A NAK Suppression Scheme for Group Communications Considering the Spatial Locality of Packet Losses

Jinsuk Baek,[†] *and Munene W Kanampiu* [††],

Winston-Salem State University, Winston-Salem, NC, USA

Summary
Today's extensive use of multicast communication demands efficiency, scalability, and reliability. Numerous schemes have been proposed to improve these critical areas and some progress has been realized. Among the most recent and effective schemes that address this issue include the tree-based NAK suppression scheme that emphasizes on reducing NAK implosion at the repair node. The procedure employed would be an optimal solution, if in fact, for every multicast session transmission, at least one receiver node is guaranteed to receive the packet. However, the largest portion of packet loss in multicasting is due to buffer overflow and as a result there is significant spatial locality of packet losses among nodes of a local group. We propose a NAK suppression scheme that considers this property. Compared to the existing schemes, the proposed scheme will significantly reduce traffic congestion, NAK implosion at repair nodes, and the overall error recovery delay. While the issue of traffic congestion reduction is obvious when receiver nodes do not have to make unfruitful attempts to acquire retransmissions from other local group members that are already starved of the same packets, the much reduced NAK implosion at repair nodes and error recovery delay are clearly illustrated in our simulation results.

Key words:
Reliable Multicast, Spatial Locality, NAK Suppression, Error Recovery, and Implosion.

Introduction

A growing number of distributed applications require a sender to transmit the same data to a large group of receivers. These applications include bulk data transfer, distance learning with streaming continuous media, video-conferencing with shared data applications, data feeds, Internet TV, Web cache update, and distributed interactive gaming.
These applications fall in the category of group communication as opposed to traditional one-to-one communication. Multicasting, that is the delivery of a single message to multiple recipients using the same IP address, is the only efficient scalable solution to support these kinds of applications [5]. It requires a high level error free transmission and fast recovery mechanism for each client node. To achieve this, multicasting commands a high level buffer management, optimal packet discarding and retransmission schemes. Although research has focused on these three major areas, error-free transmission problem still looms.
The standard method of providing reliable transmission of data is by employing positive acknowledgements (ACKs) for every successfully received packet. It involves requiring each receiver node to send an ACK for each packet that it has successfully received. The sender keeps track of these ACKs and retransmits all packets that have not been properly acknowledged within a given time window. TCP [15]–[16] is a well-known protocol that uses positive ACKs to provide reliable unicast transmission. But the same approach fails when applied to reliable multicast because of the ACK implosion [2]–[6], [12, 15, 21] it creates. Since each receiver has to acknowledge each packet it has correctly received, the sender's ability to handle these ACKs limits the number of nodes participating in a reliable multicast session therefore compromising scalability.
NAK-based schemes provide a more scalable solution because receivers only contact the sender when they have not correctly received a packet. Scalable Reliable Multicast (SRM) [9] is a well-known NAK-based multicast protocol that guarantees out-of-order reliable delivery using NAKs from receivers. Every time a receiver detects a lost packet, it multicasts a NAK message to all participants in the multicast group. Unlike the regular NAK [22] where each individual receiver independently sends a NAK to the sender as soon as it detects a loss, the SRM protocol requires that each receiver use a randomized NAK-Timer to send the NAK to the sender. This allows the nearest receiver among those that successfully received the packet to retransmit it by multicasting. This is a very common NAK suppression scheme. Unfortunately, this requires all receiver nodes to indefinitely retain all of their successfully received packets for eventual retransmissions, consequently leading to poor buffer management.
Among the many protocols [1]–[6], [9]–[10], [12], [15]–[17], [21] that have been proposed to solve this problem, the tree-based protocols [2]–[6], [12, 15, 21] have emerged to be the most efficient in scalability and reliability. They construct a logical tree at the transport layer for error recovery. This logical tree comprises of three types of nodes: a sender node, repair nodes, and receiver nodes. The sender node is the root of the tree and controls the overall tree construction. Each repair node maintains in its buffer all the packets it has recently received and performs local error recovery for all its group

Manuscript received October 5, 2006.
Manuscript revised October 25, 2006.

member nodes. As a result, tree-based protocols achieve scalability by distributing the server retransmission workload among the repair nodes.

In the non-tree-based protocols [1], [9]–[10], [16]–[17], every individual receiver is responsible for its retransmission requests to the sender. The consequence of this is a NAK implosion at the sender. Tree-based protocols have avoided this problem by assigning a repair node in every group of receiver nodes to do the retransmission requests necessary in the local group. This, however, creates not only a NAK implosion at the repair node but also a buffer management issue at these repair nodes. For the repair node to efficiently handle all of its receiver node's retransmission demands, it must be able to keep for some time all the received packets handy for eventual retransmission requests. On the other hand, it must at the right time be able to discard any unneeded packets in order to optimize its buffer management. This critical section issue has led to further research demands to improve the widely accepted tree-based protocols. In a bid to solve this problem, some proposed schemes [4]–[6] have employed a combination of both the positive and negative acknowledgments and some success has been achieved.

A typical tree construction scheme [2]–[3], [12, 14] mimics a routing tree-like logical tree in which each receiver node uses a TTL distance value to select its repair node. As a result, all group member nodes are likely to branch off from the same router. In the existing tree-based schemes, a receiver node immediately requests from its repair node a retransmission of a lost packet as soon as the loss is detected. These schemes do not consider the impact of spatial locality of packet losses among nodes. When some degree of spatial locality is considered, there exists, in every group session, a chance that a packet loss experienced by a receiver node will also be experienced by the other group members including the repair node whenever all under the same router. The consequences of omitting this fact have resulted in extended error recovery delays caused by receiver nodes having to contact the sender node for packet loss retransmissions since their repair nodes suffer the same packet deprivation. In turn this has also rendered these schemes inefficient both in scalability and reliability.

To reduce NAK implosion at the repair node, the existing solution is the conventional NAK suppression scheme. Here, instead of the repair node, a receiver node retransmits the packet to receiver nodes that broadcast a NAK for the packet within the local group. But we observe that this would only work well if it was guaranteed that for every transmission at least one receiver node in the local group would successfully receive the packet. But considering spatial locality of packet losses among nodes of a group in tree based protocols, these losses are not independent events but are strongly correlated. As such the conventional NAK suppression scheme is viable but not an optimal solution.

We propose a new and improved NAK suppression scheme that considers spatial locality of packet losses in multicasting. The proposed scheme will start with exploring possible packet loss scenarios that were overlooked in the existing tree-based schemes. Under the proposed scheme, if the repair node detects a packet loss, it will multicast an extended-NAK suppression (ENAK_SUPP) message to all its local group members. This ENAK_SUPP value for each individual receiver node will be dynamically evaluated and must be sufficient to allow a complete packet retransmission process for that packet by the repair node to the receiver. Simultaneously the repair node will request a retransmission of that packet to the original sender. On the other hand, if a receiver node detects a packet loss and does not receive an ENAK _SUPP message from its repair node, it will multicast an ENAK_SUPP message to its group members including the repair node. If the member already has the packet or the ENAK_SUPP message it should ignore the new suppression message. Otherwise the member will honor it and therefore refrain from sending the NAK to the repair node. If the repair node receives an ENAK_SUPP message for a packet that it has successfully received, it will multicast the packet to all its group members. In the rare event that a receiver node receives a duplicate packet it should ignore it. Our scheme's advantages over the existing schemes include an increased repair-receivers ratio due to reduced feedbacks from receivers, and faster error recovery since a repair node will usually be located above its receivers on the logical tree hierarchy. The latter also holds true when repair node and its receiver nodes are on the same level since our individual node NAK-SUPP evaluation ensures that a NAK suppression message from the repair node will reach each receiver node promptly. As a result, the proposed scheme will be able to implement a more scalable repair server that ensures optimal multicast service to all its receiver nodes.

The outline of the rest of this paper is as follows. In Chapter 2, we will briefly introduce the existing reliable multicast protocols and corresponding schemes focusing on reliability and scalability. In Chapter 3, we will describe the details of the proposed scheme and in Chapter 4 we will analyze and compare the performance of the proposed scheme to that of the existing ones. We will conclude the paper in Chapter 5.

2. Related Work

Although all multicast protocols strive for the same goals, they essentially differ in how they choose the member node(s) that will buffer packets for the group for eventual retransmission needs and also how long to optimally retain these packets without compromising buffer space.

One of the well known reliable multicast protocols is the *Scalable Reliable Multicast* (SRM) [9]. In this non-tree-based protocol, all receivers use a NAK suppression scheme that ensures out-of-order reliable delivery. When a receiver node experiences a packet loss, it multicasts a NAK to all the participants of the multicast group. This allows the nearest receiver that successfully received the packet to retransmit it by multicasting it to all its neighbors. The result of this is a distribution of error recovery duties to all receiver nodes in the multicast

session instead of leaving the entire workload to the sender node. The drawback of this, however, is that it requires all receiver nodes to retain all their packets in their buffer for eventual retransmission requests. It also results in packet exposure, a case of duplicated packets for the receiver nodes that received the packet successfully the first time. This phenomenon has the consequences of increased bandwidth consumption and Internet traffic.

Reliable Multicast Transport Protocol (RMTP) [15, 21], the first tree-based reliable multicast protocol, employs the construction of a physical tree of the network layer. For each local region, it selects a *Designated Receiver* (DR) that will be responsible for error recovery for all the other receivers in the region. Instead of sending an ACK for every packet received, a process that causes ACK implosion at the designated node, each receiver periodically unicasts an ACK to the designated receiver. The periodic ACK bears the highest packet number that the receiver has successfully received as well as the packet number that this receiver needs a retransmission for. The drawback of this periodic feedback policy, however, is a significantly increased error recovery delay since the receivers do not immediately request for retransmission for a lost packet as soon as the loss is detected. This renders RMTP unfavorable in time-sensitive multimedia data applications. Moreover, since RMTP stores the whole multicast session data in the secondary memory of the DR for retransmission, it makes it unfavorable for transfers of large amounts of data.

The stability detection Algorithm proposed by Gou [11] organizes receiver nodes into groups where they collectively take part in error recovery. These receivers exchange history information periodically about their ACKs and when one of them becomes aware that all the others have successfully received a particular packet, then these receivers can discard the packet. Noticeably this scheme has a disadvantage of a high traffic overhead when it is applied to large number of message exchanges.

The *Bimodal Multicast Protocol* (BMP) [7] employs a buffer management policy where each group member receiver buffers received packets for a certain amount of time. To enhance the effectiveness of BMP, the *Randomized Reliable Multicast Protocol* (RRMP) [22] was proposed. Unlike the original BMP, RRMP carries out the buffering in two separate phases: feedback based short-term and randomized long-term buffering. In the feedback short-term buffering, every member that receives a packet buffers it for a short timed period for eventual retransmission requests in its group. After the elapse of this time, only a small randomly selected number of receivers will continue to buffer this packet. The inefficiency of RRMP is that random selection of the long-term buffers could render it difficult and time consuming for a client receiver to trace a long term buffer especially in cases of large number of participants.

Another protocol that was proposed to solve this problem is the search party protocol [8]. In this protocol, each member of a group discards the packet after a certain amount of time. The unresolved dilemma of the search party protocol remains how to calibrate the optimal discarding time interval.

A recent proposal has suggested two buffer management schemes: The first scheme [4] suggests that each receiver node send NAKs to its repair node for all its packet retransmission needs. At the same time this receiver also periodically sends randomized sequence numbered ACKs back to the repair node to signal for a safe discard from this repair node, which in turn observes the least packet sequence number among these packets and discards all packets with a sequence number below this minimum number. But still not explored here is the optimal ACK transmission interval for both the repair and the receiver nodes. The second scheme [5] suggests for the repair node to discard some packets by considering the ACKs from the most unreliable receivers. In both schemes, there is a reduction in error recovery delay since a request for a packet from the repair node is almost always satisfied. Both schemes also minimize the number of repair nodes needed per number of receiver nodes by minimizing ACK implosion. It goes without noticing, however, that neither of these schemes addresses how to solve the NAK implosion problem.

The only current alternative is to apply the conventional NAK suppression scheme proposed in SRM, to each local group. But the scheme would only work well if it were guaranteed that at least one receiver node in the local group would successfully receive the packet. This case can never be guaranteed especially when considering spatial locality of packet losses in multicasting.

3. Proposed Scheme

Most researches have addressed NAK suppression assuming a general scenario where all the receiver nodes and their associated repair node are attached to a common router. As a result, they have focused on single level NAK suppression approach. However, our research has observed that due to the hierarchical nature of a logical tree structure, a NAK suppression in tree-based multicast protocols must also consider the hierarchical multi-level routers and nodes distribution where depending on the tree level at which a buffer overflow occurs, packet loss can be experienced not only by those nodes immediately under this router but also this loss can propagate to any level of the logical tree depending on the tree's linkage behavior.

Due to this omission, these schemes do not deliver optimal throughput. In lieu of this observation, we propose a scheme that will optimize NAK suppression in tree-based multicast protocols by first addressing the various case scenarios that lead to and the implication resulting from buffer overflow and link errors in a tree-based multicast. We will then present a solution based on these findings.

The proposed scheme explores carefully these cases separately.

In the first case, the repair node and its receiver nodes are under completely different routers. For example, a repair node is under router r_1 while all receiver nodes are under router r_n, where n is greater than 1. This is depicted in Fig. 1.(a). The second case is where the repair node and its receiver nodes are under the same router. For example,

repair node and its receivers are under router r_1, a case possible in a LAN environment. This is shown in Figure 1.(b). The third case involves the repair node and some receiver nodes being under same router r_1 while some other receiver nodes are under a different router r_n such that n is greater than 1 as shown in Figure 1.(c).

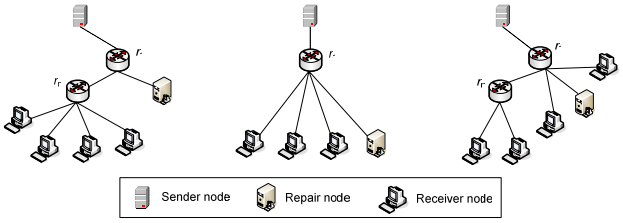

Fig. 1 Node distribution.

Looking at the above three cases, an error can almost always occur from two reasons. First, when a router of a level experiences a buffer overflow, the router will drop some packets from its buffer to make available buffer space. As a result, the repair node or receiver nodes under the router will experience this packet loss. Secondly when there is a link error. Of these two, buffer overflow is responsible for the most packet errors.

Buffer overflow occurs when a sender attempts to stuff more data packets into a router's buffer than it can hold. The excess data packets get dropped and therefore never reach their destination receivers. A link failure typically appears a period of consecutive packet loss that can last for many seconds, followed by a change in delay after the link is re-established. Link failure can be caused by equipment problems (e.g. a failed "blade" in a switch or router, power failure and so on), a cable being unplugged or cut, a configuration change in the transport network or potentially a denial of service attack. Routers are generally intelligent enough to recognize a link failure and find an alternative route. Link failure result in significant gaps in received message. It is unlikely that link failures will occur frequently however they could potentially last for several seconds. Regular occurrence could be symptomatic of equipment or power supply reliability problems. By use of trace-routes the point at which link failures are occurring can be determined and corrected or avoided all together.

Due to today's availability of high quality link medium, link error is a negligible occurrence. One the contrary, due to today's high congestion in network data transfer needs, buffer overflow is a problem to reckon with and as such although the proposed scheme will address link error shortcomings, it will primarily focus on the bigger problem of buffer overflow.

Referring to the node distribution diagrams above, the following observations can be made:

- In all three cases, if the buffer of router r_1 suffers from overflow, then all the nodes will experience packet loss.
- In case (a), if router r_1 successfully receive the packet but router r_n experience buffer overflow, then the repair node will receive the packet, but all the receiver nodes will experience packet loss while in case (c), the repair node and some receiver nodes will receive the packet but those nodes under router r_n will experience loss of the packet.
- In all three cases, if there is no buffer overflow, but some isolated link error occurs, then those nodes whose link have no error will receive the packet but those nodes that are connected by the error link will not.
- In the case of case (a), a link error between router r_1 and the repair node coupled with a link error between router r_n and some receivers would render some of the receiver nodes under the router r_n to request for retransmission from the already deficient repair node. Since the repair node will not be able to satisfy these request, this will result in retransmission request having to be redirected to the upstream node, the recipe not only for more NAK implosion at the upstream node but also an elongated error recovery delay.

Our basic mechanism is as follows. If the repair node detects the packet loss, it will broadcast *Extended NAK Suppression* (ENAK_SUPP) message for the packet to all its group members. Simultaneously, it will request a retransmission of that packet to the original sender node rather than upstream repair node. In this case, the proposed scheme will avoid the existing scheme fault of always contacting the adjacent upstream repair node. By contacting the root sender directly, a repair node in the proposed scheme will be able to bypass not only same level link error packet loss occurrences but also the worst case where the router adjacent to the root experiences a buffer overflow.

If the receiver node detects a packet loss, but did not receive an ENAK_SUPP message for the packet, it will multicast an ENAK_SUPP message for this packet to its group members including the repair node. Other receiver nodes will not send any NAK for this packet. When the repair node receives the ENAK_SUPP message of a packet that it already has, it will multicast the packet to group members. Worth noting here is that in some cases, where all nodes are under the same router (in case of case (b)) and the link speed of the receiver node is faster than that of the repair node, the receiver nodes are bound to send NAK to the repair node before the ENAK_SUPP message from the repair node reaches them. The result of this being that the receiver node will receive the ENAK_SUPP message from the repair node after they have already sent a NAK to this repair node.

Our solution to this is to employ a random NAK sending delay for the receiver nodes to ensure that they always send NAK messages later than their ENAK_SUPP message reaches them from the repair node. This will guarantee minimal error recovery since the random NAK timer ensures that the ENAK_SUPP message from the repair node reaches receiver nodes not later than even the fastest node in the session can initiate a NAK message bound for this repair node. In order to define the random NAK timer delay for each receiver node, we make the following assumptions:

1. The logical tree is already constructed using a well known tree construction scheme [2]–[3], [12, 14] before the multicast session begins.
2. T_s is the multicast session period.
3. The repair node for every local group of receiver nodes is strategically allocated.
4. There are *n* receiver nodes in a local group.
5. The repair node has packet retransmission responsibility for *n* receiver nodes.
6. Each receiver node *i* has well defined its independent NAK TIMER$_i$ satisfying the following condition.

   $$\text{NAK\_TIMER}_i > \max\{\text{OTT}_{<s,\,i,\,t>} \mid 1 \le i \le n \text{ and } 0 < t < \text{Ts}\}.$$

   where, $\text{OTT}_{<s,\,i,\,t>}$ is one-way transit time between the sender *s* and receiver node *i* at time *t*.
7. The repair node also has well defined NAK_TIMER$_{rp}$, satisfying the following condition.

   $$\text{NAK\_TIMER}_{rp} > \max\{\text{OTT}_{<s,\,rp,\,t>} \mid 0 < t < \text{Ts}\}.$$

   where, $\text{OTT}_{<s,\,rp,\,t>}$ is one-way transit time between the sender *s* and repair node *rp* at time *t*.

The random NAK_TIMER value information for each receiver in the local multicast session is pre-registered with its associated repair node. In order to do that, each receiver node has to report its NAK_TIMER value to its repair node. The repair node periodically evaluates the $\text{OTT}_{<rp,\,i>}$ between itself and receiver node *i* via adjoining router. Having evaluated the $\text{OTT}_{<rp,\,i>}$, the repair node will then use it to compute the a priori delay (XNAK_TIMER) needed for this receiver node by subtracting the NAK_TIMER value of this receiver node from the sum of the repair node's NAK_TIMER and the evaluated $\text{OTT}_{<rp,\,i>}$. This process will obey the formula

$$\text{XNAK\_TIMER}_i = \text{NAK\_TIMER}_{rp} + (\text{OTT}_{<rp,\,i>}) - \text{NAK\_TIMER}_i \quad (1)$$

By applying the above formula, each receiver node will now evaluate its own threshold delay (DNAK_TIMER) after which it can send its NAK to the repair node. This is the time it takes for the repair node to establish a complete ENAK_SUPP message to this receiver node. It ensures that each receiver node in the group will receive an ENAK_SUPP message from the repair node before this receiver node initiate its own NAK message for the repair node. This avoid cross messaging between the receiver node and the repair node. This DNAK_TIMER value is given by

$$\text{DNAK\_TIMER}_i = \text{NAK\_TIMER}_i + \text{XNAK\_TIMER}_i. \quad (2)$$

As a result, the proposed scheme will guarantee that no receiver in the session, including the fastest one, will send a NAK message for a lost packet to the repair node before the repair node has monitored and evaluated the situation on the packet and has enough time to multicast this situation to all its session client nodes. This will dramatically reduce unnecessary traffic as well as repair node NAK implosion from its client receiver nodes.

The proposed scheme observes the fact that all nodes are not of homogeneous behavior. At any given time of the dynamic tree construction, a receiver node of a local group session can be rendered out of pace by new adjoining members or the existing mutating ones. For example a new or mutated node could render a neighboring member node's TTL value or loss probability very high or vise versa comparatively.

The effects of a significantly large discrepancy in performance of a local group receiver node will limit the performance of the entire group session. For example, in the proposed scheme, before the repair node can discard any packet it has to ensure that all receiver nodes have received it, therefore, an extraordinarily long OTT or high loss probability receiver node in the group will significantly delay the entire local group's throughput, a feature not considered in the previous schemes.

The solution to this problem would be to introduce the *Candidate Node Threshold Value* (CNTV) algorithm that can be implemented in the dynamic tree construction to enable only those nodes of closely related characteristics to bond together in the existing dynamic tree local grouping mechanism. CNTV will also employ a frequent random check to ensure this rule is obeyed by all members by relocating to the appropriate local group any member that mutates away from the threshold value and that new dynamic joins fit within and adhere to the regulations. This will result in local groups of receiver nodes that consist of almost similar characteristics in their OTT and loss probability. This will not only improve the error recovery for the group region but also significantly improve the repair node's buffer management due to a reduced packet retention time. To calculate the repair node's retention time, RET_TIMER$_{rp}$, the proposed scheme would use the NAK suppression formula as follows:

$$\text{RET\_TIMER}_{rp} = \text{NAK\_SUPP}_{SR} + \text{OTT}_{<SR,\,rp>}, \quad (3)$$

where, *SR* is the selected receiver node with the largest NAK suppression timer value in the group; NAK_SUPP$_{SR}$ is NAK suppression timer value of SR; and $\text{OTT}_{<SR,\,rp>}$ is One-way transit time from SR to its repair node.

Although the proposed scheme will produce better results compared to the existing ones, it has some limitations in some specific cases. The first case is where the repair node and some receiver nodes successfully receive the packet but other receiver nodes experience a loss of this packet. Although the packet is available at the repair node and could be retransmitted right away, these receivers will still have to obey their timer value wait-time before they can send their NAK to the repair node. The result of this is unnecessary delay in the error recovery. The second scenario is where all the receivers of a local area successfully receive the packet but their associated repair node suffers a loss for that packet. The repair node will multicast an ENAK_SUPP message to these receivers although they successfully received the packet. This will lead to duplicating of packets, and in so doing it will also cause unnecessary traffic. We can reduce this problem by implementing a "Overlook and Ignore protocol" where any node that successfully receives a packet overlooks and

ignores any attempts by other nodes to communicate suppression messages pertaining to this packet. Our research notes, however, that these scenarios are rare and their probability in tree-based protocols is negligible because there is significant spatial locality of packet loss among members of a group.

4. Performance

In this section, we analyze and compare the performance of the proposed ENAK_SUPP scheme to both the pure NAK-based scheme where all receiver nodes immediately transmit NAKS to their repair node as soon as they detect a packet loss, and the NAK-based SRM scheme where receiver nodes use a delay timer value to transmit their NAK for a lost packet.

Fig. 1 shows the network topology we use in our analysis. We assume there are N_{rp} repair nodes in a multicast session and the repair nodes are pre-determined. This is the standard hypothesis made by all tree construction schemes [2], [3], [12]–[14]. In a tree-based multicast, in order to construct logical repair tree, each receiver node actively finds the candidate repair node that is available in the session for its error recovery. Each receiver node selects and binds to this repair node, usually with the shortest TTL distance among the candidate repair nodes. In our analysis, we also make the following assumptions:

- There are n receiver nodes for every repair node. Hence, the repair node is responsible for handling NAKs from its n receiver nodes.
- Each of the n receiver nodes has an independent packet loss probability $L_{\text{rc\_LE\_}i}$ caused by link error, for $i = 1, 2, \ldots, n$.
- The repair node also has an independent packet loss probability $L_{\text{rp\_LE}}$ caused by link error.
- Each of the n receiver nods and repair node has a packet loss probability $L_{\text{rc\_BO\_}i}$ and $L_{\text{rp\_BO}}$ respectively, caused by router's buffer overflow, for $i = 1, 2, \ldots, n$. Actually, when considering spatial locality, $L_{\text{rc\_BO\_}i}$ is equal to $L_{\text{rp\_BO}}$ as long as all nodes are under the same router.
- Each receiver node i culminates with a packet loss probability $L_{\text{rc\_}i}$ such that

$$L_{\text{rc\_}i} = L_{\text{rc\_BO\_}i} + L_{\text{rc\_LE\_}i}, \text{ for } 1 \le i \le n.$$

- The repair node culminates with a packet loss probability L_{rp} such that

$$L_{\text{rp}} = L_{\text{rp\_BO}} + L_{\text{rp\_LE}}.$$

- The sender node will transmit m number of packets to the multicast group.

4.1 NAK Implosion

The efficiency of a repair node strongly depends on the number of NAKs arriving from its receiver nodes. Therefore to provide scalability it is imperative that we minimize NAK implosion at the repair node.

Under the proposed scheme, a receiver node will not immediately send a NAK to the repair node as soon as it detects a packet loss. Instead, it will delay for a period of its DNAK_TIMER. If after the expiration of this time the receiver node has not received an ENAK_SUPP message from the repair node, it will then multicast an ENAK_SUPP message to its local group members including the repair node. Over a session involving a transmission of m packets from sender, the maximum number of feedbacks to the repair node from its n receiver nodes will obey the inequality

$$F_{\text{NAK\_SUPP}} \le m\sum_{i=1}^{n} L_{rc\_LE\_i} \tag{4}$$

We need to mention that this case scenario is not favorable in the proposed scheme, because it always assumes the worst case scenario that an ENAK_SUPP message from a receiver node always arrives at other receiver nodes after their DNAK_TIMERs have expired.

On the other hand, if we apply the NAK suppression concept of SRM protocol to the local group, the number of feedbacks F_{SRM} can be given by

$$F_{\text{SRM}} \ge mL_{rc\_LE\_f}(1 - \prod_{i=1, i\ne f}^{n} L_{rc\_LE\_i}) + mnL_{rc\_BO}$$

$$= m[L_{rc\_LE\_f}(1 - \prod_{i=1, i\ne f}^{n} L_{rc\_LE\_i}) + nL_{rc\_BO}] \tag{5}$$

In the SRM protocol, there is no network entity acting as a repair node because it is not a tree-based protocol. Hence, the repair node of the tree-based protocol acts as the sender node. When the receiver node f, the one with the fastest NAK_TIMER detects a lost packet, it will multicast NAK for that packet to its local group. This retransmission request will be satisfied as long as there is at least one node in the group that has successfully received the packet. This is only possible when the NAK_TIMERs of all receiver nodes are perfectly and efficiently set. As a result, we can conclude this is the minimum number of feedbacks since we assume this NAK message from the receiver node f will arrive at all receiver nodes in the group before their NAK_TIMERs expire. Otherwise, they will also multicast the same NAK message. Moreover, if we consider the spatial locality of packet losses caused by the router's buffer overflow, the requested packet will always be unavailable at every node of the local group. Consequently, all these receiver nodes will have to send their NAKs to the sender node, akin to the pure NAK scheme. Under this assumption, the number of feedbacks F_{NAK} for a NAK-based scheme, where all receiver nodes simply send their NAKs to the repair node will be given by

$$F_{NAK} = m\sum_{i=1}^{n} L_{rc\_i} \tag{6}$$

In order to observe the number of feedbacks reduced by applying the proposed scheme and compare the results to those of the other schemes, we will look at the number of feedbacks generated in each of these scheme for a given repair node serving a given number of receiver nodes. Our experiments are performed for up to 100 receiver nodes per repair node.

To generate the loss probability of each receiver node, we applied the following formula from [19], where S is the packet-sending rate in packets/sec, $RTT_{<s, rc\_i>}$ is the round trip time from the sender node to receiver node i, and L_{rc_i} is the packet loss probability between the sender node and receiver node i.

$$S = 1.22/(RTT_{<s, rc\_i>} \sqrt{L_{rc\_i}})$$

This assumes that the sender node transmits packets in a TCP-friendly manner and that each node in the multicast session uses the UDP protocol. We set S to 128 packets/second and simulated round-trip times $RTT_{<s, rc\_i>}$ as Poisson random variables, each a having value mean of 100ms. Similarly, the round-trip times $RTT_{<rc\_i, rp>}$ between a receiver node i and its repair node rp are also simulated as Poisson random variables with a mean value of 50ms. Fig. 2 and 3 show our measurements for round-trip times from the sender node to receiver nodes and packet loss probability for 100 receiver nodes.

As packet loss caused by buffer overflow is universal in a local group, $L_{rc\_BO\_i}$ and L_{rp_BO} can be set to a constant, in our case 0.005456. This implies that the overall average loss probability of each receiver node and repair node caused by link error is 0.00384 since 0.005456 buffer overflow loss probability will be universal for every group member due to the spatial locality phenomenal. Fig. 4 shows how many feedbacks the proposed scheme can reduce compared to the other schemes as we increase n.

We assumed that the number of transmitted packets m = 10,000, which roughly represent a transfer of 10 megabytes with a packet size equal to 1 kilobyte. As can be seen from the graph above, for the 100 receiver nodes, the minimum difference in feedbacks between the proposed scheme and the NAK-based scheme is more than 5000 while the minimum difference between the proposed scheme and the SRM scheme is about 1700 feedbacks. We also need to note that these differences increase as we increase the number of receiver nodes. This result indicates that the proposed scheme provides scalability since the repair node can serve more receiver nodes in its local group.

We will now show how the errors caused by buffer overflow affects the number of feedbacks from the receiver nodes in the proposed scheme versus the SRM scheme. We do not need to consider the pure NAK-based scheme here because it does not distinguish between error types but only observes the overall error count.

At start spatial locality is low meaning loss due to buffer overflow is less than loss due to link error. The implication of this is that a smaller portion of the total packets lost by a receiver node will be subject to NAK suppression and the larger portion will be reported for retransmission as soon as the NAK timer of the receiver node expires. This will be the case until when the spatial locality factor is greater than the link error factor. After this point, as the spatial locality factor increases and link error diminishes, the number of feedbacks to the repair node decreases. As the loss due to buffer overflow infinitely increases and the link error decreases towards zero, as it is the case in today's multicasting, the proposed scheme is capable of significantly reducing NAK implosion at the repair node. As we can see in Fig.5, the proposed scheme is better than SRM as long as the spatial locality factor is greater than 50%.

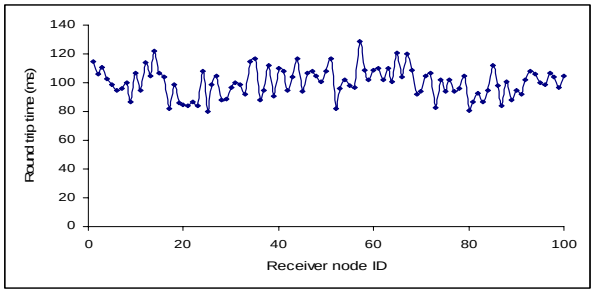

Fig. 2. Simulated round-trip times.

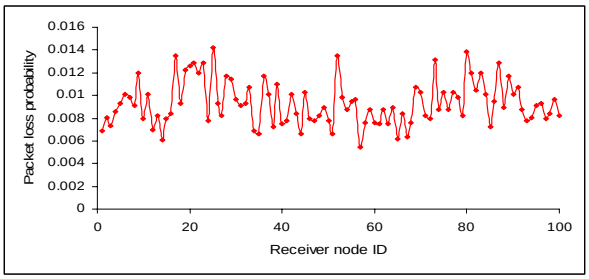

Fig.3. Simulated packet loss probabilities.

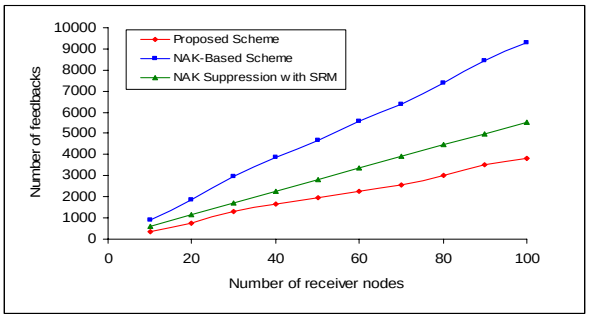

Fig. 4. Difference Δ_{min} vs. the number of receiver nodes per repair node.

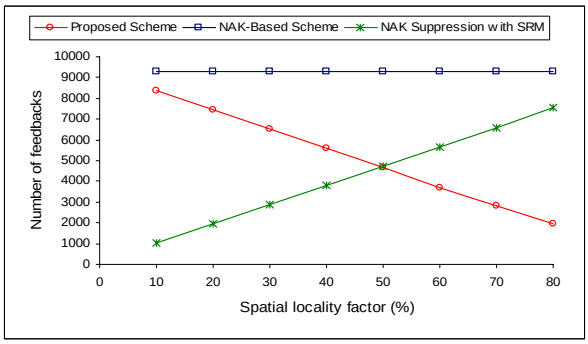

Fig. 5. Number of feedbacks vs. spatial locality of packet losses.

4.2 Error Recovery Time

In this section, we evaluate the error recovery time for each receiver node. Error recovery time represents the time it will take for a receiver node to receive a packet

retransmission after a request to the repair node. The first thing we need to put in consideration is the height of the repair tree. If the requested packet is not available in the local repair node's buffer, the error recovery delay becomes more prominent as the height of the repair tree increases. In order to reach a reasonable evaluation of the error recovery delays in the proposed scheme and the other scheme, we assumed that the height of the repair tree is equal to 2. That is, the tree comprises of a sender node, repair node, and the receiver nodes. In case of incorporating the SRM in a local group of the repair tree, the number of receiver nodes in a local group also affect the error recovery time since a receiver node will first attempt its retransmission requests from its receiver node neighbors before finally sending the request to the repair node. The more the neighbors the longer it will take before the receiver node redirects the request to the repair node. When we consider spatial locality of packet loses, then, in SRM, the receiver nodes will almost always end up redirecting their retransmission requests to the repair node since a packet missed by one receiver node in a local group almost always means the same loss for all the other receivers.

In the case of the pure NAK-based scheme that does not apply any NAK suppression the error recovery time depends largely on the group's spatial locality factor. If we assume the receiver node experiences 10 lost packets for every 100 packets transmitted from the sender node and that of these, 8 are lost due to the router's buffer overflow, this indicates that the repair node will also experience loss of these 8 packets in its buffer. Therefore, the repair node will request for these packets from its upstream repair node. But if the loss was caused by the last level router buffer overflow, this request will end up at the sender node resulting in more error recovery delay.

But in the proposed scheme, as soon as the repair node detects a packet loss, it simultaneously sends a NAK to its upper stream repair node and an ENAK_SUPP message to all its group receiver nodes in order for them to suppress sending a NAK for the packet. This serves the purpose well since, considering spatial locality, most packet losses are caused by buffer overflow and a packet loss experienced by the repair node will also be experienced by the receiver nodes in the group assuming the repair and receiver nodes are all under the same router. In this case, if we use the same example used above, the repair node will be able to immediately retransmit the 2 packets from its buffer to any receiver node that requests for them, assuming the repair and receiver nodes did not experience loss of the same 2 packets, a case that would be rare. The repair node can only retransmit the other 8 packets to the receiver nodes only after it receives a retransmission for them from the sender node.

We can evaluate a receiver node's error recovery time in the proposed scheme by considering two different scenarios, namely (a) loss due to link error, and (b) loss due to buffer overflow.

In the case of a loss due to link error, the situation can be subdivided into two cases. If the repair node has the requested packet in its buffer, it can immediately retransmit that packet to the receiver node obeying the formula,

$$\mathrm{ER}_{\mathrm{NAK\_SUPP\_1}} = L_{\mathrm{rc\_LE}\_i}\,(1 - L_{\mathrm{rp\_LE}})\,\mathrm{RTT}_{<\mathrm{rc,\ rp}>} \qquad (7)$$

If the repair node also did not successfully receive the requested packet, it can resend the packet after it receives a retransmission from the sender node obeying the formula,

$$\mathrm{ER}_{\mathrm{NAK\_SUPP\_2}} = L_{\mathrm{rc\_LE}\_i}\,(L_{\mathrm{rp\_LE}})(\mathrm{RTT}_{<\mathrm{rc,\ rp}>}+\mathrm{RTT}_{<\mathrm{rp,\ s}>}) \qquad (8)$$

In the case of a loss due to buffer overflow, the repair node will simultaneously send an ENAK_SUPP for the lost packets to the group's receiver nodes and at the same time request for the packets from the sender node. The error recovery time can be calculated using the formula,

$$\mathrm{ER}_{\mathrm{NAK\_SUPP\_3}} = L_{\mathrm{rc\_BO}\_i}\,(\mathrm{RTT}_{<\mathrm{rp,\ s}>} + \mathrm{OTT}_{<\mathrm{rp,\ rc}>}) \qquad (9)$$

In NAK-based schemes, the repair node batches NAKs for a packet and retransmits the packet periodically as long as there is a pending NAK for that packet. Let us call the period δ and assume that the packets arrive at a repair node in a Poisson process with a mean arrival rate λ. If the repair node has B buffers, we can define the random variable $N_A(\delta)$ to represent the number of packet arrivals at the repair node within a time interval of length δ. In order to perform at least one retransmission successfully, the following condition should be satisfied:

$$P(N_A(\delta) \geq B) = 1 - \sum_{n=0}^{B-1} \frac{(\lambda\delta)^n e^{-\lambda\delta}}{n!} = 0, \qquad (10)$$

which simplifies into $\sum_{n=0}^{B-1} \frac{(\lambda\delta)^n}{n!} = e^{\lambda\delta}$.

Since we have $e^{\lambda\delta} = \sum_{n=0}^{\infty} \frac{(\lambda\delta)^n}{n!}$, (10) can only be satisfied when B goes to infinity.

Hence, a NAK-based scheme must require the repair nodes to buffer all packets for an infinite amount of time in order to achieve full coverage for all retransmission requests from the receiver nodes.

In NAK-based schemes that use a timer mechanism, the repair nodes discard packets from their buffers after a time interval I without considering whether these packets were successfully received by all their receiver nodes or not. As a result, some packets could be prematurely removed from the repair node buffer while their retransmission requests from some receiver nodes are still pending. In this case, the missing packets will have to be re-sent from either an upstream repair node or the sender node. This is usually so especially in cases where all repair nodes apply the same buffer management policy and therefore discard the same packets at the same time. This not only increases error recovery time for receiver nodes but also generates unnecessary traffic and consequently decreasing the over all Internet performance.

If we assume the packet discarding timer value is optimally set, the repair node does not prematurely discard packets from its buffer. Hence, the only case that the repair node cannot retransmit a requested packet is if it

also did not receive it from the sender node. As such the error recovery time for a receiver node i in a NAK-based scheme with a timer can be given by

$$ER_{NAK} = L_{rc\_LE\_i}\,(1-L_{rp\_LE})\,RTT_{<rp,\,rc>} + (L_{rc\_LE\_i}\,L_{rp\_LE} + L_{rc\_BO\_i})(RTT_{<rp,\,s>} + RTT_{<rp,\,rc>}) \quad (11)$$

In case of incorporating the SRM in the NAK-based scheme in a local group, the requested packet from receiver node i can be retransmitted from any receiver node member of the local group as long as the loss is not due to buffer overflow of the underlying router, and that there is at least one receiver node in the group that successfully received the packet. If the packet loss is due to the group's router buffer overflow, there will be no successful retransmission from any of the other receiver node members in the group since they are all subject to the same loss due to spatial locality. As a result, the requesting receiver node will wait through its NAK timer and then request for the retransmission from the repair node, and since the repair node also suffered the same loss, the receiver node will have to proceed with its request to the sender node. As such the error recovery time can be given by

$$ER_{SRM} = L_{rc\_LE\_i}\,(1-\prod_{j=1,\,j\neq i}^{n} L_{rc\_LE\_j})\,RTT_{<rc\_i,\,rc\_k>} + L_{rc\_BO\_i}\,(R\_NAK\_TIMER_i + RTT_{<rp,\,rc>} + RTT_{<rp,\,s>}) \quad (12)$$

where receiver node k is a receiver node in the local group who received the packet and $R\_NAK\_TIMER_i$ is a random NAK timer of receiver node i.

To evaluate the error recovery time for different schemes, we set $RTT_{<rp,\,s>}$ to 50ms. In case of SRM, since the $R\_NAK\_TIMER_i$ is larger than the NAK_Timer_i of the NAK-based scheme, we set the $R\_NAK\_TIMER_i$ to the sum of NAK_Timer_i and random delay. In our simulation, we generate the random delay of each receiver node between 25ms and 50ms. Fig. 6 shows the error recovery time for each scheme. As we can see, the proposed scheme provides a much faster error recovery than the other schemes. We assume the height of the tree is 2 since in case of the other schemes, if we assumed that there were one or more upstream repair nodes between the repair node and the sender node, it would introduce additional overhead because

a) In case of the NAK-based scheme with packet discarding timer, if the immediate repair node discards a packet and it happens that its upstream repair node also uses the same timer value to discard packets, these two repair nodes will simultaneously carry out a discard for the packet. Therefore, if the requested packet is not available at the immediate repair node, the packet will also not be available at the upstream repair node and hence a request for a retransmission by a receiver node will culminate at the sender node.

b) In case of incorporating SRM in the local group of a tree based scheme, if the packet loss is due to buffer overflow of a router located at a higher level of the tree hierarchy, and this is the router that their upstream repair node is attached to, retransmission attempts from the receiver nodes will continue to every receiver node member at the upper level. Not until every receiver node at this other level has been visited and found not to have the requested packet will any attempts be made to request for the retransmission from the repair node of this new level. Since these upper level receiver nodes likewise suffer the same spatial locality loss as the ones below them, this could amount to a substantial extended recovery delay depending on the height of the repair tree and also the number of receiver nodes per local group.

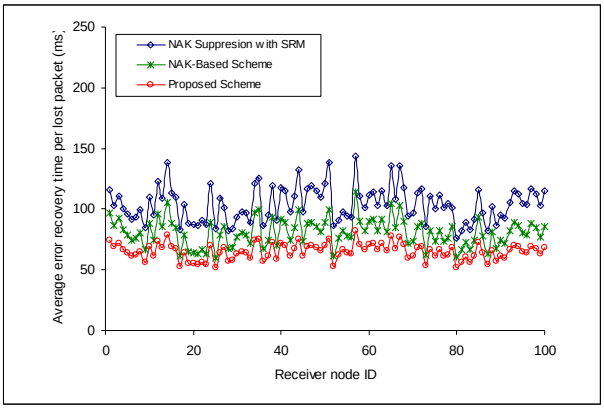

Fig. 6. Difference of error recovery delay vs. the number n of receiver nodes per repair node.

Unlike the previous schemes, the proposed scheme's error recovery delay is not affected by the height of the tree. This is because as soon as a packet loss is detected by the most upstream repair node, this repair node will immediately send an extended suppression message to all its children nodes. As such these children nodes will not send NAKs to their repair nodes. Instead they will delay sending their NAKs until their repair nodes receive a retransmission of the packet and automatically retransmit it to them. The only error recovery time involved therefore will be the sum of the round trip time between the upper most repair node that detected the loss and its repair node (sender), and one way transit time between this repair node and the receiver. The error recovery performance of the proposed scheme therefore is only affected by the total path length between the upper most repair node affected and the receiver node, and not the height of the tree.

5. Conclusion

A big part of packet loss in multicasting is as a result of router's buffer overflow. In the widely accepted tree-based protocols, a router's buffer overflow results in all nodes attached to this router consequently suffering the loss. This spatial locality packet loss phenomenal has not been considered in the existing schemes that have been proposed for efficient multicasting. The consequence of this omission has been traffic congestion, repair node implosion, and extended error recovery times. We have proposed a NAK suppression scheme that considers spatial locality of packet losses in multicasting. The

proposed scheme introduces an Extended NAK suppression aspect whereby receiver nodes delay sending their NAKs until the repair node has had enough time to request and receive retransmission of this packet from the sender node and send it to the receiver node. In the event the loss was due to a link error and that the repair node does not receive the packet but the receiver node does, on receiving the NAK suppression message from the repair node, the receiver node retransmits the packet to this repair node. The result of the proposed scheme is a significant reduction of traffic congestion, NAK implosion at repair node, and error recovery delay compared to the existing schemes.

References

[1] B. Adamson, C. Bormann, M. Handley, J. Macker, "NACK-Oriented Reliable Multicast Protocol (NORM)," IETF draft-ietf-rmt-pi-norm-10, July 2004.

[2] J. Baek, "A Hybrid Configuration of ACK Tree for Multicast Protocol," *Proc, of the* SPECTS 2002, pp. 852–856, July 2002.

[3] J. Baek, E. Lee, "An Improved Logical Tree Construction Scheme for Tree-Based Reliable Multicast", *Proc. of the* ICTS 2003, pp. 110–121, October 2003.

[4] J. Baek, J. F. Pâris, "A Buffer Management Scheme for Tree-Based Reliable Multicast Using Infrequent Acknowledgments," *Proc. of the* IPCCC 2004, pp. 13–20, April 2004.

[5] J. Baek, J. F. Pâris, "A Heuristic Buffer Management and Retransmission Control Scheme for Tree-Based Reliable Multicast," *ETRI Journal*, volume 27, No 1, February 2005.

[6] J. Baek, J. F. Pâris, "An Efficient Retransmission Control Scheme for Tree-Based Reliable Multicast," *Proc. of the* SPECTS 2004, San Jose, California, USA, pp. 145–152, July 2004.

[7] K. P. Birman *et al.*, "Bimodal Multicast," *ACM Transactions on Computer Systems*, 17(2):41–88, May 1999.

[8] M. Costello and S. McCanne, "Search Party: Using Randomcast for Reliable Multicast with Local Recovery," *Proc. of the IEEE ICC*, pp. 1256–1264, March 1999.

[9] S. Floyd et al, "A Reliable Multicast Framework for Lightweight Sessions and Application-Level Framing," *IEEE/ACM Transactions on Networking*, 5(6):784–803, December 1997.

[10] T. Gemmel *et al.*, "The use of Forward Error Correction in Reliable Multicast," IETF draft-ietf-rmt-info-fec-02.txt, October 2002.

[11] K. Guo, and I. Rhee, "Message Stability Detection for Reliable Multicast," *Proc. of the IEEE ICC*, pp. 814–823, March 2000.

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

[13] S. K. Kasera, J. Kurose, and D. Towsley, "Buffer Requirements and Replacement Polices for Multicast Repair Service," *Proc of the* NGC 2000, pp. 5–14, November. 2000.

[14] S. J. Koh et al., "Configuration of ACK Trees for Multicast Transport Protocols," *ETRI Journal*, 23(3):111–120, September 2001.

[15] J. C. Lin and S. Paul, "RMPT: A Reliable Multicast Transport Protocol," *in Proc. of the* INFOCOM 96, pp. 1414–1424, March 1996.

[16] M. Luby, L. Vicisano, "Compact Forward Error Correction (FEC) Schemes," RFC 3695, February 2004.

[17] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft, "Asynchronous Layered Coding (ALC) Protocol Instantiation," RFC 3450, December 2002.

[18] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgement Options," RFC 2018, October 1996.

[19] J. Mahdavi and S.Floyd, "TCP-friendly unicast rate-based flow control," January 1997. http://www.psc.edu/networking/papers/tcp_friendly.html

[20] Ozkasap, R. van Renesse, K. P. Birman, and Z. Xiao, "Efficient Buffering in Reliable Multicast Protocols," *Proc. of the First International Workshop on Networked Group Communication*, pp. 188–203, Nov. 1999.

[21] B. Whetten and G. Taskale, "The Overview of Reliable Multicast Transport Protocol II," *IEEE Networks*, 14(1):37–47, Jan.-Feb. 2000.

[22] Z. Xiao, K. P. Birman, R. Renesse, "Optimizing Buffer Management for Reliable Multicast," *Proc. of the 2002 ICDSN*, pp. 187–202, June 2002.

Acknowledgments

This work is supported in part by the North Carolina Space Research Scholarship under 2006-2007grant.

Jinsuk Baek is Assistant Professor of Computer Science at the Winston-Salem State University (WSSU). He is the director of Network Protocols Group at the WSSU. He received his B.S. and M.S. degrees in Computer Science and Engineering from Hankuk University of Foreign Studies (HUFS) in Yougin, Korea in 1996 and 1998, respectively and his Ph.D. in Computer Science from the University of Houston in 2004. Dr. Baek was a post doctorate research associate of the Distributed Multimedia Research Group at the University of Houston. His research interests include scalable reliable multicast protocols, mobile computing, network security protocols, proxy caching systems, and formal verification of communication protocols. He is a member of the IEEE.

Munene W Kanampiu received his B.S. degree in Computer Science from the Winston-Salem State University (WSSU), Winston-Salem, NC in 2004. He is working for his M.S. degree at the WSSU and a member of the WSSU Network Protocols Group. He was awarded the NC Space Grant Fellowship award in March 2005. His research interests include Scalable Reliable Multicast Protocols and Mobile Computing.