PATH-DIVERSITY OVERLAY RETRANSMISSION ARCHITECTURE FOR RELIABLE MULTICAST

Wenjun Zeng, Yingnan Zhu, Haibin Lu, and Hongbing Jiang
Computer Science Dept., Univ. of Missouri-Columbia, MO 65211
{zengw, luhaibin}@missouri.edu

ABSTRACT

IP-multicast is a bandwidth efficient transmission mechanism for group communications. Reliability in IP-multicast, however, poses a set of significant challenges. To address the reliability and scalability issues in IP-multicast, this paper proposes a novel overlay retransmission architecture that exploits path-diversity by taking advantages of both IP multicast and an overlay network. We show that the proposed path diversity overlay retransmission architecture has the potential to significantly improve the reliability, delay, playback quality, and scalability of IP-multicast based multimedia applications. The general concept of using P2P overlay networks to help improve the QoS performance of multimedia applications as illustrated in this paper is expected to have significant impact on the deployment of next generation multimedia services.

1 INTRODUCTION

Transporting audio/video over the best-effort IP networks has been considered a cost-effective way to deploy many multimedia applications such as video conferencing, video-on-demand, distance learning, remote collaboration. To help address data loss in difficult network environments, forward error correction (FEC) and selective re-transmission have been proposed to recover lost packets. FEC introduces bit overhead in the transmission, thus should be used judiciously. For example, it is typically considered more appropriate for wireless channels. On the other hand, selective re-transmission is generally considered more bandwidth efficient, with the cost of some delay. There is on-going work in the Internet Engineering Task Force (IETF) to support RTP (Real-time Transport Protocol) retransmission (with the support of early feedback and more frequent RTCP (Real-time Control Protocol) feedback [1]) which is considered to be an effective packet loss recovery technique for real-time applications with relaxed delay bounds [2].

Different applications may have different delay requirements. For example, a video conferencing application has very stringent delay requirement (less than a few hundred milliseconds), while a video-on-demand streaming application can tolerate a delay of a few seconds. Some other applications such as live distance learning with one way audio/video streaming and some limited interactivity, e.g., student feedbacks through message board, may have a delay requirement somewhere in between. There are also scenarios where different Quality of Service (QoS) requirements need to be simultaneously met in a single transport session [3]. Selective retransmission, when done wisely, would significantly improve the audio/video playback quality, especially when the network condition is challenging.

IP multicast is a bandwidth efficient transmission mechanism for applications where there are multiple participants. Although current Internet does not fully support IP multicast, it will be fully supported in the next generation Internet. For example, Internet-2 [4], a private network used for education and research, fully supports IP multicast, and application platforms such as Microsoft Research's ConferenceXP platform [5] have been developed for such networks. Reliability in multicast scenarios is typically more challenging than unicast scenarios as it suffers from several problems, including diverse client bandwidth and capabilities, bandwidth inefficiency in multicast retransmission, and potential sender overloading. This paper investigates the retransmission architectures and mechanisms to achieve more effective and reliable packet delivery in video multicast. We propose a novel overlay retransmission architecture to exploit path diversity to address the above reliability-related issues. We evaluate its performance in comparison to the traditional approach and show the significant advantages it provides in supporting the QoS performance (reliability, delay, playback quality, scalability) of multimedia applications. Note that the concept of peer-to-peer networking and path diversity has been exploited recently to improve the performance of file sharing, content distribution, and media streaming, e.g., in [6][7]. In this paper, we introduce a novel exploitation of path diversity and peer-to-peer overlay network to improve the retransmission performance for *both live and on-demand* media distribution.

The rest of the paper is organized as follow. Section 2 briefly discusses the challenges in reliable IP multicast. Section 3 presents our proposed path-diversity overlay retransmission architecture. The algorithm developed to construct the overlay retransmission network and dynamically choose the retransmission nodes is presented in Section 4. Section 5 presents the simulation results. We conclude in Section 6.

2 RELIABILITY IN IP MULTICAST

Traditionally, to address reliability, the receiver sends the retransmission requests to the original sender who then may choose to retransmit the lost packets if deemed important [2]. A common problem in this approach is that it is very likely that the retransmitted packets will go through roughly the same routes as the original packets. However, the loss of the original packets is an indication that the route is probably congested. As a result, the retransmitted packets are also likely to experience congestion and loss, especially in a bursty loss scenario. If not done wisely, the retransmitted packets may even deteriorate the congestion condition [2]. In an IP multicast scenario, there is also scalability problem in supporting retransmission. Typically the retransmitted packets are also multicast to ALL participants who have subscribed to the retransmission session, even though only one or few participants might have experienced the loss of that packet. This would be a waste of the bandwidth. The original sender may choose to only retransmit (multicast) packets that are requested by many participants. The downside of this strategy is that additional delay may be introduced in order for the sender to collect the aggregated feedbacks from multiple participants, and that some lost packets will not be retransmitted. Separate unicast session can also be established by the original sender to convey retransmissions to each of the requesting receivers. This, however, will significantly increase the load of the original sender.

As discussed above, to achieve reliable IP multicast delivery is very challenging. To address the limitations of traditional approaches, we propose a path-diversity overlay retransmission architecture in the next section.

3 A PATH-DIVERSITY OVERLAY RETRANSMISSION ARCHITECTURE

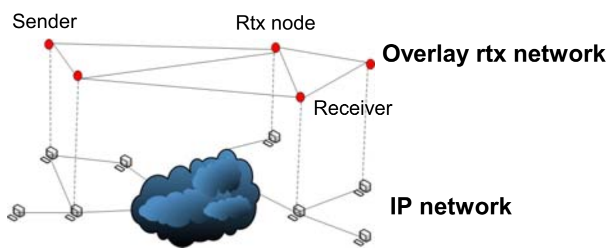

Fig. 1: An overlay retransmission network architecture

The basic idea of the proposed path diversity overlay retransmission is to build a very simple overlay network among the participants. Since the overlay network is mainly used for retransmission purpose, the construction and maintenance of the overlay network should be very lightweight. Each receiver only needs to identify one or two nodes for retransmission purpose, called *retransmission nodes* (with possibly a couple of backups). The retransmission nodes for the receiver should be those end hosts that have been recently having little problem receiving packets from the original sender (source), have a good network connection to the receiver, and are typically closer to the receiver than the original sender. The quality of the network connection between the candidate retransmission nodes and the receiver can be estimated by the receiver, for example, based on the regular data packet transmission from the retransmission node to the receiver in a videoconferencing scenario (i.e., the retransmission node itself is one of the sending sources), or by regular probing in a streaming scenario. The quality of the network connection between the original sender and the candidate retransmission nodes is estimated by the candidate retransmission nodes themselves and then conveyed to the receiver upon probing. When there are multiple sending sources (e.g, in videoconferencing), each receiver identifies a couple of "good" retransmission nodes for each potential sending source.

With the identification of a couple of good retransmission nodes (and possibly a couple of backup retransmission nodes), a receiver can send the re-transmission request (with the original sequence number of the lost packets and some other necessary information) to one of its retransmission nodes who typically would have received or will receive that packet. The retransmission node then, upon receiving the re-transmission request, forwards the requested packet using a separate *unicast* RTP session to the requesting receiver. If the retransmission nodes are chosen intelligently, the retransmitted packet would have a much better chance of getting to the requesting receiver reliably and timely, improving the user experience of the requesting receiver. Furthermore, this overlay retransmission architecture provides load balancing that redistributes the retransmission load of the original sender to other peers, making the system much more scalable. The retransmission nodes are highly distributed without the limitation of a regular topology, which makes the system very robust to network failure. It also addresses the potential bandwidth inefficiency problem of the traditional approach that uses multicast retransmission.

To further improve the reliability and reduce the delay, the receiver can explore a few options. It can choose to send the retransmission request to the original sender as well so that two retransmitted packets may be sent for some *important* packets to increase its chance of being reliably and timely received by the receiver. For example, the original sender may have the information about the importance of the lost packet and can choose to participate in the retransmission of only those important packets. The second option a receiver has is to ask two of its retransmission nodes to retransmit the lost packet in a difficult network environment. This may help in case one of the retransmission nodes itself will receive the packet late.

We choose to use a separate unicast session to send the retransmission packets for each receiver, a good strategy that has been discussed sufficiently in [2]. As a matter of fact, we can exactly follow the recommendations made in [2] on the retransmission payload format, association of a retransmission stream to its original stream, use of the retransmission payload format with the extended RTP profile for RTCP-based feedback (including retransmission requests) [1], congestion control, and other considerations, keeping in mind that the retransmission session is between the receiver and the retransmission node, as opposed to the original sender.

As mentioned above, the overlay network considered here is for retransmission purpose only. The traffic generated by the retransmission is only a small portion of the overall video/audio streams. Thus it may be an overkill to try to optimize the overlay structure as it may incur large overhead to maintain such *optimum* (i.e., dynamically changing the overlay based on the current network condition). Therefore, we decided to use a *lightweight* overlay network. There is no central controller, and each receiver picks a couple of retransmission nodes. Each receiver periodically probes its retransmission nodes including backups. Based on its own load, the retransmission node only accepts a limited number of overlay requests. Each receiver monitors the quality of the network connection between the original sender and itself, and this information is conveyed to other receivers through regular RTCP control packets when probed by other receivers. Therefore, the only extra traffic is generated by probing. To reduce the traffic, the probing interval can be relatively large, e.g., every tens of seconds. We can also use triggered probing when the receiver detects that the connection from the source is about to get worse, e.g., based on measured packet delay. We will show in Section 5 that with a small cost of probing, the overlay retransmission network can significantly improve the system performance in the case of difficult and dynamic network environment.

4 CONSTRUCTION OF THE OVERLAY RETRANSMISSION NETWORKS

In this section, we describe in more detail the distributed algorithm we developed to build the light-weight overlay network. The process of building the overlay network mainly concerns the selection of the best retransmission nodes for each receiver. Once the retransmission nodes are determined, RTP unicast channels can be established directly between each receiver and its retransmission nodes. The algorithm is run at each receiver so that each receiver can dynamically determine its own retransmission nodes.

In each RTP multicast session, each receiver in the multicast group can easily measure the RTT (Round Trip Time) between itself and a particular original sender, as well as the packet loss ratio of the data it receives from that sender. Only the very recent RTT and packet loss ratio that reflect the current network condition are of relevant. Let us define RTT_i^j as the RTT between sender *j* and receiver *i* as measured by receiver *i*, and PLR_i^j as the packet loss ratio (PLR) observed by receiver *i* for the data sent by sender *j*. As mentioned previously, each receiver will periodically probe other receivers about their receiving statistics with respect to the original sender. Once a node receives a probing packet, it will send its own measured RTT and packet loss ratio with respect to the original sender back to the probing node. For example, if node 1 receives a probing packet from node 2 for sender 3, node 1 will send RTT_1^3 and PLR_1^3 to node 2. After receiving all the probing feedback, the node that sent out the probing can determine a good retransmission node. The best retransmission node is chosen using the following strategy.

For node *k*, the retransmission node for sender *j* is

$$RTX_k^j = \arg\min_{i=1,2,3\ldots \quad i\neq k} \left(\alpha \frac{RTT_k^i}{RTT_k^j} + \beta \frac{PLR_i^j}{PLR_k^j} \right) \Gamma(1/G_k^i) \tag{1}$$

s.t. α+β = 1, $\frac{RTT_k^i}{RTT_k^j} \leq 1$ and $\frac{PLR_i^j}{PLR_k^j} \leq 1$.

where RTX_k^j is the retransmission node chosen by node *k* for sender *j*, $\frac{RTT_k^i}{RTT_k^j}$ and $\frac{PLR_i^j}{PLR_k^j}$ are normalized RTT and PLR respectively for node *i* with respect to sender *j*, which are not greater than 1 as nodes with larger RTT or *PLR* are not considered as good candidates, and α, β are the weighting factors. G_k^i is the success rate of retransmtting packets from node *i* to node *k*, which is initially set to 1. After node *i* is chosen by node *k* as the retransmission node, G_k^i will reflect the quality of the path between node *i* and node *k* and also reflect how well node *i* receives data from the original sender. The choice of the function Γ will be determined by the system design. We use the linear function as the Γ function in our simulations. If G_k^i is low, it means that either node *i* often does not have the data that node *k* requests, i.e., it probably also has a packet loss problem, or the path between node *i* and *k* is bad such that the retransmitted data from node *i* can not reach node *k*. In fact, G_k^i is more important in the decision process than the normalized RTT and PLR. In some cases, two nodes that are physically very close to each other and are in the low level of the multicast tree will probably face the same congestion problem. We need to be able to avoid these nodes being retransmission node of each other. Using Eq. (1), this problem can be successfully addressed very quickly. Because the decision will be frequently adjusted based on the feedback from others including the performance of the selected retransmission node, it can address any potential faulty decision and find better retransmission nodes quickly.

5 SIMULATION RESULTS

We evaluate the performance of the proposed path-diversity overlay retransmission mechanism, and compare it to the traditional approach in which the retransmission is done only by the original sender through unicast. The weights α and β in Eq. (1) are chosen to be 0.3 and 0.7, respectively. We choose larger β than α because when choosing a retransmission node, a lower packet loss rate is more important than the RTT. The probing is done every 10 s, representing a very small overhead.

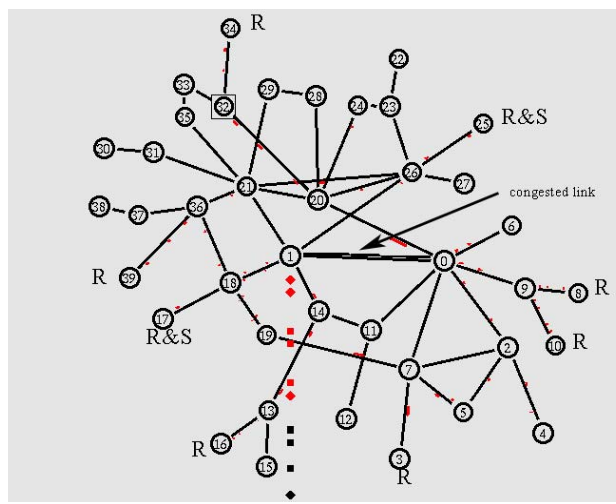

Fig. 2: One of the test topologies with 40 nodes. "R" and "S" represent receiver and sender, respectively. "Falling dots" below node 1 represent packet dropping at node 1 (for both RTP packets and background traffic packets). Other small dots on some of the links are multicast RTP data packets.

We use the *network simulator2* (ns2) [8] with some add-on codes to implement our protocol. The test topology is a transit-stub graph created by Georgia Tech gt-itm network topology generator [1]. Figure 2 shows one of the test topologies. There are 40 nodes including the routers. Most of the link capacity is 1 Mbps and some of them are 2 Mbps. The propagation delays of the links range from 5 ms to 50 ms. The queue in the router can buffer 50 packets by default. The packet drop policy used in the routers is Droptail. In our tests, there are eight nodes joining the multicast group and two of them are data senders. In particular, node 17 and node 25 both send data to the multicast group, and therefore act as both a sender and a receiver. Other active nodes, i.e., nodes 3, 8, 10, 16, 34, 39, serve only as receivers.

Each of the two senders, node 17 and node 25, sends a CBR (constant bit rate) video to the multicast group. The bit rate is 400 kbps. The delay constraint for the retransmission is set to 1 second, and the late packets will be treated as lost packets. We only consider the case of performing only one retransmission per lost packet. From the 11th second, there is a CBR (about 1.8 Mbps) background traffic on the link between node 1 and node 0 that has a bandwidth of 2 Mbps, which is a main path in both of the two sender-based multicast trees. For node 8, the path from sender 17 in the multicast tree is 17→18→1→0→9→8, which includes the congested link. For node 10, the path is the same except the final link. But for node 3, the multicast path from node 17 is 17→19→7→3, which does not include the congested link.

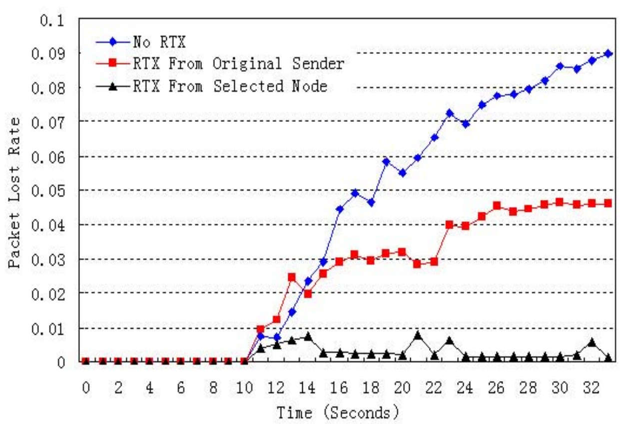

Fig. 3: Packet loss ratio of node 8 for the conventional retransmission and the path-diversity overlay retransmission.

Figure 3 shows the packet loss ratio of node 8 for three different cases, i.e., without retransmission, the conventional retransmission by the original sender, and the proposed overlay retransmission. The congestion occurs at about 11th second when the background traffic starts to take effect. We can see that without retransmission, the packet loss ratio keeps increasing to around 10%. With the conventional retransmission approach, the packet loss ratio decreases to some extent, but is still as high as 5%. Note that at the very beginning of the congestion (around 11-13 s), the original sender based retransmission has slightly higher packet loss ratio than the case without retransmission. This is because the retransmission packets further deteriorate the congestion and cause the loss of more regular packets, at the same time some of the retransmission packets may not have arrived at the measurement time (counted as lost in the measurement). With our proposed overlay retransmission approach, we can see that the packet loss ratio is reduced to almost 0 (only 0.2% on average). We observe a small peak at the beginning of the congestion. This is because initially node 8 took node 10 as a good retransmission candidate because prior to congestion, node 10 was in an excellent receiving condition and is close to node 8. Then in a very short time, node 8 was able to detect that node 10 had problem too, thus changing the retransmission node to node 3 who was really in a good receiving condition. This demonstrates the adaptability of our algorithm in choosing the best retransmission node.

We also observe that when the retransmission is from the original sender, the average delay for retransmission is 322 ms. With the overlay retransmission, the average delay for retransmission is reduced to 121 ms. In summary, the simulation results show that our proposed retransmission architecture can significantly decrease the packet loss ratio, at the same time significantly reduce the retransmission delay.

6 CONCLUSION

We introduce a novel path-diversity overlay retransmission architecture. Initial study shows that it can significantly improve the reliability, delay, and scalability of IP-multicast based multimedia applications. It is an innovative exploitation of peer-to-peer collaboration and resource sharing to address the QoS of multimedia applications. Future work will include more quantitative evaluation of the impact of the proposed framework on the quality of the multimedia applications, as well as a thorough investigation on the optimized construction of the overlay network. The proposed architecture also has the potential to be extended to bridge IP multicast networks and non- IP multicast networks where nodes in non- IP multicast networks can dynamically identify a “forwarding” node that will forward the packets it receives.

Acknowledgement: This work is supported, in part, by Microsoft Research, and in part, by NSF grant CNS-0423386.

7 REFERENCES

[1] J. Ott, at al, “Extended RTP profile for RTCP-based feedback,” draft-ietf-avt-rtcp-feedback-04.txt, Sept. 2002.

[2] J. Rey, et al. “RTP retransmission payload format,” IETF Internet Draft, Jan. 2004. http://www.ietf.org/internet-drafts/draft-ietf-avt-rtp-retransmission-10.txt.

[3] M. Sullivan, and W. Zeng, “A protocol for simultaneous real time playback and full quality storage of streaming media,” in *IEEE Inter. Confer. Communication*, Seoul, May, 2005.

[4] Internet-2 homepage: http://www.internet2.edu/.

[5] J. Beavers, et al, “The learning experience project: enabling collaborative learning with ConferenceXP,” Technical Report MSR-TR-2004-42, Microsoft Research, April 2004.

[6] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, “Distributing streaming media content using cooperative networking,” *ACM NOSSDAV*, Miami Beach, FL, May 2002.

[7] J. Li, “PeerStreaming: A Practical Receiver-Driven Peer-to-Peer Media Streaming System,” Technical Report MSR-TR-2004-101, Microsoft Research, 2004.

[8] The network simulator ns-2, http://www.isi.edu/nsnam/ns/.

[9] E. W. Zegura, “*GT-ITM: Georgia tech internetwork topology models”* http://www.cc.gatech.edu/fac/Ellen.Zegura/gtitm/gtitm.tar.gz, 1996.