-Gu, Seoul, KOREA
choejn@sookmyung.ac.kr

Abstract. For reliable multicast service, the mechanism based on hierarchical control tree can be a promising solution to avoid well-known feedback implosion. However, configuration of an efficient control tree is very difficult for IP Multicast because it does not provide explicit membership and routing topology information to upper layer protocol. If the transport layer tree and the network layer tree are very different, it may take large cost to handle control messages. Especially, when a node at a downstream link of the network routing tree becomes a parent node of an upstream link at the control tree of transport layer, the discrepancy between routing tree and control tree causes to waste network resources by redundant messages. This problem can be solved if router that knows the information on routing topology can support configuration of a control tree and reliable delivery. However, the change of router function embraces deployment problem. Thus, this paper proposed a very simple method of router assist to minimize change of router functions. With this method, routers are only required to recognize message types of control messages in order to forward the messages to correct direction: upstream or downstream.

1. Introduction and Problem Statement

As current IP multicast[2] provides only best-effort service, reliable multicast transport service should be implemented on it. Over the past several years, many studies on the reliable multicast transport have been made[4], [7], [16], and of those research works, the Tree-based ACK(TRACK) protocol is one of the promising RMT protocols of which the error recovery, congestion control and other functionalities are provided over a pre-configured logical tree[9], [15]. This mechanism is known to provide high scalability as well as reliability[10] since it reforms a multicast group into a logical tree rooted by a sender. It classifies its receivers into Service Nodes(SN) and receivers. SNs are intermediate nodes which take charge of local retransmission, while receivers only denotes leaf nodes of the tree.

As the performance of this mechanism may highly depend on the efficiency of the control tree, many researchers have been making their efforts to build an efficient control tree[9]. However, to configure a logical tree is difficult because IP Multicast does not provide the information of explicit membership and routing topology to upper layer protocols.

I. Chong (Ed.): ICOIN 2002, LNCS 2343, pp. 84-93, 2002.

Because of this problem, a transport layer tree may not be congruent with a underlying routing tree, and in this case, reliable multicast services may take large cost. The Fig. 1 shows an original multicast tree, and Fig. 2 draws a possible control tree derived from it. We should notice that due to the lack of information on the underlying network topology, node *a* would bind to *f* lying in its downstream link rather than *b* placed on its upstream link, as *f* and *b* have the same hop distance from *a*. It is also noticeable that *f* can choose *c* as its SN, while *b* and *f* have the same hop distance. Let's suppose that link *e3* is failed for a moment. Children nodes of *c* including *d*, *e* and *f* will request retransmission to *c*, while *a*, *g*, and *h* request the error repair to *f*. The node *c* also sends repair request to *b*. In this situation, some links should carry the same data on several times, because of the discrepancy between the routing tree and the control tree.

For example, *e8* should deliver the data three times for one repair, when *b* multicasts the repair data, *c* repairs the loss for *f*, and *f* transmits the retransmission data to *a*. The node *a* is put on the most severe situation among the receivers. It undergoes redundant retransmission as well as long delay of repair. When we deliberate the case of *a*, it could have been considerably reduced the overhead if it bound to its upstream SN, *b*. It can be possible when it knows the underlying network topology. Eventually, we can say that the best control tree is to match the underlying multicast routing tree topology[14].

There have been some researches to support underlying routers to assist reliable service like GRA [1] or PGM [4]. However, router functions should be modified in order to apply the router assist mechanism, in which the process takes times to deploy. It can be used as an assisted method when it is supported. So, an easy and simple solution which helps to prompt fast deployment should be studied.

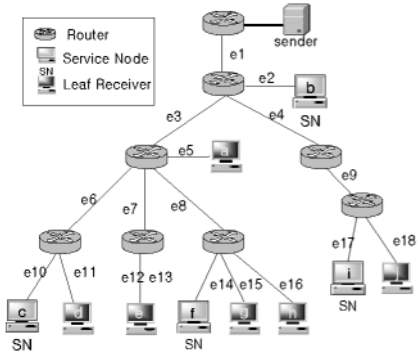

Fig. 1. Multicast Routing Tree

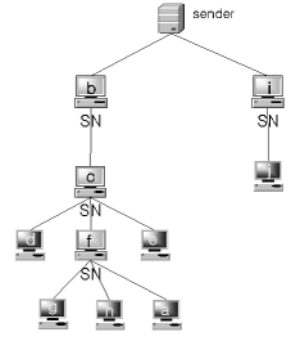

Fig. 2. A Possible Control Tree

Thus, this paper proposes a simple router assist mechanism aimed at configuration of a scalable, efficient, and robust control tree that keeps a close shape with multicast tree. The proposed mechanism does not require the underlying routers to keep the state information for reliable service unlike the previous works[4]. In addition, it does not require routers to be aware of error recovery[14]. Routers simply provide the path

for receivers to reach other receivers. We believe this simple way may facilitate router to assist reliable service.

This paper is organized as follows: Section 2 provides a brief summary of related works and Section 3 describes the proposed scheme. The evaluation and analysis of the proposed scheme are shown in Section 4. In Section 5, we conclude this paper.

2. Related Works

Addressable Internet Multicast(AIM) [11] is a scheme that uses forwarding services that require routers to assign per-multicast group labels to all routers participating in that group. AIM uses these labels to send a request towards the sender which get redirected to the nearest upstream member. If data is available, the NACK receiver responds with a retransmission which is also forwarded according to the router labels. However, to keep the label information may cause scalability problem when the group size grows.

PGM [4] peeks into transport headers to filter messages. NACKs create state at the routers that are used to suppress duplicate NACKs and guide retransmissions to receivers that requested them. PGM creates a hierarchy rooted at the source, but provision is made for suitable receivers to act as Designated Local Retransmitters(DLSs) if desired. However, this work requires that routers keep and check the packet sequence number from transport layer. For this operation, routers should maintain buffer and it may add overhead of routers.

LMS [14] proposes to use minimal router support not only to make informed parent/child allocation, but also to adapt the hierarchy under dynamic conditions. With LMS, each router marks a downstream link as belonging to a path leading to a *replier*. A replier is simply a group member willing to assist with error recovery by acting as a parent for that router's immediate downstream nodes. Because repliers are selected by routers, parents are always upstream and close to their children. The forwarding services introduced by LMS allow routers to steer control messages to their replier, and allow repliers to request limited scope multicast from routers.

Active Error Recovery(AER) [7] is very similar to LMS. In AER, each router that has a repair server attached periodically announces its existence to the downstream routers and receivers, and serves as a retransmitter of the lost data on the subtree below it, or collects and sends NACKs upstream.

However, in these works, the function which routers select and maintain repliers should be added in routers. It may not be easy to routers to be largely changed and give some overhead to routers to function the operation.

Generic Router Assist(GRA) [1] gives guidelines to design router assist mechanism for reliable multicast. It proposes to use minimal router support for loss recovery. In these router assisted schemes, hierarchy construction is achieved by routers keeping minimal information about parents for downstream receivers, then carefully forwarding loss recovery control and data messages to minimize implosion problem. With this approach, the constructed control tree can 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 concerned multicast group. In the GRA, hierarchy construction requires little explicit mechanism at the expense of adding router functionality. The proposed work would be designed to keep the principles of the guidelines.

3. The Proposed Router Assist Mechanism

The control trees are used for forwarding control information towards the root or data towards the leaf nodes. A generic tree configuration of Reliable Multicast Transport(RMT) proceeds in the order of session advertisement, SN discovery, and binding to the best SN.

In the session advertisement, each receiver realizes the existence of a session and the sender by using an out-of-band mechanism such as SAP[6], or Web announcement. In this process, the receiver will obtain the multicast group address, the sender's address, and the other information needed for the construction of a control tree. When the sender indicates creation of a control tree, each receiver begins SN discovery process to find one or more candidate SNs that are active in the session. Among the candidate SNs discovered, a receiver selects and binds to the best SN by using a pre-configured rule such as TTL distance or IP address.

The message types used for control tree creation are *BEACON*, *ADVERTISE* (or *QUERY* and *QUERY-RESPONSE)*, *BIND*, and *ACCEPT* or *REJECT*. *BEACON* message is multicast to the all group members to indicate a control tree creation. When a *BEACON* message is transmitted to the group, SN discovery process will be started. There is two ways to seek an SN for a receiver. One is that each SN seeks its child receivers by sending *ADVERTISE* messages within a scoped area[12]. In the other way, each receiver solicits its SN with *QUERY* messages as it widens the message scope until it receives a response message(*QUERY-RESPONSE*)[16]. For both of methods, each receiver who receives the message tries to bind to a SN with *BIND* message, and each SN should respond with *ACCETP* or *REJECT* message about the bind request.

This paper chooses the first method because it has shorter procedures of message exchanging, and may guarantee less message overhead, because the message overhead is not inclined to grow depending on the number of receivers unlike the latter method which *QUERY* messages are used.

With the message exchanging scheme, the most important fact of control tree to be efficient is that it should be congruent with underlying routing tree as possible as it can, in order to reduce redundant control messages. As Fig. 2 shows, a control tree would suffer from redundant messages if a downstream node becomes a parent SN of an upstream node. To prevent from making such relationship, the proposed mechanism configures a control tree as follows:

Step 1. The sender multicasts *BEACON* messages for indication of the tree creation into the multicast session. Each router recognizes the input link of the message as upstream link.

Step 2. Each SN sends *ADVERTISE* messages to invite receivers after a *BEACON* message is indicated. It forwards *ADVERSITE* messages only to its downstream links. For example, in the Fig. 1, we assume that the node *b*, *c*, *f*, and *i* are designated as SNs. In this case, *ADVERTISE* messages from *b* are forwarded to all downstream receivers, and *ADVERTISE* messages from *c* are forwarded to *d*, the messages from *f* passes to *g* and *h*, and the messages from *i* are delivered to *j*.

Step 3. Receivers now can explicitly send a *BIND* message to their closest parent. Closeness is estimated by hop distance in this stage. RTT or other metrics can be considered in later. In Fig. 1, node *c*, *e*, *f*, and *i* only receive *ADVERTISE* messages from *b*, whereas node *d* receives *ADVERTISE*s >from *b* and c, *g* and *h* get from *f*, and *j* receives them from *b* and *i*, respectively. Now, it is clear that *c*, *e*, *f* and *i* try to bind to node *b*, and *d* will try to bind to *c*. The nodes *g* and *h* will try to bind to *f*, and *j* will do to *i* as estimating hop distance.

Step 4. Each SN determines whether the BIND request must be accepted or rejected. In the *REJECT* case, it should notify the reason.

Fig. 3 illustrates the forwarding process of *ADVERTISE* messages. We see that the operation of tree configuration can be simplified with this router assist, as there is no modification of router functions except choosing downstream or upstream by message types. In this mechanism, we do not need to calculate *Sender Distance*[1] and *Neighbor Distance*[2], or tree level of each node which is needed to select an SN and to avoid a loop under no router assist mechanisms, which may add complexity in the process of a control tree creation. Moreover, it can avert that a SN placed on a downstream link from a receiver becomes the parent SN, which causes redundant data delivery and inefficient link usages.

Fig. 4 shows the control tree established from Fig. 1 with the proposed router assist mechanism. As we see, it maintains congruency from the underlying routing tree as every parent is an upstream node of its children, where the link overhead for exchanging control messages can be minimized.

When a receiver reports losses to its SN via *ACK* packet, the SN sends *RDATA*, retransmitted data, corresponding the *DATA* packet. The only change for this router assist mechanism is that each router forwards *RDATA* only to the downstream links.

With this mechanism, the TTL-scope, which is controversial of the stability on the current Internet, is not needed. Furthermore, error request and repair processes are efficiently performed because SN of each receiver is located in the close area on real network topology.

For instance, if we suppose again that link *e3* is failed for a moment. In Fig. 2, children nodes of *c* including *d*, *e* and *f* will request retransmission to *c*, while *a*, *g*, and *h* request the error repair to *f*. The node *c* also sends repair request to *b*. In this case, each link should carry the same data several times, such as *e8* should deliver the data three times for one repair, when *b* multicasts the repair data, *c* repairs the loss for *f*, and *f* transmits the retransmission data to *a*. The node *a* is put on the most severe situation among the receivers. It undergoes redundant retransmission as well as long delay of repair.

However, in Fig. 4, *e8* only gets request of retransmission from *f*, and delivers retransmitted data one time towards *f*, *g*, and *h*. We can obviously see how the case of node *a* can be improved in the proposed scheme. It is simply served by *b* without unnecessary transit path.

---

1 Distance >from an SN to the sender

2 Distance from a receiver to a neighboring SN

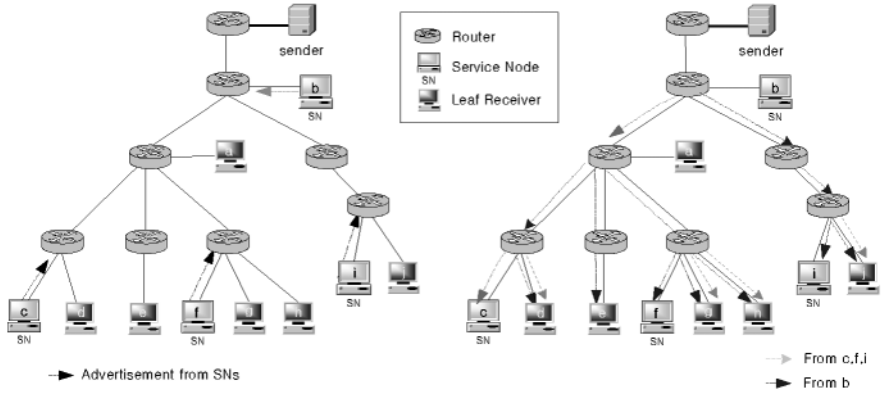

Fig. 3. Forwarding path of the ADVERTISE message

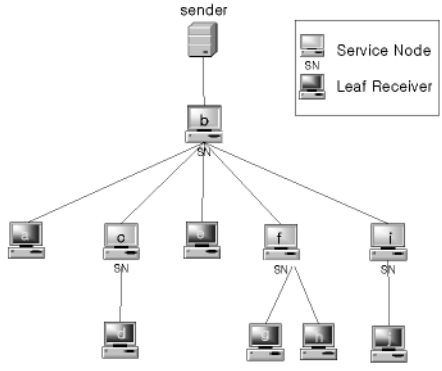

Fig. 4. A configured control tree from Fig. 1

4. Evaluation and Analysis

In section 3, we insist that a parent SN always lies on upstream link or at least in the same level of its children in the proposed mechanism. With the feature, we see the configured control tree keeps congruent with routing tree and reduce inefficient usage of network links We will evaluate the proposed mechanism with both of mathematical evaluation and simulation.

4.1 Mathematical Evaluation

We will show that the proposed mechanism keeps congruency with the given routing tree as we verify the following proposition:

Proposition. A parent SN is always in the upstream link or at least in the same network of its children.

To prove this, the notations below are defined first.

$N = (V, E)$, where V = a set of node, and E = a set of link: N indicates network which is illustrated by a directed graph.

$e_{ij} = (v_i, v_j) \in V$, where $i \neq j$, $v_i \geq v_j$ which means v_i is an upstream node of v_j: link e implies a path between v_i and v_j;

$Cost(e_{ij}) : E \rightarrow I+$, where $i \neq j$: link cost of e from v_i to v_j. The cost value is assigned as hop counts between the two nodes;

$G = \{v_1, \ldots\ldots, v_k\} \subseteq V$: a set of multicast group members;

$R = \{r_1, \ldots\ldots, r_n\} \subseteq G$, where $n \leq k$: a set of multicast group receivers;

$SN = \{sn_1, \ldots\ldots, sn_m\} \subseteq G$: a set of SNs;

$CSN(r) = \{sn_1, \ldots\ldots, sn_p\} \subseteq SN$: a set of candidate SNs for a receiver, r;

$Child(sn) = \{c_1, \ldots\ldots, c_q\} \subseteq R$: a set of children for sn;

$Deliver(e_{ij}) \in \{0, 1\}$: Delivery vector of *ADVERTISE* messages, indicating if a message is forwarded via the link. *0* implies no message delivery.

Now, we can define the following rules of SN measurement and selection.

$$r \in Child(sn_k),$$

$$\text{iff } Cost(sn_k, r) = min\,(Cost(sn_m, r)),$$

$$\text{where } \forall\ sn_k \in CSN,\ sn_m \in CSN,\ m = 1 \text{ to } p, \text{ and } 1 \leq k \leq m \quad ⓐ$$

$$Cost(sn_i, r) < Cost(sn_j, r),\ \forall\ sn_i < sn_j,\ iff\ r \leq sn_i \quad ⓑ$$

Together with the above formulas we can prove the Assertion 1 as follow:

Proof.

We suppose that there is a receiver, $\{r_i, r_j\} \subseteq R$ and $CSN(r) = \{sn_k, sn_l\}$, where $r_i < sn_k < r_j < sn_l$, and we know that an *ADVERTISE* message is always forwarded by downstream of a router. Then,

$$Deliver(sn_k, r_i) = 1,\ Deliver(sn_l, r_i) = 1,\ Deliver(sn_k, r_j) = 0,\ Deliver(sn_l, r_j) = 1 \quad ⓒ$$

$$\text{From ⓒ, } CSN(r_i) = \{ sn_k, sn_l\}, \text{ and } CSN(r_j) = \{ sn_l\}$$

$$\text{Then, } Cost(sn_k, r_i) < Cost(sn_j, r_i),$$

Thus, $r_i \in Child(sn_k)$ from ⓐ and ⓑ, and $r_j \in Child(sn_j)$ because the sn_j is the only SN available to r_j.

Now, we see that both of r_i and r_j have their SNs with upstream nodes from themselves.

Thus, we can say that the proposed mechanism guarantee the receivers choose their parents among the upstream nodes.

4.2 Simulation Results

In order to estimate the performance of the proposed mechanism, we perform experimental simulations by network simulator *ns*[9]. Control traffic overhead and data recovery time are evaluated under conditions that 1024byte fixed sized packets are generated with 10ms CBR(Constant Bit Rate) stream, a packet loss is forced to occur in 3 consecutive packets, and it is supposed that every loss happens in only data packets not in control packets.

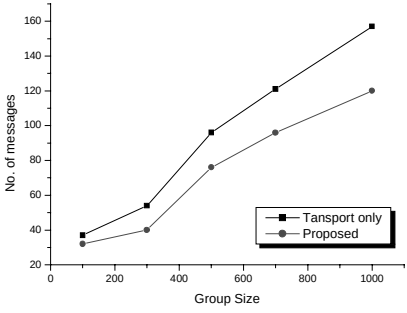

Fig. 5. Control traffic overhead

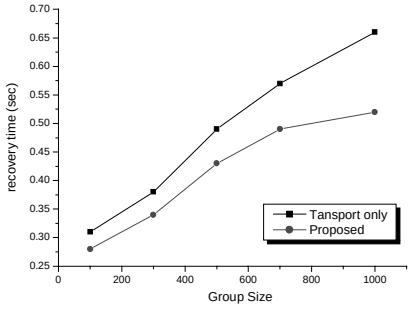

Fig. 6. Data recovery time

Fig. 5 and Fig. 6 shows the evaluation results of control traffic overhead and data recovery time respectively. 'Transport only' means a control tree built in transport layer. We model HiRM[8] for our simulation. In both metrics, proposed mechanism shows better performance. Especially, the gab between two mechanisms has become bigger when the group size gets growing.

In short, these simulation results stand for stability and scalability of the proposed mechanism, because the performance that this mechanism shows does not much depend on the group size.

4.3 Relationship with Network Layer

The proposed mechanism is independent from a special network layer, but it will be well-matched with SSM[5] or IPv6[3]. Current reliable multicast protocols targets one-to-many multicast service, and SSM provides simple and efficient routing mechanism for one sender based multicast service. It is well matched with the proposed mechanism.

However, the router assist mechanism has a deployment problem. It is not easier to be changed for already-existing routers. So, if we put this functionality into IPv6

routers which is not much deployed yet, it can be smoothly deployed as the network grows. In addition, its address scheme is hierarchically designed, so the each address reflects in which network it belongs to. It is very useful to choose an SN when there exists more than one SN in the same region.

5. Conclusions

In tree-based reliable multicast protocols, it is very important to design an efficient logical tree. However, to configure a logical tree is difficult because IP Multicast does not provide explicit membership and routing topology information to upper layer protocols.

Thus, there have been some researches to use routers to build a control tree which is to match the underlying multicast routing tree topology. However, it has deployment problem, so an easy and simple solution to router to assist reliable delivery.

This paper proposes a simple router assist mechanism aimed at configuration of a scalable, efficient, and robust control tree that keeps a close shape with multicast tree. The proposed mechanism does not require the underlying routers to keep the state information for reliable service and to be aware of error recovery. Routers simply provide the path for receivers to reach other receivers.

With this simple way, we see that the proposed mechanism guarantee that the receivers choose their parents among the upstream nodes. In addition, the simulation results show that this mechanism stably operates in different group sizes. We believe this simple way may facilitate router to assist reliable service.

To evaluate the proposed mechanism more specifically, now we are simulating the proposed scheme with various topology and metrics. In addition, we put our efforts to enhance the mechanism.

References

1. B. Cain, T. Speakman and D. Towsley, Generic Router Assist(GRA) Building Block – Motivation and Architecture, IETF Internet Draft, March 2000
2. S. Deering, Host Extensions for IP Multicasting, RFC1112, August 1989
3. S. Deering and R. Hinden, “Internet Protocol, Version 6(IPv6), Specification,” RFC 2460, December 1998
4. D. Farinacci, A. Lin, T. Speakman and A. Tweedly, PGM reliable transport protocol specification, IETF Internet Draft, Auguest 1998
5. H. Holbrook and B. Cain, Source-Specific Multicast for IP, IETF Internet Draft, November 2000
6. M. Handley, C. Perkins and E. Whelan, Session Announcement Protocol, IETF Internet Draft, March 2000
7. S. K. Kasera, S. Bhattacharyya, M. Keaton, et al., Scalabel Fair Reliable Multicast Using Active Services, IEEE Network Magazine (Special Issue on Multicast), January/February 2000
8. E. Kim, Design and Analysis of a Hierarchical Reliable Multicast, Ph. D Dissertation, June 2001

9. M. Kadansky, B. Levine, D. Chiu, et al., Reliable Multicast Transport Building Block: Tree Auto-Configuration, IETF Internet Draft, draft-ietf-rmt-bb-tree-config-01.txt, November 2000
10. B. Neil Levin and J.J. Garcia-Luna-Aceves, A Comparison of Known Classes of Reliable Multicast Protocols, Proceedings of International Conference on Network Protocol (ICNP-96), 1996
11. B. N. Levine and J. J. Garcia-Luna-Aceves, Improving Internet Multicast Routing with Routing Labels, IEEE International Conference on Network Protocols (ICNP '97), October 1997
12. S. McCanne, S.Folyd, NS (Network simulator), http:://www-nrg.ee.lbl.gov/ns, 1995
13. S. J. Koh, E. Kim, J. park, S. G. Kang, et al., Cofiguration of ACK Trees for Multicast Transport Protocols, ETRI Journal, Vol.23, September 2001
14. C. Papadopoulos, G. Parulkar and G. Varghese, An Error Control Scheme for Large Scale Multicast Applications, In proceedings of IEEE INFOCOMM '98, March 1998
15. B. Whetton, D. Chiu, M. Kadansky, and G. Taskale, Reliable Multicast Transport Building Block for TRACK, IETF Internet Draft, draft-ietf-rmt-bb-track-00.txt, November 2000
16. R. Yavatkar, J. Griffioen and M. Sudan, A Reliable Dissemination Protocol for Interactive collaborative Applications, University of Kentucky, 1995