A Dual Mode Buffer for Reliable Multicast in Mobile IP Networks

Jinsuk Baek,[†] and Eunjung Lee[††],

[†]Department of Computer Science, Winston-Salem State University, Winston-Salem, NC, USA
[††] PSF Technology, Inc, Winston-Salem, NC, USA

Summary
In mobile IP networks, a local mobility agent is extended to assist reliable multicast for mobile nodes. Each agent keeps in its buffer all packets that are likely to be requested by any of its mobile nodes. They can provide reliable service by performing a retransmission for the requested packet from its mobile nodes. When a mobile node moves to another subnet, transmission errors tend to happen in bursts, because the two subnets have differential transmission delays from the sender. We propose a new scalable buffer management scheme taking into account this fact. Under our scheme, mobile nodes do not acknowledge for every correctly received packet to their local mobility agent; instead, they send infrequent feedbacks to their agent. The feedback schedule of each mobile node is adaptive to their transmission error patterns. The mobility agents can also perform adaptive packet discarding based on the feedbacks.

Key words:
Mobile IP, Multicast, Mobility agent, Buffer, Retransmission

Introduction

Mobility support is becoming increasingly important due to the widespread use of portable computers and handheld devices such as PDAs and cellular phones. A growing number of mobile services require a sender to distribute the same data to a large group of mobile nodes (MNs). Multicast is an efficient way to support this kind of applications. One of the most difficult issues in end-to-end multicasting is that of providing an error-free transmission mechanism.

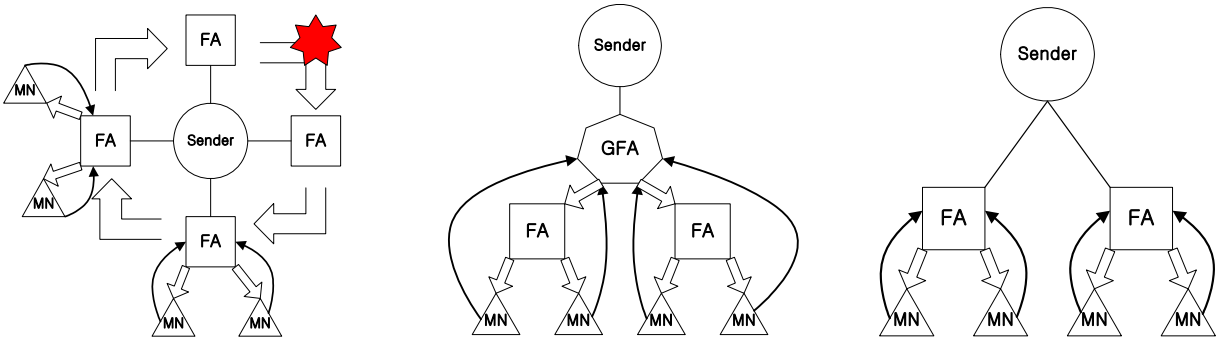
Mobile IP does not guarantee reliable packet delivery as it uses best-efforts transmission mechanism. Hence, the lost or damaged packets have to be retransmitted from sender or other nodes. This process can be performed at the transport layer. In a mobile environment, dedicated agents, such as a Home Agent (HA), Foreign Agent (FA) and Gateway Foreign Agents (GFA), participate in a multicast session for error recovery of their MNs [8, 10]. These agents integrate the status information of their MNs and perform local error recovery for these nodes using the data stored in their buffers.

The dedicated buffer space for a multicast session is limited because the buffer is likely to be used for other multicast sessions or applications for error recovery or other purposes. Hence, the buffer of each agent should be managed in an efficient manner. We also have to reduce bottlenecking such as ACK or NAK implosion at the agent since the number of MNs that can be handled by a single agent will be limited by the agent's ability to handle these feedbacks.

All previous schemes [1, 2, 7, 8, 10] focused on this issue assumed that packet losses were independent events that were not correlated with previous transmission failures. This, however, is not the case for a mobile wireless environment. Instead, packet losses tend to happen in bursts because (a) each MN suffers from "out-of-synch" problem when they perform handoff from one network to another network, and (b) packet losses are much more likely to be occasioned by router buffer overflows. As a result, packet losses are often strongly correlated.

We propose a more efficient buffer management scheme taking advantage of this temporal locality. It assumes that transmission errors tend to happen in bursts separated by long periods of relatively error-free transmission. Our scheme allows the agent's buffer to operate on dual mode (normal mode and error mode). In a normal mode, the agent periodically performs packet discarding by using timeout mechanism. Whenever it receives a NAK from its MN, it will switch to the error mode and stop discarding packets until it has received k consecutive ACKs from the MN. In addition, we found that the previous schemes that have been proposed to counter "out-of-sync" problem introduce duplicate packets at the MN during handoff. Our scheme also includes a solution to this problem.

Our proposal outperforms previous schemes in several aspects. First, we are able to implement more scalable agent, because in a normal mode, each MN sends infrequent ACKs indicating which packets can be safely discarded from the agent's buffer. Second, it provides a reliable solution since most of the requested packets from MNs are available at the agent's buffer for retransmission. As a result, our scheme satisfies the requirements of the mobile applications.



(a) Ring-based approach (b) Hierarchical approach (c) FA-based approach

Fig. 1. Reliable Mobile Multicast Approaches

The remainder of this paper is organized as follows. Section 2 briefly introduces mobile IP multicast and related work. Section 3 describes how our new scheme operates in performing buffer refreshment functionality in mobile IP networks. In Section 4, we analyze the performance of the proposed scheme. Finally, Section 5 contains our conclusions.

2. Related Work

To make continuous network coverage for MNs possible, these MNs should stay connected to the network regardless of their location. This requirement creates a conflict between two mobility supports. First, a MN should change its IP address in order to allow correct packet routing. At the same time, it cannot change its IP address without breaking all its existing connections. Mobile IP solves these mobility problems by using two IP addresses: *permanent home address* assigned at the home network and *temporary care-of address* (CoA) representing the current location of the MN. Whenever a MN obtains the CoA from a foreign network, the binding between its home address and the new CoA should be maintained transparently. There are two specialized routers known as mobility agents that maintain this mobility binding and tunneling; the home agent (HA) in the home network and the foreign agent (FA) in the visited network. Mobile IP supports seamless handoff for MNs within the scope of unicast delivery, through the cooperative support of these two agents.

In Mobile IP networks, multicasting is supported by two different approaches – namely, *Bi-directional Tunneled multicast* (MIP-BT) and *Remote Subscription* (MIP-RS) [5]. In a MIP-BT, the multicast packets, destined to a MN, are first routed to the HA of the MN. The HA encapsulates and delivers the packets to the FA of the MN. However, each packet easily suffers from triangle routing problem. Moreover, it does not scale well, because each MN must always subscribe to the groups of interest through the HA. MIP-RS requires a MN to re-subscribe to the multicast group on the new FA whenever it performs handoff. Hence, it provides a simple implementation, indicating efficient accommodation for the large number of MNs. However, we need to mention that it suffers from "out-of-sync" problem, because the two different FAs have different packet delivery delay from a sender node. When a MN moves to new network and the new FA has a shorter delivery delay than that of the previous FA, the MN experiences packet losses. Hence, the lost packets should be retransmitted from the original sender or other node such as a FA.

In ring-based scheme [7], a token is exchanged among the FAs in some fixed order. Each FA collects ACKs from its MNs. When they receive a token, they agree on the maximum packet sequence number that can be safely discarded from their buffer. As shown in Fig. 1.(a), this global token passing is decentralized and has a high efficiency owing to its simplicity. However, the failure of one FA can crash the entire multicast session. Or if a FA accidentally neglects to release the token, then error recovery procedure must be invoked to get the token back in circulation. Additionally, it should consider the optimal path among the FAs to reduce the error recovery delay.

The hierarchical approaches [1, 2] employ supervisory agents such as GFA. These dedicated supervisory agents perform packet error recovery task for MNs. This is depicted in Fig. 1.(b). These approaches provide seamless data delivery to MNs at handoff, since each MN needs not to change its CoA as long as it is under same GFA, because GFA supports regional registration [5]. The drawback of these approaches is their poor scalability. First, the GFA has to buffer some packets that have not properly acknowledged from its MNs in its network domain. It also has to retransmit the requested packets from those nodes. Second, it needs to keep track of the MNs movements. As a result, the GFA is likely to be a bottleneck point.

A more recent solution [8, 10] is to let the FAs participating in the multicast session perform error recovery task for their MNs. This process is depicted in Fig. 1.(c). These schemes can distribute the error recovery tasks to the end points (FAs) acting as local repair servers for their MNs. The important issue in this approach is how to efficiently manage the FA's buffers and how to reduce bottleneck at the FA due to centralized control. While the schemes focused on this issue were found to be efficient, we need to point out that they did not take advantage of the temporal locality of packet losses as our new scheme does.

3. Proposed Scheme

Our scheme is effectively designed for satisfying requirements of many mobile applications wishing scalable, reliable multicast services assuming a single sever and multiple non-multicast capable MNs. In this chapter, we define how the MNs can receive seamless packets even when they move to another network. Also, we propose an efficient buffer management scheme considering the temporal locality of the packet losses.

3.1 Basic Operations for Mobile IP Multicast

Fig. 2 shows basic operations of the FAs and MN in a Mobile IP environment. When the MN visits a foreign network (Network 2), it registers with the FA_2 by sending a *Registration Request* message. The FA_2 then updates its visitor list and relays the *Registration Reply* message to the MN. The MN now sends a join message to FA_2 for its specific multicast group with its highest packet sequence number, P_{2H}, it has received from FA_1. The FA_2 then sends an IGMP-join message for the multicast group to its first-hop multicast router.

Upon joining the multicast group, the FA_2 allocates some buffer spaces for this multicast group. This buffer is a temporary storage for multicast packets starting from the lowest packet sequence number P_{2L} sent by a sender node. The FA_2 is able to perform error recovery for its MNs by using the packets stored in its buffer.

At first, the FA_2 compares the packet sequence number of its least recently received packet, P_{2L}, with P_{2H} sent from its MN. If the FA_2 has a higher packet sequence number, that is ($P_{2L} > P_{2H} + 1$), it requests the offset packets, [$P_{2H} + 1$, $P_{2L} - 1$] to the FA_1. Our scheme requires the FA_1 to keep some recently received packets, [$P_{2H} + 1$, P_{1L}], for some amount of time. It encapsulates the request packets and sends them to the FA_2 using a unicast tunneling mechanism. The FA_2 de-capsulates the received packets and saves them to its buffer. These packets might be retransmitted when its MNs request them with NAKs.

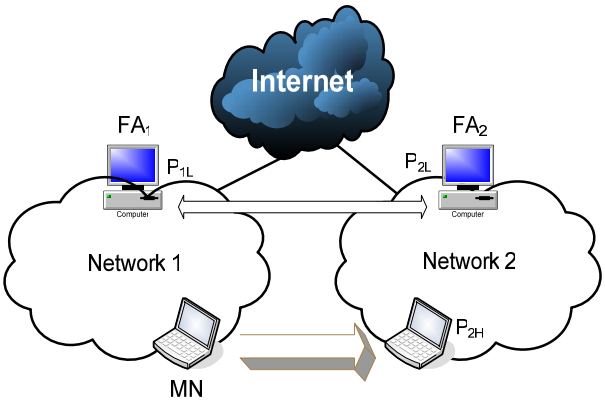


Fig. 2. Mobile IP multicast

If $P_{2L} = P_{2H} + 1$, the FA_2 immediately transmit the packet to the MN starting from packet P_{2L}. If $P_{2L} < P_{2H} + 1$, the FA_2 transmit the packets to the MN starting from packet $P_{2H} + 1$. In both cases, there is no more communications between FA_1 and FA_2 for compensating the offset packets. We also need to mention that this approach will not introduce any duplicate packet caused by out-of-sync problem.

3.2 Buffer Refreshment Policy with Dual Mode

The buffer of the FA should be managed in an efficient manner. The important issue is when to discard packets from the buffers of FAs. Discarding packets that might still be needed is unacceptable because it will force the MNs to contact the sender node whenever they need a retransmission of a discarded packet. Schemes addressing this issue can be broadly divided into ACK-based and NAK-based schemes. ACK-based schemes require each FA to receive one ACK from each of its MN for each packet that node has received. NAK-based schemes impose a much lower feedback load on the FAs but do not let FAs know when they can safely discard packets from their buffers. Also, all previous schemes assumed that packet losses were independent events. But this is not the case for most real networks including Mobile IP networks. We present a new buffer management scheme that takes advantage of the bursty nature of packet losses. Under our scheme, The MNs and FAs operate on dual mode, namely *normal mode* and *error mode*.

3.2.1 MN in A Normal Mode

Our scheme assumes a receiver-initiated error recovery process and requires MNs to send a NAK to their FA every time they detect a packet loss. As a result, a MN that does not experience any packet loss will not send back any feedback to its FA. This property allows us to significantly reduce the ACK implosion at the FA. We refer to this mode of operation as the *normal mode*.

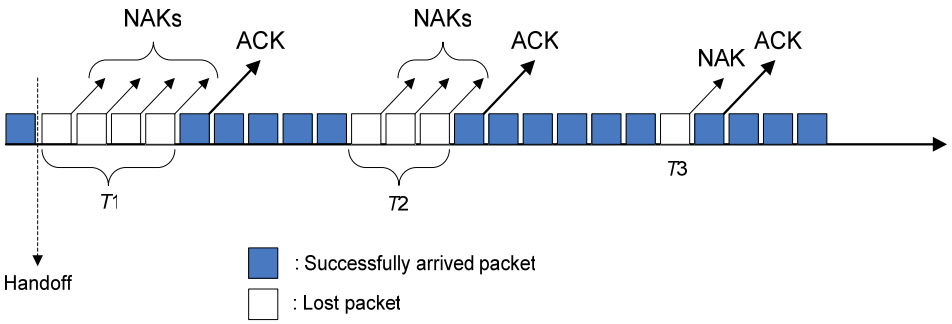


Fig. 3. Example of the feedback schedule of a MN for $k = 1$

3.2.2 MN in An Error Mode

Whenever a MN detects a transmission error, it sends a NAK to its FA and switches to a new mode of operation called the *error mode*. While a MN is in error mode, it sends an ACK for each received packet including retransmitted packets. It will stay in that mode until it has correctly received and acknowledged k consecutive packets. After that it will return to the normal mode and cease to acknowledge the packets it receives. Fig. 3 illustrates this behavior with three different types of packet losses. In this example, k is equal to 1, which means that the MN will return to the normal mode once it has received and acknowledged exactly one correct packet.

Packet loss by a MN can be occasioned by several occurrences. First, packet loss can be occurred due to a MN moving to a new foreign network. We call this *T*1 type packet loss. Second, T2 type packet loss that represents the case where a MN experiences continuous packets loss when one of the underlying routers between its FA and sender node suffers buffer overflow. In this case, the FA should request retransmission of the lost packets from the sender node. Finally, T3 type packet loss occurs when there is unreliable wireless link between the MN and its FA.

3.2.3 FA in A Normal Mode

FAs also have two distinct modes of operation. Under their normal mode, they keep in their buffer recently received packets for a time sufficient to handle a majority of retransmission requests. That is, packet discarding is done by a timeout mechanism with enough retention time.

3.2.4 FA in An Error Mode

Whenever the FAs receive a NAK, they switch to an error mode preventing them from discarding packets that have not been acknowledged by all nodes that have reported a packet loss. They will stay in that mode until they have received k consecutive ACKs from each of these nodes. After that they return to their normal mode.

3.2.5 The Mechanism of FA

In order to integrate the status information of its MNs, each FA will maintain:

1. One *error list* containing all the MNs that are currently operating in error mode: the FA will operate in error mode whenever this list is not empty and in normal mode otherwise.
2. One *ACK list* per acknowledged packet containing all the MNs that have acknowledged the packet: these lists only exist when the FA operates in error mode.
3. One counter per MN to keep track of the number of consecutive assignments it should receive from that node before removing it from its error list.

Observe that our scheme assumes that a FA operating in normal mode will immediately discard any packet that has exceeded its retention time. In practice, we expect these packets to be expelled whenever the FA schedules a buffer sweep. MNs that leave the multicast session without giving any notice can disrupt the multicast session for all MNs in a network. We need to detect a receiver node who does not send k consecutive ACKs after sending a series of NAKs. In order to deal with this situation, the FA will use a timeout mechanism to detect and cut them off.

Our scheme does not guarantee that every FA will always have in its buffer all the packets requested by any of its MNs: it only reduces the likelihood of that event. Retransmission failures can still happen if a NAK arrives after that packet has been discarded but these failures can only happen when the FA is in the normal operating mode. These failures will either occur at the beginning of an error burst or after the nodes have incorrectly assumed that the current error burst has ended.

We can, however, eliminate most of the other retransmission failures by increasing the number k of consecutive ACKs the FA must receive from a node before removing that node from its error list. Finally, we need to mention that the remaining retransmission failures will have to be forwarded to the sender node itself.

4. Performance

In this section, we evaluate the performance of our scheme assuming that all packets in an error burst will always be lost. We assume each MN has two states, namely, state <1> meaning it has correctly received the last packet and state <0> meaning it has not correctly received that packet. Every time a packet is sent to the MN, it will experience a transition that could either leave it in its current state or move it to another state. We will focus our discussion on the two transitions leading to state 0, that is <00> and <10>, as they both correspond to a packet loss. We assume that the probabilities of these transitions, p_{00} and p_{10}, follow Easton's model [4] which are given by

$$p_{00} = r + (1 - r)L$$
$$p_{10} = (1 - r)L$$

where r and L are positive integers ≤ 1.

The steady-state probability p_0 of losing a given packet is given by

$$p_0 = p_0 p_{00} + p_1 p_{10} = p_0 r + p_0(1 - r)L + (1 - p_0)(1 - r)\, L\ ,$$

which simplifies into

$$p_0 = p_0 r + (1 - r)L$$

and

$$p_0 = L.$$

Hence, the L parameter represents the steady state probability of not correctly receiving a packet. The r parameter affects the duration of error bursts. With $r = 0$, all packet losses are independent events. When r increases, packet losses become more and more correlated. Below we show how that parameter can be estimated from the average duration of error bursts. The probability that an error burst will affect exactly b packets is then given by

$$P(b \text{ lost packets per error burst}) = 1p_{01} + 2p_{00}p_{01} + 3p_{00}^2 p_{01} + \ldots$$
$$= \sum_{b=0}^{\infty} (b+1)\, p_{00}^{\ b}\, p_{01}\ ,$$

which is the mean of a geometric distribution. Hence the mean number of lost packets per error burst is given by

$$\mu = \frac{1}{1 - p_{00}} = \frac{1}{p_{01}} = \frac{1}{1 - r - (1 - r)L}$$

Most networks are fairly reliable and have $L \ll r$. In that case, $p_{00} \approx r$. The equation above can be rewritten as

$$\mu \approx \frac{1}{1 - r} \tag{1}$$

Hence, $r = 0.8$ roughly corresponds to an average number of 5 lost packets per error burst.

4.1 Feedback Implosion

The first main advantage of our scheme is that the FA handles much less feedbacks from its MNs, required to discard the packets from FA's buffer.

Consider now a multicast session involving n MNs MN_1, MN_2, …, MN_n sharing the same FA. We assume that these n MNs are subject to independent packet losses with L_i and r_i denoting the respective L and r coefficients of node MN_i.

Since all packets in an error burst are always lost, we do not have to consider the possibility that a MN may incorrectly assume that the current error burst has ended and can safely select $k = 1$. Each MN will thus send to its FA:

1. A NAK every time they do not receive a packet; and
2. An ACK for the first packet they receive correctly after having sent one or more NAKs.

Over a session involving the transmission of m packets, the number of feedbacks from a MN is given by

$$m(p_0 p_{00} + p_0 p_{01} + p_1 p_{10})$$

The number of feedbacks sent by node MN_i to FA can then be rewritten as

$$m(L_i + (1 - L_i)(1 - r_i)L_i) \tag{2}$$

Hence the total number F_{BURST} of feedbacks received by the FA from its n MNs will be given by

$$F_{BURST} = m\sum_{i=1}^{n} (L_i + (1 - L_i)(1 - r_i)L_i) \tag{3}$$

When all data link failure probabilities are equal, that is, $L_1 = L_2 = \ldots = L_n = L$, equation (3) simplifies into

$$F_{BURST} = mn(L + (1 - L)(1 - r)L) \tag{4}$$

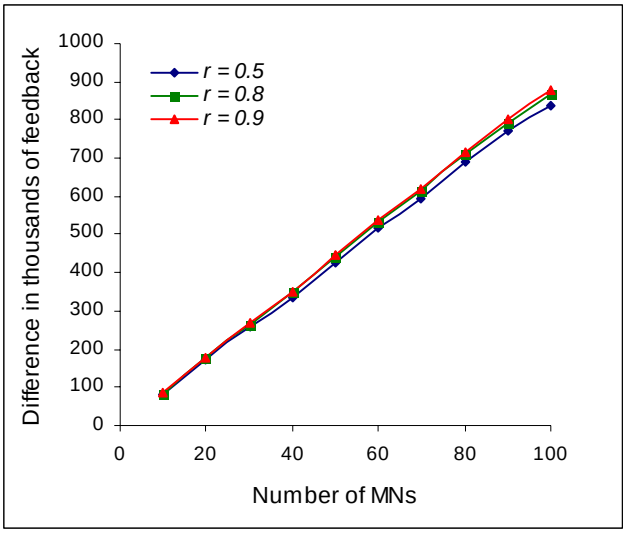


Fig. 4. Difference Δ vs. the number n of MNs per FA_T

Under the same assumptions, the number of feedbacks F_{ACK} for an ACK-based scheme, where all MNs acknowledge all the packets they receive, will be given by

$$F_{ACK} = mn.$$

The difference Δ between the number of feedbacks of the two schemes will be given by

$$\Delta = mn(1 - L - (1 - L)(1 - r)L) \qquad (5)$$

Fig. 4 shows how this difference increases with n for three different values of r when the loss probabilities L_i are uniformly distributed between 0 and 1. We selected the number of transmitted packets m = 10,000, which roughly represents a transfer of 10 megabytes with a packet size equal to 1 kilobyte. When there are 100 MNs in a foreign network, the difference is more than 800,000 feedbacks in all r values. This numerical result indicates that our scheme provides efficient buffer management functionality for the FA by reducing the number of feedbacks sent by its local MNs. This feature provides scalability, since each FA will be able to handle more MNs.

4.2 Packet Availability

Whenever the FA receives a NAK from MN MN_i, it will switch to error mode and cease discarding packets until they have been acknowledged by all MNs that are operating in error mode.

An upper bound for the probability P(FA) that FA will not have in its buffer a packet that is requested by a MN_i is then given by the probability that either MN_i enters an error burst or the FA did not correctly receive the requested packet.

$$\begin{aligned} P(\text{FA}) &= p_1\, p_{10} + p_0\, p_{00} L_{FA} \\ &= (1 - L_i)(1 - r_i)L_i + L_i(r_i + (1 - r_i)\, L_i)\, L_{FA}\,. \end{aligned}$$

where L_{FA} is the packet loss probability of the FA.

However, this upper-bound is extremely pessimistic because it assumes that the FA will never be able to find in its buffer the first packet of any error burst. This is not true because the FA will always keep in its buffer all the packets it receives for a reasonable time interval. Hence, the requested packets are available at the FA if the NAKs arrive before the timer expires. Let us call this probability A. The probability might be very close to 1 if the FA has a large enough timer value. In addition, the packet could still be in the FA's buffer because the FA was waiting for the ACK of another MN that was already inside an error burst. Hence, a more realistic estimate of the probability P(FA) for n MNs is given by

$$\begin{aligned} P(\text{FA}) = {} & p_1\, p_{10} \times [\, P(\text{NAK was lost}) \\ & + P(\text{NAK was not lost but FA did not correctly receive the packet}) \\ & + P(\text{NAK was not lost and FA correctly receive the packet but NAK did not arrive on time and no other MN was in error mode})] \\ & + p_0\, p_{00} L_{FA} \end{aligned}$$

$$\begin{aligned} = {} & p_1\, p_{10}[L_i + (1 - L_i)\, L_{FA} \\ & + (1 - L_i)(1 - L_{FA})(1 - A) \prod_{\substack{j=1 \\ j \neq i}}^{n} (1 - L_j)\,] \\ & + p_0\, p_{00} L_{FA}\,, \quad \text{for} \le i, j \le n \end{aligned}$$

$$\begin{aligned} = {} & (1 - L_i)(1 - r_i)L_i\, [L_i + (1 - L_i)\, L_{FA} \\ & + (1 - L_i)(1 - L_{FA})(1 - A) \prod_{\substack{j=1 \\ j \neq i}}^{n} (1 - L_j)\,] \\ & + L_i(r_i + (1 - r_i)\, L_i) L_{FA} \end{aligned} \qquad (6)$$

In NAK-based schemes using a timer mechanism, FAs discard packets from their buffers after a time interval. Under the same assumptions, the packet missing probability $P(FA_{NAK})$ for NAK-based scheme can be given by

$$P(FA_{NAK}) = L_i \times [\, P(\text{NAK was lost}) + P(\text{NAK was not lost but FA did not correctly receive the packet}) + P(\text{NAK was not lost and FA correctly receive the packet but NAK did not arrive on time})]$$

$$= L_i\,[L_i + (1-L_i)\,L_{FA} + (1-L_i)(1-L_{FA})(1-A)] \quad (7)$$

In order to evaluate the probability that the requested packets will not be present in the FA, we generate the round-trip times between the MNs and FA as Poisson random variables, each having a mean of 40ms. We also uniformly distribute the packet loss probability of each MN between 0.01 and 0.15. In particular, we compare the performance of our scheme with that of a NAK-based scheme keeping all packets in the FA's buffer for 120ms.
Fig. 5 shows how the probability of not finding a requested packet in the FA buffer is affected by the number of MNs per FA. We can see that our scheme always achieves very low packet missing probabilities for any number of MNs per FA. The probability is below 10^{-3} when there are 100 MNs. This result means that the FA will send only single NAK to the sender node when the sender node transmits 10 megabytes of data. In addition, our simulations also indicate that the lowest packet missing probabilities are achieved whenever there are at least 40 MNs per FA.

4.3 Duplicate Packets

In mobile environments, each MN receives multicast packets from more than one FA and these FAs have different one-way transit times from the sender node. Let us consider the network model in Fig. 2. In [8], FA_2 immediately transmits a multicast packet as soon as it receives it from the sender node. If FA_2 has a longer one-way transit time than FA_1, the MNs receive duplicate packets as $P_{2L} < P_{2H} + 1$.
The number of duplicated packets at the MNs can be evaluated as follows. We assume that the MN's current FA has some neighboring FAs. Let us define X as the number of these neighboring FAs. That is, the MN has moved to the current FA from one of these X FAs. Among these neighboring FAs, we assume R FAs have faster one-way transit time than that of the current FA. When we consider n MNs in a network and set the average offset $[P_{2L}, P_{2H} + 1]$ to σ, the average number of duplicate packets D is given by

$$D = \frac{n\sigma}{R} \quad (8)$$

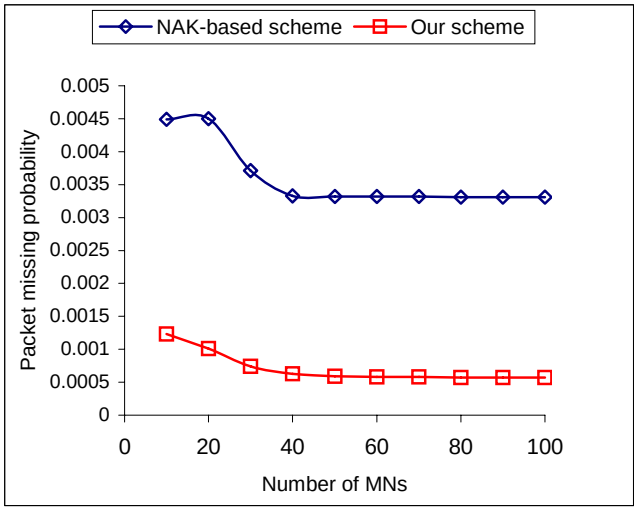


Fig. 5. Packet missing probability $P(\text{FA})$ and $P(FA_{NAK})$

This results in linear increase of the number of duplicate packets as the number of MNs increase, indicating bandwidth consumption in the foreign network. On the other hand, as we saw in section 3, our scheme does not impose any duplicate packets during the handoff.

5. Conclusions

We have presented a new buffer management scheme that can be applied in Mobile IP networks. Our scheme takes into account the temporal locality of packet losses using two possible operation modes. It limits both the number of feedbacks sent by MNs to their FAs and the probability that a given FA will not be able to handle a given packet retransmission request. In addition, our scheme allows MNs to avoid duplicate packets during handoff.

Acknowledgments

This work benefited from a lot of discussions with Dr. Jehan-François Pâris. We would also like to thank Mr. Wood Kanampiu for his discussions. Their advice was a great help to our work.

References

[1] A. Acharya, B.R. Badrinath, "A Framework for Delivering Multicast Messages in Networks with Mobile Hosts," *Wireless Networks, Special Issue on Routing in Mobile Communication Networks,* 1(2): 199-219, October 1996.

[2] K. Brown, S. Singh, "RelM: Reliable Multicast for Mobile Networks," *Journal of Computer Communications*, 21(16): 1379-1400, 1998.
[3] P. Chumchu, A. Seneviratne, "Multi-Level Reliable Mobile Multicast Supporting SRM," *Proc. of the 2002 IEEE* VTC 2002, pp. 1410~1414, Spring 2002.
[4] M. C. Easton, "Model for database reference Strings Based on Behavior of Reference Clusters," *IBM Journal of Research and Development*, 22(2):197–202, March 1978.
[5] E. Gustafsson, A. Jonsson, and C. Perkins, "Mobile IP Regional Registration," Internet draft, draft-ietf-mobileip-reg-tunnel-05, txt, September 2001.
[6] S. K. Kasera, J. Kurose, and D. Towsley, "Buffer Requirements and Replacement Polices for Multicast Repair Service," *Proc. of the 2nd Network Group Communication Workshop*, pp. 5-14, Nov. 2000.
[7] I. Kilolaidis, J. J. Harms, "A Logical Ring Reliable Multicast Protocol for Mobile Nodes," *Proc. of the 7th IEEE ICNP*, pp.106-113, November 1999.
[8] W. Liao, J. A. Ke, and J. R. Lai, "Reliable Multicast with Host Mobility," *Proceedings of the IEEE GLOBECOM 2000*, pp. 1692-1696, November 2000.
[9] Whetten and G. Taskale, "The Overview of Reliable Multicast Transport Protocol II," *IEEE Networks*, 14(1):37–47, Jan.-Feb. 2000.
[10] W. Yoon, D. Lee, C. Yu, and M. Kim, "Tree-Based Multicast in Combined Fixed/Mobile IP Networks," *Proc. of the 25th LCN*, November 2000.

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.

Eunjung Lee received her B.S. degree in Computer Science and Engineering from Hankuk University of Foreign Studies (HUFS) in Yougin, Korea in 1998 and M.S. degree in Computer Science from the University of Houston in 2005. She served as a software developer at Kinesix Software in Houston, TX for two and a half years. She is currently working at the PSF technology as a research scientist. Her research interests include mobile computing, image processing, and network security protocols.