INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2012; **25**:553–570
Published online 20 April 2009 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.1009

A reliable overlay video transport protocol for multicast agents in wireless mesh networks

Jinsuk Baek[1], Paul S. Fisher[1], Minho Jo[2] and Hsiao-Hwa Chen[3,*,†]

[1]*Department of Computer Science, Winston-Salem State University, Winston-Salem, NC, U.S.A*
[2]*Graduate School of Information Management and Security, Korea University, Seoul, South Korea*
[3]*Department of Engineering Science, National Cheng Kung University, Tainan City, Taiwan*

SUMMARY

A new video transport protocol for multicast agents in wireless mesh networks (WMNs) is proposed in this paper. The proposed protocol enables a significant reduction in the transmission overhead, while providing reliable communication for its use in multicast applications. This proposed reliable protocol provides a practical approach for an overlay peer-to-peer multicast facility supported within the application layer. This obviates the need to give upgraded routers capable of handling multicast broadcasting or modify the existing protocol stack. The protocol tolerates partial losses in multimedia transmissions, while supporting control of the delay sensitivity of such transmissions in WMNs. The key issue in this protocol is the ability to detect packet loss, anticipate retransmission requests, and use the anticipated retransmission requests to transmit the lost packets prior to requests from other receiving agents. The proposed protocol allows for the receiver to determine if retransmission of lost packets is required, ensuring the greatest flexibility needed for a reliable multicast protocol. Copyright © 2009 John Wiley & Sons, Ltd.

Received 25 November 2008; Revised 29 January 2009; Accepted 4 February 2009

KEY WORDS: wireless mesh network; transport protocol; peer-to-peer multicast; video signal transmission

1. INTRODUCTION

A growing number of distributed applications require a single sender to transmit the same data to multiple receivers in wireless mesh networks (WMNs). These applications include bulk data transfer, distance learning with continuous streaming media, video-conferencing with shared data applications, data feeds, IPTV, Web cache update, and distributed interactive gaming over the WMN backbone. These applications are classified as point-to-multipoint communication rather than traditional point-to-point communication. Multicasting, the capability to deliver a single message to multiple recipients using the same Internet Protocol (IP) address, is the only efficient and scalable solution supporting these kinds of applications [1].

In order to provide this capability, multicasting must be realized in different layers of the Internet protocol stack including the link layer, IP network layer, and application layer in WMNs. IP multicasting does not guarantee reliable datagram delivery, due to best-effort for transmission only. It also requires all deployed routers to be upgraded with a multicast capability. Therefore, a more practical approach is an overlay peer-to-peer multicast facility [2–8] that is supported from the

*Correspondence to: Hsiao-Hwa Chen/Minho Jo, Department of Engineering Science, National Cheng Kung University, No. 1, University Road, Tainan City 701, Taiwan and Graduate School of Information Management and Security, Korea University, #5 Anam-Dong 1 Ga, Seungbuk-Gu, Seoul 136-701, South Korea.

†E-mail: hshwchen@ieee.org, minhojo@korea.ac.kr

Contract/grant sponsor: Institute for Information Technology Advancement (IITA)
Contract/grant sponsor: National Science Council research; contract/grant number: NSC97-2219-E-006-004

application layer. Owing to its simple deployment and central management for all multicast states at the end system, overlay multicasting has been considered as a promising approach for providing a reliable multicast capability. The success of overlay multicasting depends on the transport layer mechanism, because it uses a reliable unicast mechanism to provide a reliable multicast service, and an unreliable one when the application does not require a reliable service. As a variety of networks (e.g. LAN, Wireless Local Area Network (WLAN), and 3rd Generation Partnership Project (3GPP)) are currently deployed, and the next generation network is converging to an all-IP-based unified network, many mobile terminals are configured to run in multi-homed environments by simply installing two or more network interfaces. In addition, WMNs [9], a key technology providing a wireless backbone to multi-homed mobile terminals, have shown a rapid growth and inspired numerous applications. In this environment, stream control transmission protocol (SCTP) [10] has been considered as a proper transport layer protocol supporting overlay multicast, owing to its multi-streaming and multi-homing features. On the other hand, the data transfer services offered by traditional transport protocols such as TCP and User Datagram Protocol (UDP) are inadequate for these environments.

Unfortunately, with the conventional overlay, multicast architecture does not scale well, since the application of the multicast sender (MS) has to fully handle the separate sessions for each connection. In order to alleviate this problem, multiple multicast agents (MAs) have been deployed for scalable services [11]. The MAs can be used to relay the multicast data from an MS to multicast receivers (MRs). More importantly, each MA buffer temporarily maintains all packets it has recently received from the MS, and performs local error recovery for all MRs in its group. Such a design, defined as the logical tree topology of the application layer, achieves scalability by distributing the MS retransmission workload among multiple MAs.

We should note that the legacy SCTP is basically a connection-oriented transport protocol providing reliable end-to-end message delivery via selective Acknowledged (ACK) (SACK), flow control, and congestion control. This creates a tradeoff in video multicast service for MRs. First, the legacy SCTP incurs an unavoidable overhead in transmitting loss-tolerant, delay-sensitive, real-time, video data packets. At the same time, video data are loss-tolerant only when the data packet losses are independent as a result of link errors at the physical layer or frame collisions at the Media Access Control (MAC) layer. Finally, for SCTP, each MA is required to send a SACK to the MS for each packet it has correctly received. As a result, the MS's ability to handle these SACKs limits the number of MAs participating in an reliable overlay multicast session, thus compromising scalability. To the best of our knowledge, all previous protocols [12–22] designed to provide reliable transmission focused on defining the interactions between the MA and its MRs. However, interactions between the MS and MAs are often overlooked by these protocols, even though such actions can significantly affect transmission performance. Therefore, we need a new protocol applicable to MS-to-MA communications.

We propose a new protocol, focused on the interactions between the MS and its MAs in WMNs. The proposed protocol considers partial loss-tolerant and delay-sensitive characteristics for video data. In our protocol, each MA adaptively decides if retransmission is needed whenever it detects packet loss. This decision is based on the packet loss patterns. In other words, if the packet losses are determined to be continuous, the MA sends an *urgent* Not Acknowledged (NAK) message rather than a regular NAK. This *urgent* NAK message not only affects the packet that transmitted the *urgent* NAK message, but also affects the packets within the defined range. Our prediction scheme calculates how many more consecutive packets will be lost, and a single *urgent* NAK message requests retransmission for these packets. After receiving this message, the MS immediately retransmits the packets to the MA, without waiting for other NAK messages for the same packet from other MAs. At fixed infrequent intervals, they also send cumulative ACKs, to indicate which packets can be safely discarded from the MS buffer. Our cumulative ACK also differs from the conventional cumulative ACK, because it tolerates intermediate packet losses, since we consider partial loss-tolerant characteristics of video data.

Our protocol has many advantages over the existing protocols. First, we avoid any risk of ACK implosion at the MS, because each MA sends only infrequent ACKs and their sending timings are randomized among the MAs. Second, the proposed protocol can avoid NAK implosion at an MA

DOI: 10.1002/dac

because the MA autonomously detects the potential spatial locality of the packet losses among the MRs in its group, and sends an *urgent* NAK for these packets at once. Third, it guarantees fast recovery of transmission errors for local group members, since the packets requested from MAs are immediately retransmitted by the MS. Finally, because our well-defined dynamic threshold can adapt to the average error burst, duplicate packets at the MA are rare. In addition, its dynamism does not impose an overhead for evaluating an average error burst.

The remainder of this paper is outlined as follows. Section 2 reviews the existing transport layer protocols for providing overlay, reliable, and multicast services. Section 3 introduces our new protocol. In Section 4, we show the performance of the proposed protocol, followed by the conclusion given in Section 5.

2. RELATED WORKS

2.1. Overlay multicast architecture

First we consider the overlay multicast tree configuration in WMN proposed in [23], which consists of an MS, a session controller (SC), MAs, mobility controllers (MCs), and MRs. The configuration is shown in Figure 1. The SC is a functional entity having a dedicated communication channel with an MS for session control, while the MA can be used to relay multicast data from the MS to MRs, and can be deployed in both relayed multicast backbone networks and wireless mesh backbone networks. Both SCs and MAs can be deployed in either relayed multicast backbone networks or wireless mesh backbone networks. The MC is used to manage the session in the access network to support mobility. For data delivery, the MS transmits multicast data to MRs in cooperation with MAs in the network. The MS assigns a proper MA for an MR and maintains this information in its data forwarding table. Accordingly, the MR receives the data packets from the MA, and all

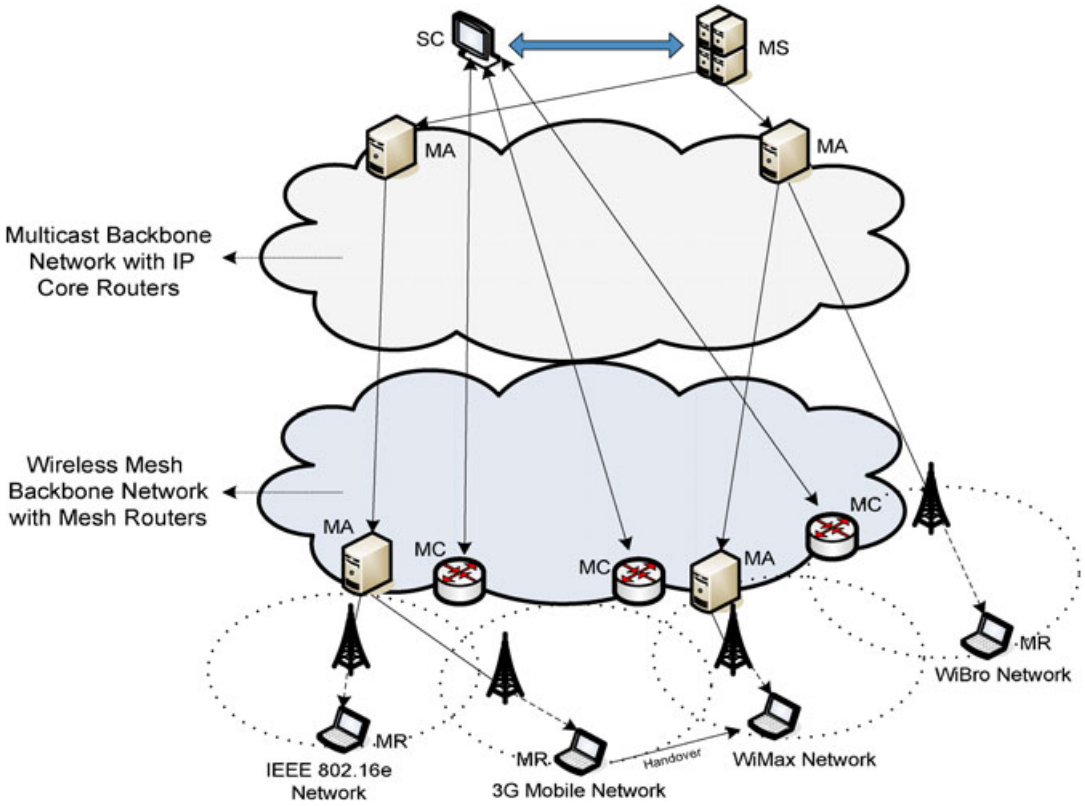

Figure 1. Configuration in overlay video multicast service in WMNs.

DOI: 10.1002/dac

retransmissions are performed via MA-to-MR communications. For control, the MR informs SC of a session to join and the SC informs the MR about the corresponding MC for the MR. For monitoring the session, each MR reports its current status (e.g. link layer handover with an access point or IP layer handover with MC) to MC or SC, which will eventually be aggregated with the SC.

Both data delivery and the control service can be scaled up by constructing corresponding logical trees. The logical tree construction includes several steps: (1) advertising the multicast session, (2) discovering MA, SC, and MC nodes for each MR node, and (3) binding each MR node to its MA node. In the multicast session advertisement phase, all nodes obtain the group address of the multicast, the addresses of the MS and SC node, and other necessary information for tree construction. This process can be realized by using a mechanism such as a web page announcement. This tree-based service provides both high scalability and reliability as well.

2.2. Reliable multicast

Many transport layer protocols [1, 12–22] have been proposed to provide reliable multicasting. The same goal is achieved with a variety of retransmission control schemes, feedback scheduling schemes, and strategies for deciding which network entity should buffer packets for retransmission and the period these packets should remain in the buffer.

Scalable reliable multicast [14] is a well-known receiver-initiated multicast protocol that guarantees out-of-order reliable delivery using NAKs from receivers. When a receiver experiences a packet loss, it multicasts an NAK to all the participants of the multicast group allowing 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 receivers in the multicast session instead of leaving the entire workload to the sender. The drawback is that it requires all receivers 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 receivers that received the packet successfully first time. This phenomenon has the consequences of increased bandwidth consumption and Internet traffic.

Reliable multicast transport protocol (RMTP) [19], 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 DR. This ACK contains the maximum packet number that each receiver has successfully received. The drawback for this periodic feedback policy, however, is a significant increase in error recovery delay since the receivers do not immediately request 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 Guo and Rhee [15] organizes receivers into groups where they collectively take part in error recovery. This is achieved by letting receivers periodically exchange history information about the set of packets they have received. Eventually, one receiver in the group becomes aware that all the receivers in the group have successfully received a given packet and broadcasts this to all the members in the group. Then all members can safely discard that packet from their buffers. Noticeably this feature causes high message traffic overhead because the algorithm requires a frequent exchange of messages.

An efficient buffering policy has been proposed in [20]. In order to reduce the amount of the total space for the required buffer, only a small set of receivers buffer the packet. Receivers that have not correctly received a given packet use a hash function to select the members that have the packet in their buffers and request a retransmission of the packet from one of them. Unfortunately, their selection method does not consider geographic locations between different receivers. Hence, its scalability is constrained because the latency for error recovery increases with the number of participants.

DOI: 10.1002/dac

The bimodal multicast protocol (BMP) [12] employs a buffer management policy where each group member receiver buffers arrived packets for a certain amount of time. To enhance effectiveness, it 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 time period for eventual retransmission requests in its group. After the expiration of time, only a small randomly selected number of receivers will continue to buffer this packet. The inefficiency of BMP is that the 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.

The Search Party protocol [13] uses a timer to discard the packet from the buffer: each member in the group simply discards packets after a fixed amount of time. The protocol remains vague on the problem of selecting the proper time interval for discarding packets. More importantly, all the mentioned protocols have the aim of providing a bulk data transfer service rather than a video multicasting service.

However, all the protocols have the aim of providing a bulk data transfer service rather than a video multicasting service. Moreover, they have the aim of providing a reliable multicast service over an unreliable IP multicast, rather than an overlay multicast system in WMNs.

As the performance of unicast-based, overlay, reliable, multicast capability is highly dependent on the transport layer technology as the basis for data transfer, and since many MR configurations provide a multi-homed environment, SCTP has been considered as a promising approach supporting overlay, reliable multicast [24, 25]. The multi-homing feature of SCTP can maximize the utilization of the multi-homed environment when it is used as a transport layer protocol for reliable multicast service. Moreover, it increases network availability, since it enables an MR to switch its primary path to an alternative path when the MR detects primary path failure. A link failure can easily occur, especially when an MR performs a vertical handover between heterogeneous access networks.

3. PROPOSED PROTOCOL

We propose a new protocol supporting MS-to-MA communications for video multicasting service over the overlay multicast architecture in WMNs. The SCTP provides connection-oriented reliable transmission over the IP core network via SACK, flow control, congestion control and avoidance, as well as failure detection and recovery. It also provides faster transmission than TCP, since the multi-stream mechanism is designed to overcome the head-of-the-line blocking problem of TCP. It divides the overall SCTP message flow into sub-flows, and the partial ordering of the message is performed within each of the sub-flows. This message-oriented feature prevents the message of one sub-flow from interfering with another sub-flow, resulting in reduced transmission delay.

Here, we consider how SCTP works when it is applied to a video multicasting service handling time-sensitive data. The retransmitted packets based on SACK at the MA are worthless to the MA after their lifetimes have expired, and can only increase network congestion. Nonetheless, retransmission is required when the MA incurs continuous packet losses over a significant time period. Such a time measure is a function of the key frame rate in the video stream. Therefore, we need a different transport layer protocol to provide partially reliable multicast service for video data. In order to satisfy its loss-tolerant but delay-sensitive characteristics, the transport layer protocol must be designed to tolerate only intermittent packet losses. Instead, the packets, which are continuously lost, must be retransmitted to the MA as quickly as possible, because they directly affect the quality of service (QoS) of its MRs. We begin by considering the following three facts:

(1) Current wireless links are generally reliable, but link errors remain to be a major problem. Therefore, packet losses can be independent and are not correlated with previous transmission failures. The packet losses can be independent among all MAs of the same MS. However, these packets do not need to be retransmitted, especially when the packets are for video data, which have loss-tolerant characteristics. Even with packet retransmission, such retransmitted packets may be worthless, since the video stream will not back up to accommodate the retransmitted packets, unless the video stream is being viewed with a small delay time by the MA.

DOI: 10.1002/dac

(2) Packets losses are also likely to be the result of the router's buffer overflow. In this case, packet losses tend to occur in bursts. Again, based upon the time criticality of video packets, retransmission must be available within a narrow time window. As a result, retransmission without considering timing simply results in bandwidth hogging, as the non-timely retransmitted packets are simply discarded by the MA.
(3) The MS buffer maintains all the packets it has recently transmitted to perform error recovery for all dependant MAs. As the buffer size for a specific multicast session is limited, the MS must periodically discard packets from its buffer. Discarding packets too late results in inefficient use of the limited available buffer space on the MS. On the other hand, discarding packets that might still be needed is unacceptable for a reliable multicast service.

3.1. Urgent NAK with prediction

In order to simplify our discussion, we assume that the multicast flow consists of multiple error bursts. In our protocol, C denotes the current packet sequence number (PSN) where loss is incurred. $E_i(b)$ is the expected length of the error burst at error burst i, which can be calculated using the historical lengths of the actual error bursts. $A_i(b)$ is the actual number of consecutive packet losses at error burst i. The most recent $E_n(b)$ at the current error burst n can then be expressed as a weighted average by

$$E_n(b) = \sum_{i=2}^{n} \omega(i-1) A_{i-1}(b) \tag{1}$$

$$\omega(i-1) = \alpha\omega(i), \quad 2 \leqslant i \leqslant n-1, 0<\alpha<1 \tag{2}$$

$$\omega(n-1) = \alpha, \quad 2 \leqslant i \leqslant n-1, 0<\alpha<1 \tag{3}$$

where $\omega(i)$ is the weight value for error burst i. The proposed protocol only considers a limited number of recent historical values rather than the entire history, by maximizing the minimum size of i, and assigns more weight to the most recent $A(b)$ via the weight factor α.

In (4) the variable max_T represents the maximum number of consecutive packet losses before an *urgent* NAK message is sent to the MS. $E(b)$ and max_T are key parameters showing the efficiency of our protocol. $E(b)$ is contingent upon the actual lengths of previous error bursts, which are determined via a prior observation period. Subsequently, $E(b)$ remains fixed until the number of packet duplications d is greater than the threshold T_d. Then, $E(b)$ is reevaluated. The format of an *urgent* NAK message of the MA i is given as follows:

$$\begin{aligned} urgent\ NAK &= \{\max\_\mathrm{PSN}(i), \min\_\mathrm{PSN}(i), \max\_\mathrm{PSN}wP(i)\} \\ &= \{C - \max\_\mathrm{T}, C, C + E(b) - (\max\_\mathrm{T} + 1)\}, \quad 1 \leqslant i \leqslant N \end{aligned} \tag{4}$$

where max_PSN(i) is the maximum PSN that the MS can discard, min_PSN(i) is the minimum PSN that the MS needs to retransmit to MA i, and max_PSN$wP(i)$ is the predicted maximum PSN that the MS also needs to retransmit to MA i.

A packet loss scenario is depicted in Figure 2. In order to simplify our explanation, let us assume that $E(b)$ is set to five, by assuming that d is equal to or less than T_d, and max_T is set to one. Also, there is an error burst in the middle of every 12 packet transmissions in our example. For the first error burst scenario, packets 1 and 2 exhibit successful transmissions and, however, packet 3 is lost. This does not cause an NAK message, because we cannot conclude that the packet loss is continuous. Next, packet 4 is lost. This causes an *urgent* NAK message to be transmitted. The MA eventually requests packet retransmissions for packets 4–7 at once, while it informs the MS that it can safely discard up to packet 3. The first error burst, a series of consecutive packet losses, continues with packet losses for packets 5, 6, and 7. The error burst ends with the successful receipt of packet 8, followed by the successful receipt of packets 9, 10, 11, and 12. In the second error

DOI: 10.1002/dac

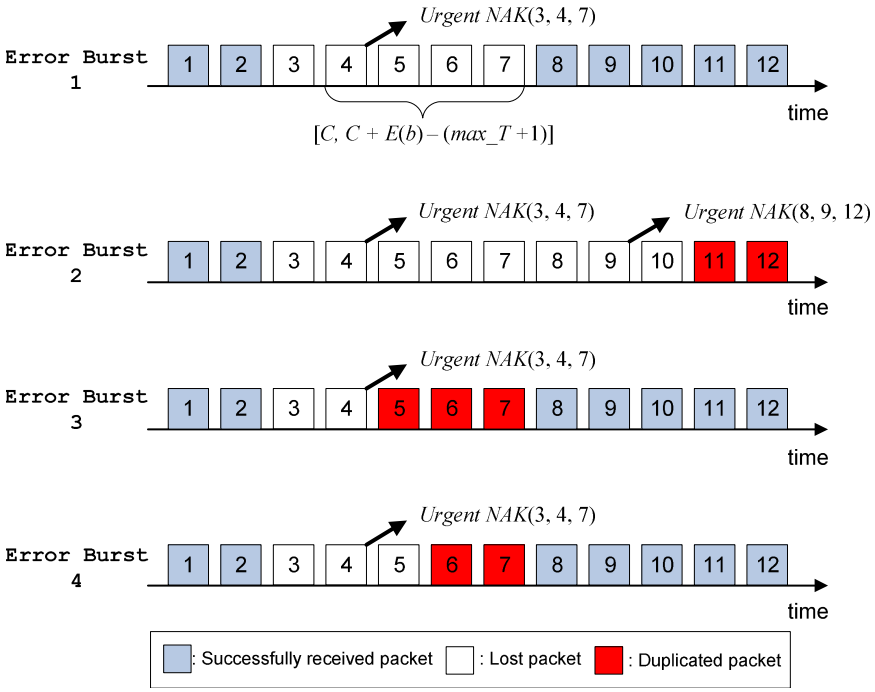

Figure 2. An example of packet loss scenario at a multicast agent.

burst scenario, packet 3 is lost, but no NAK is required either, because our protocol tolerates partial packet losses for video data. It then sends an *urgent* NAK message, where C equals four. The MS retransmits packets 4–7 and packets 9–12 after it receives the second *urgent* NAK message, at packet 9. Packets 1, 2, and 5–12 are successfully delivered in the third error burst scenario, however packets 3 and 4 are lost. By generating an *urgent* NAK message for packet 4, packets 4–7 are retransmitted. While packet 4 is corrected, packets 5, 6, and 7 are received as duplicated packets, all of which are discarded. In the fourth error burst scenario, packet losses are incurred for packets 3, 4, and 5, triggering an *urgent* NAK message when packet 4 is lost. Packets 4–7 are retransmitted, resulting in duplication with packets 6 and 7. Even though we assume that $E(b)$ is set to five in this example, each round actually uses the $A(b)$ values determined in previous error bursts, to create a range in which packet retransmission will occur. Of course, this calculation is performed only when d is greater than T_d.

3.2. *Packet duplication*

This component of our proposed protocol has the following characteristics: for the first error burst, the MA incurs five lost packets, with no duplication after retransmission; for the second error burst, it incurs eight lost packets, with two duplications after retransmission; for the third error burst there are two lost packets and three duplications; for the fourth error burst there are three lost packets and two duplications after retransmission. The number of duplicated packets, d_i, at the error burst i can be formalized for the various cases.

Given the same scenario, when max _T is zero, there is faster retransmission with no duplications; however, there is no means to distinguish between an intermittent and continuous events. Given the same scenario, when max _T is two, there is no duplication for the first and the third error bursts, and two duplications for the second and fourth error bursts. Based on this, we can conclude that the lower the number of packet losses incurred before an *urgent* NAK message is transmitted, the greater the degree of duplication, but with a much higher retransmission rate. Conversely, higher numbers result in little duplication, but a lower retransmission rate. Since the number of duplicated packets is relative to the number of packet losses incurred before transmitting an *urgent* NAK message, when

DOI: 10.1002/dac

one packet loss is incurred before transmitting an *urgent* NAK message, this approach provides an efficient retransmission rate, as well as the fastest means to distinguish between an intermittent and continuous events. Evaluating the error burst and its relationship to the number of lost packets and packet duplication, we find that the number of duplications depends on $E(b)$.

Given the scenario where $E(b)$ is equal to five, in the worst case, three duplicated packets result, along with the two lost packets. In the best case, there are no duplications with five lost packets. Our goal is to ensure that there are neither extreme numbers of duplicated packets nor lost packets, to ensure that the number of duplicated packets and lost packets are balanced.

To counter the high duplication rate associated with our proposed protocol, we institute a threshold T_d. This threshold is used until the number of duplications is greater than the threshold. Then, we reevaluate $E(b)$ with previous $A(b)$'s and associated weights ω. We concluded that a dynamic threshold is useful and efficient for the protocol. In determining the threshold, we found that large thresholds result in more duplicate packets. Such a threshold results in reevaluating the $E(b)$ at a lower rate, but also affects storage use, with an increased number of duplicate packets resulting in less efficient use of storage. However, with a small threshold, the rate at which $E(b)$ is reevaluated is increased. To ensure the efficiency of our protocol, we set the threshold at double of $E(b)$, and this provides us with a reasonable storage and reevaluation frequency.

When determining whether the threshold should be a fixed or dynamic, there are several considerations. If we consider a fixed threshold over a large series of error bursts, such as the one composed of 100 error bursts, the number of duplicated packets should eventually be greater than the threshold, due to the accumulation of duplicated packets within that series. Additionally, even when we reevaluate $E(b)$, we find that there is a greater probability that the new $E(b)$ is the same as the current $E(b)$. This leads to unnecessary calculation of values, resulting in the same performance. A threshold over a small series of error bursts, such as the one comprised of three error bursts, is only useful when the number of duplicated packets is greater than the threshold over a small series of error bursts. Because of these concerns, with fixed threshold values, we selected a dynamic value. Our dynamic threshold conforms to the following relationship: $T_d = 2N(E)$. In other words, we set the threshold T_d to double the current number of error bursts, $N(E)$. When the cumulative number of duplicated packets, d, in the current error burst is equal to or greater than double the current $N(E)$, we reevaluate $E(b)$. By this means, we found that if every error burst has more than one duplicated packet, the current $E(b)$ may not be correct, and does not accurately represent the current transmission status.

Consider the following example. In the first scenario of 12 packet transmissions, there was no duplication. Since the number of duplicated packets is less than the current number of error bursts ($N(E) = 1$), no reevaluation of $E(b)$ is necessary. This remains true for the second and third error bursts, since the second error burst resulted in only two duplications and the third one resulted in only three duplications. The fourth error burst results in two more duplicated packets, and d becomes seven. Because d remains less than double $N(E)$, reevaluation of $E(b)$ remains unnecessary. However, if the fifth error burst results in three more duplicate packets, d becomes 10. Because it is equal to double ($N(E) = 5$), $E(b)$ is then reevaluated. The algorithm used in the proposed protocol is shown in Algorithm 1.

3.3. Packet discarding

In this section, we propose a packet discarding scheme to provide an efficient buffer management mechanism for the MS. When an MA i joins a multicast session, it receives a control packet from the MS specifying the maximum number $N_{\max}$ of sent packets the MS usually maintains in its buffer. The MA then generates a random number between one and $N_{\max}$ identifying the first packet after which it will send an ACK packet to the MS. This packet serves as a cumulative ACK for packets 1 to N^i_{rand}. Subsequently, MA i sends a similar ACK packet after each packet max_PSN(i), such that

$$\max\_\text{PSN}(i) = N^i_{\text{rand}} + kN_{\max}, \quad k = 0, 1, 2, 3, \ldots \tag{5}$$

DOI: 10.1002/dac

Algorithm 1 : Algorithm used in the proposed protocol.

```
1. if [A_i(b) ⩾ E_i(b)] then
2.   if A_i(b) = E_i(b) or A_i(b) = k Ė_i(b) then
3.     for k = 1, 2, 3, ... do
4.       d_i = 0
5.     end for
6.   else
7.     if [A_i(b) mod E_i(b)] < max T+1 then
8.       d_i = 0
9.     else
10.      d_i = E_i(b) − [A_i(b) mod E_i(b)]
11.    end if
12.  else
13.    if [A_i(b) < max T+1] then
14.      d_i = 0
15.    else
16.      d_i = E_i(b) − A_i(b)
17.    end if
18.  end if
19. end if
```

It is noted that our cumulative ACK differs from the conventional cumulative ACK, because it does not imply that the MA has successfully received up to packet max_PSN(i). Instead, it immediately transmits an ACK for packet max_PSN(i) when it receives a packet having a higher PSN than max_PSN(i). Therefore, our cumulative ACK implies that the MS can safely discard up to packet max_PSN(i) from its buffer, because the packets that have smaller PSNs than this packet are not needed for retransmission to MA i. As a result, our protocol tolerates partial intermittent packet losses for video data. In addition, the MAs always acknowledge the last packet of the transmission. Since N different MAs start ACK transmissions at random offsets:

$$N_{\text{rand}}^{i} = \text{random}(1, N_{\max}), \quad 1 \leqslant i \leqslant N \tag{6}$$

These are uniformly distributed over time. Therefore, we eliminate the risk of a sudden burst of ACKs sent for the same packet. In order to ensure that the MS receives at most one ACK from MA i every $N_{\max}$ packets even after the MA sends an *urgent* NAK, each MA sets its max_PSN(i) as follows:

$$\text{max\_PSN}(i) = \text{max\_PSN}wP(i) + N_{\text{rand}}^{i} + kN_{\max}, \quad k = 0, 1, 2, 3, \ldots \tag{7}$$

Let max_PSN(i) be the last packet sent by MA i. Assume that there is no pending retransmission packet. The MS can then safely discard up to packet D whenever it receives an ACK or *urgent* NAK from its MAs.

$$D \leqslant \min\{\text{max\_PSN}(i) | 1 \leqslant i \leqslant N\} \tag{8}$$

We also note that our cumulative ACK ensures that the MS buffer always has all the packets that can be requested by any of its MAs. Assume, for instance, that MA i predicts continuous packet losses from packet min_PSN(i) to packet max_PSN$wP(i)$. It sends an *urgent* NAK for these packets to the MS. The MS buffer maintains all packets with PSN greater than the last acknowledged packet max_PSN(i) and max_PSN(i)<min_PSN(i). Hence, packet min_PSN(i) is always available in the MS buffer, and can be retransmitted. In addition, the arrival time of the next acknowledgment packet max_PSN$(i)'$ is always larger than that of max_PSN$wP(i)$ because

$$N_{\max} \gg \max\{A_i(b) | 1 \leqslant i \leqslant N(E)\} \tag{9}$$

As a result, our ACK scheduling scheme provides reliability for continuous packet losses at the MAs.

DOI: 10.1002/dac

4. PERFORMANCE ANALYSIS

In this section, we evaluate the performance of the proposed protocol by using numerical analysis. We assume that all packets are lost in an error burst. In our protocol, each MA has two states, namely, state $\langle 1 \rangle$ that means it has successfully received the last packet and state $\langle 0 \rangle$ that means it has failed to receive that packet. Every time a packet is sent to the MA, it incurs a transition that will either leave it in its current state or change it to another state. We focus on the two transitions leading to state $\langle 0 \rangle$, that are $\langle 00 \rangle$ and $\langle 10 \rangle$, as they both correspond to a packet loss. We assume that the probabilities of these transitions conform to Easton's model [26], defining r as the packet loss correlation factor and L as the packet loss probability. This model is given by

$$P_{00} = r + (1 - r)L \tag{10}$$

$$P_{10} = (1 - r)L, \quad r \leqslant 1, L \ll r \tag{11}$$

The steady-state probability P_0 of losing a given packet is given by

$$P_0 = P_0 P_{00} + P_1 P_{10} = P_0 r + P_0(1 - r)L + (1 - P_0)(1 - r)L \tag{12}$$

This can be simplified as

$$P_0 = P_0 r + (1 - r)L \;\Rightarrow\; P_0 = L \tag{13}$$

Hence, the L parameter represents the steady-state probability of failing to receive a packet. The r parameter affects the duration of error bursts. When $r = 0$, all packet losses are independent. When r increases, packet losses become increasingly correlated. We show how this parameter can be estimated from the average duration of error bursts. The probability that an error burst will affect exactly k packets is given by

$$\begin{aligned} P(k \text{ lost packets per error burst}) &= 1P_{01} + 2P_{00}P_{01} + 3P_{00}^2 P_{01} + \cdots \\ &= \sum_{k=0}^{\infty} (k + 1) P_{00}^k P_{01} \end{aligned} \tag{14}$$

which is the mean of a geometric distribution. Hence, the mean number of lost packets per error burst, $E(A(b))$, is given by

$$E[A(b)] = \frac{1}{1 - P_{00}} = \frac{1}{P_{01}} \tag{15}$$

Most networks are fairly reliable, and have $P_{00} \ll P_{01}$. Here, $P_{00} \approx r$. The above equation can be expressed as

$$E[A(b)] \approx \frac{1}{1 - r} \tag{16}$$

Hence, $r = 0.9$ roughly corresponds to an average of 10 lost packets per error burst.

4.1. Message implosion

One of the main advantages of the proposed protocol is that the MS handles far fewer feedback messages sent by its MAs. We compare our protocol with SCTP, which requires all MAs to transmit SACKs for each successfully received packet. We also compare it with legacy NAK-based protocol [16], which does not define an *urgent* NAK message.

With our protocol, the loss probability of MA i must be considered separately, as shown in (16). One is the loss probability due to link errors. The other is the loss probability due to a buffer overflow of the underlying router.

$$L(\mathrm{MA}_i) = L(\mathrm{MA}_i)_{\mathrm{indep}} + L(\mathrm{MA}_i)_{\mathrm{cont}} \tag{17}$$

DOI: 10.1002/dac

Therefore, in the case of link errors, which result in independent packet losses, the number of NAK messages is not counted in our protocol. In the case of a buffer overflow, which results in continuous packet losses, the number of *urgent* NAK messages can be obtained by calculating the loss probability for MA i, $L(\mathrm{MA}(i))$, with the error burst average $E(A(b))$, max_T, and m number of transmissions. As our well-defined threshold implies that $E(b) \approx E(A(b))$, this results in the following buffer overflow equation:

$$m\frac{L(\mathrm{MA}_i)_{\mathrm{cont}}}{E[A(b)]}[E[A(b)] - \{\max\_\mathrm{T} + (E[A(b)] - (\max\_\mathrm{T} + 1))\}] = m\frac{L(\mathrm{MA}_i)_{\mathrm{cont}}}{E[A(b)]} \tag{18}$$

In addition, our protocol requires each MA to send $\lceil m/N_{\max} \rceil$ ACKs and an additional ACK for the last packet. The total number of feedback messages, $N(\mathrm{MS}_{\mathrm{PROP}})$, the MS receives from its N MAs is given by

$$N(\mathrm{MS}_{\mathrm{PROP}}) \leqslant \sum_{i=1}^{N} (\lceil m/N_{\max} \rceil + 1) + m\sum_{i=1}^{N} \frac{L(\mathrm{MA}_i)_{\mathrm{cont}}}{E[A(b)]} \tag{19}$$

When all link failure probabilities are equal, such that $L(\mathrm{MA}_1)_{\mathrm{indep}} = L(\mathrm{MA}_2)_{\mathrm{indep}} = \cdots = L(\mathrm{MA}_N)_{\mathrm{indep}} = L(\mathrm{MA})$, Equation (19) can be simplified to

$$N(\mathrm{MS}_{\mathrm{PROP}}) \leqslant N(\lceil m/N_{\max} \rceil + 1) + m\sum_{i=1}^{N} \frac{L(\mathrm{MA}_i)_{\mathrm{cont}}}{E[A(b)]} \tag{20}$$

With the same assumption, the total number of SACKs messages for an SCTP, where all MAs acknowledge all packets that were received, is given by

$$N(\mathrm{MS}_{\mathrm{SCTP}}) = mN \tag{21}$$

The minimum difference $\Delta_{\min}$ between the numbers of feedback messages for the two protocols conforms to the inequality

$$\Delta_{\min} \geqslant N(m - \lceil m/N_{\max} \rceil - 1) - m\sum_{i=1}^{N} \frac{L(\mathrm{MA}_i)_{\mathrm{cont}}}{E[A(b)]} \tag{22}$$

On the other hand, the total number of NAK messages from N MAs for a legacy NAK-based protocol is given by

$$N(\mathrm{MS}_{\mathrm{NAK}}) = m\sum_{i=1}^{N} L(\mathrm{MA}_i)_{\mathrm{cont}} + m\sum_{i=1}^{N} L(\mathrm{MA}_i)_{\mathrm{indep}} \tag{23}$$

Note that the value of the second term of this equation might be double or triple because the NAK packets for the lost data packets can also be lost in the case of a link failure. Although the value depends upon the probability of lost NAKs, we assume the feedback will not be lost because our interest is finding the minimum difference between the legacy NAK-based protocol and our proposed protocol in terms of the number of feedbacks arriving at the MS.

In order to generate the packet loss probability of each MA, we employed the well-known equation of TCP throughput [27], where MSS is the maximum segment packet size, $\mathrm{RTT}_{\langle \mathrm{MS,MA}_i \rangle}$ is the round trip time from the MS to MA i, and $L(\mathrm{MA}_i)$ is the packet loss probability between the MS and MA i, such that

$$\mathrm{Throughput}(\Theta) = \frac{\mathrm{MSS}}{\mathrm{RTT}_{\langle \mathrm{MS,MA\_}i \rangle}\sqrt{L(\mathrm{MA}_i)}} \tag{24}$$

We set the throughput to 128 packets/s, MSS to 1 kB, and simulated round-trip times $\mathrm{RTT}_{\langle \mathrm{MS,MA}_i \rangle}$ as Poisson random variables, each with a mean of 40 ms. Accordingly, the average packet loss probability $L(\mathrm{MA}_i)$ for a MA i is about 0.038, and $L(\mathrm{MA}_i)_{\mathrm{cont}}$ is about 0.034, when $r = 0.9$.

DOI: 10.1002/dac

We considered that the number of transmitted packets $m = 1\,000\,000$, which roughly represents transfer of 1 GB with a packet size of 1 kB. As shown in Figure 3, for the 20 MAs, the minimum difference in the number of feedback messages from MAs between our protocol and SCTP is more than 20 000 000 when N_{max} is set to 100. The same between our protocol and legacy NAK-based protocol is about 500 000. As we can see in Figure 4, this difference becomes more prominent as the number of MAs increases. This result indicates that our protocol provides efficient buffer management functionality for the MS, by reducing the number of feedback messages sent by the MAs. This feature provides scalability, since each MS can handle more MAs.

4.2. Retransmission delay

Analyzing the performance of our protocol, we are cognizant of the fact that whether a packet transmitted from the MS is successfully received or lost due to buffer overflow or link error, or a regular NAK or *urgent* NAK message is transmitted, as notification of the packet status between the corresponding entities is not instantaneous. There is retransmission delay, and this is important in evaluating the performance of our protocol. In this subsection, we evaluate the retransmission delay of our protocol, and compare it with the legacy NAK-based protocol. For a fair comparison, we only

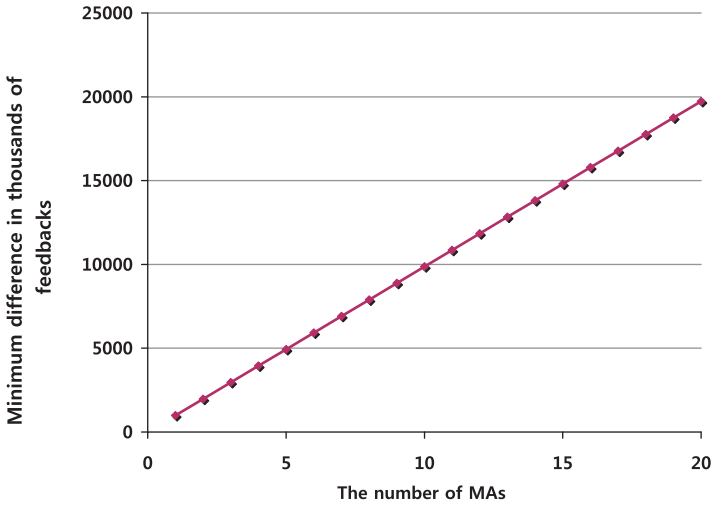

Figure 3. Minimum Δ_{min}.

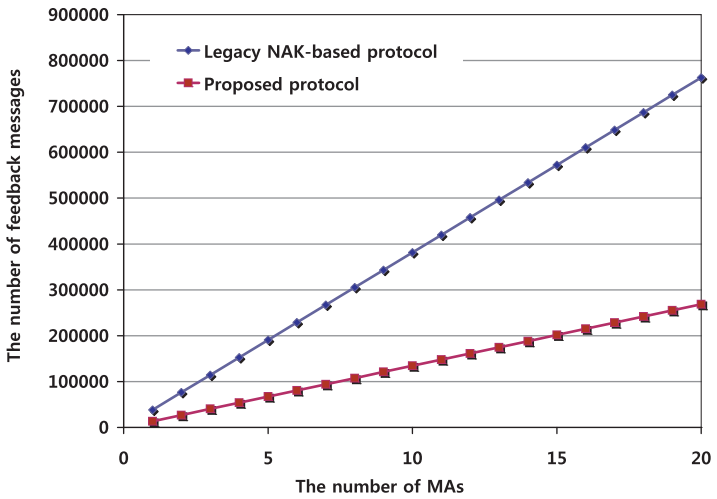

Figure 4. The number of NAK messages for different NAK-based protocols.

DOI: 10.1002/dac

consider the retransmission delay for the lost packets that is included in the error burst. In evaluating the performance, there are a few important factors, which are summarized in Table I.

If we assume that

(1) the effective link bandwidth between the MA and MRs is 4 Mbps,
(2) the packet size is 1 kB,
(3) the MS transmits packets at a send rate of 128 kbps,
(4) the NAK timer for MAi, $\mathrm{NT}_{\mathrm{MA}_i}$ is 50 ms, and
(5) the retransmission timer of MS, $\mathrm{RT}_{\mathrm{MS}}$ is 10 ms.

The average $\mathrm{OTT}_{\langle \mathrm{MS,MA}_i \rangle}$ is 20 ms, and the average inter-packet arrival time, $\Delta_T\langle \mathrm{PSN}, \mathrm{PSN}{+}1\rangle$, at the MA is 2 ms, the NAK-based protocols operate as follows. MAi transmits a regular NAK message to its MS whenever it detects a lost packet and its NAK timer $\mathrm{NT}_{\mathrm{MA}_i}$ for the packet has expired. In order to handle the pending NAK for the same packet, the MS batches NAK messages for the packet from its MAs, and retransmits the packet when $\mathrm{RT}_{\mathrm{MS}}$ expires. We assume that the NAK message is not lost. Thus, all NAK messages from the MAs arrive at the MS before the $\mathrm{RT}_{\mathrm{MS}}$ expires. The retransmission delay for the first lost packet in a error burst, $D(\mathrm{lp})$, at an MAi thus conforms to the following equation:

$$\begin{aligned} D(\mathrm{lp}) =& (\mathrm{NT}_{\mathrm{MA\_}i} - \mathrm{OTT}_{\langle \mathrm{MS,MA\_}i\rangle}) \\ &+ (\mathrm{RT}_{\mathrm{MS}} - \Delta_T[\mathrm{OTT}_{<\mathrm{MS,MA\_}i>}, \min\{\mathrm{OTT}_{\langle \mathrm{MS,MA\_}i\rangle} | 1 \leqslant i \leqslant N\}]) \\ &+ \mathrm{RTT}_{\langle \mathrm{MS,MA\_}i\rangle} \end{aligned} \tag{25}$$

In the case of the protocol without an *urgent* NAK message, the total retransmission delay for all lost packets in all error bursts, D_{NAK}, at MAi is determined via the following equation:

$$\begin{aligned} D_{\mathrm{NAK}} =& m\frac{L(\mathrm{MA}_i)_{\mathrm{cont}}}{E(A(b))}[D(\mathrm{lp}) + \Delta_T[\mathrm{PSN}, \mathrm{PSN}+1]\cdot(E(A(b))-1)] \\ &+ m\cdot L(\mathrm{MA}_i)_{\mathrm{indep}}\cdot D(\mathrm{lp}) \end{aligned} \tag{26}$$

On the other hand, our protocol enables the MS to immediately retransmit the requested packet with *urgent* NAK message without waiting for other NAKs for the same packet. Therefore, the total retransmission delay for all lost packets of MAi, $D_{\mathrm{Urgent\_NAK}}$, conforms to the equation

$$D_{\mathrm{Urgent\_NAK}} = m\frac{L(\mathrm{MA}_i)_{\mathrm{cont}}}{E(A(b))}[\mathrm{RTT}_{\langle \mathrm{MS,MA\_}i\rangle} + \Delta_T[\mathrm{PSN}, \mathrm{PSN}+1]\cdot(E(A(b))-2)] \tag{27}$$

Figure 5 shows the total retransmission delay for an MA, for both protocols. As shown, the delay at the MAs is reduced by about 5 min for 1 GB of video data. This result also significantly affects the MRs under the MA, because the requested packets from MRs are quickly retransmitted from the MA.

5. DISCUSSIONS

First, we note that the differences shown in Figures 3 and 4 increase as we increase the number of MAs. Second, we should mention that the probability distribution we used is unfavorable to our

Table I. Parameter descriptions.

Parameter	Description
$\mathrm{RT}_{\mathrm{MS}}$	The retransmission timer of the MS
$\mathrm{NT}_{\mathrm{MA}_i}$	The NAK timer for MA_i
$\mathrm{OTT}_{\langle \mathrm{MS,MA}_i\rangle}$	The one-way transmission time between the MS and MA_i
$\mathrm{RTT}_{\langle \mathrm{MS,MA}_i\rangle}$	The the round-trip time between the MS and MA_i
$L(\mathrm{MA}_i)_{\mathrm{indep}}$	Packet loss probability caused by link errors
$L(\mathrm{MA}_i)_{\mathrm{cont}}$	Packet loss probability caused by buffer overflow

DOI: 10.1002/dac

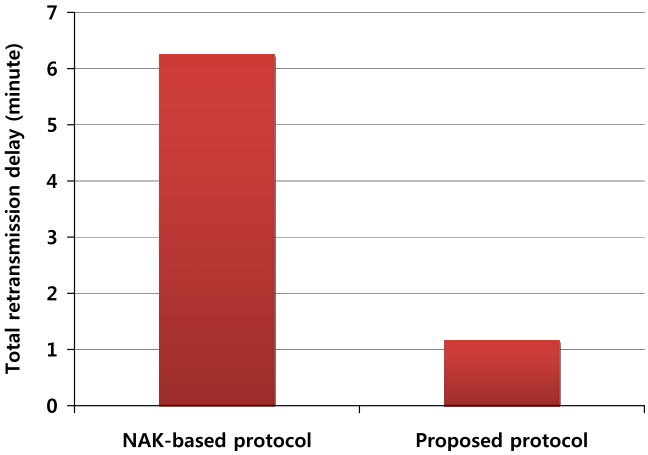

Figure 5. Retransmission delay for an MA.

protocol, because the loss probability involving a router's buffer overflow is actually much higher than 90% of all packet losses. If we increase this spatial locality factor, the difference is clearer. This is true because of the following facts:

(1) Current wireless links are somewhat reliable;
(2) Even though link errors exist, the multicast frames in the link layer are handled by a simple broadcast mechanism, where the sender does not use any backoff and feedback mechanism, and maintains its minimum contention window size until the multicast session ends. However, we also note that our protocol is a unicast-based, multicast protocol. Therefore, the error frames caused by link errors or frame collisions are locally corrected via an MAC protocol such as the IEEE 802.11a/b/g/e protocols.

Second, even though the background traffic can affect the error pattern, our error recovery is performed based upon the observed error patterns that can be obtained considering the end-to-end communication basis between the MS and its MAs. On the other hand, the link error pattern can be significantly affected by the background traffic, but those errors will be fixed using a bit-level error detection and correction technique performed between a node and any other physically connected node.

Finally, we should also mention that the legacy NAK-based protocols do not provide an efficient mechanism for buffer management. In NAK-based protocols, the MS batches NAKs for a packet and retransmits the packet periodically, provided there is a pending NAK for that packet. Let us call the period δ and assume that the packets arrive at an MS in a Poisson process with a mean arrival rate λ. If the MS has B buffers, we can define the random variable $N_A(\delta)$ to represent the number of packet arrivals at the MS within a time interval of length δ. In order to perform at least one retransmission successfully, the following condition must be satisfied, or

$$P(N_A(\delta) \geq B) = 1 - \sum_{n=0}^{B-1} \frac{(\lambda\delta)^n e^{-\lambda\delta}}{n!} = 0 \tag{28}$$

This can be simplified to

$$\sum_{n=0}^{B-1} \frac{(\lambda\delta)^n}{n!} = e^{\lambda\delta} \tag{29}$$

Since we have

$$e^{\lambda\delta} = \sum_{n=0}^{\infty} \frac{(\lambda\delta)^n}{n!} \tag{30}$$

DOI: 10.1002/dac

Equation (28) can only be satisfied when B approaches infinity.

Hence, NAK-based protocols theoretically require the MS to buffer all packets for an infinite time period, in order to achieve full coverage for all retransmission requests from the MAs. In NAK-based protocols that use a timer mechanism, the MS discards packets from its buffer after a time interval I, without considering whether these packets were successfully received by all their MAs or not. As a result, some packets can be removed from the MS buffer prematurely, while retransmission requests from some MAs remain pending. This problem has been resolved in our protocol using cumulative ACKs sent by MAs. Consequently, our proposed protocol provides faster error recovery than other protocols with minimal message overhead, while also providing an efficient buffer refreshment mechanism.

6. CONCLUSION

Clearly, the requirements for faster and time-sensitive protocols with retransmission capabilities are critical for developing multimedia transmission over WMNs. The need for retransmission is time dependant, based on the display rate of the packets involved by the MA. The advantage of the proposed protocol is two-fold, including

(1) Packet retransmission is a critical element in our proposal, due to the packet losses incurred in WMNs, especially in the mobile scenario.
(2) We allow packets to be lost when they are singletons, as retransmission in the critical timeframe for display of packets is not possible. For multiple packet losses, we utilize the *urgent* NAK message from an MA to retransmit all projected packets that may be lost, based on the weighted sum of previous packet losses.

Using this scheme, there is a high probability that packets will be retransmitted even though there is no loss or error. As a result, the packets are duplicates, and if the duplicated packets are too many, they will increase network congestion, since useless packets hog critical bandwidth. We counter unnecessary packet transmissions by establishing a threshold, which can be adjusted when the rate of packet loss exceeds the present threshold. In providing a dynamic threshold, there is a tradeoff in terms of the number of retransmissions versus the number of duplicated packets that are retransmitted. This provides the best management of bandwidth, buffer space, and error correction, where a minimal number of errors are allowed in the received packets.

We have also shown that we can reduce the number of required feedback messages for packet acknowledgements, in the case of either NAK or ACK, to a number that is significantly less than that of the comparable existent protocol SCTP. In our performance analysis, there were 20 000 000 fewer messages when we were transmitting 1 GB of data to 20 MAs, assuming a Poisson distribution for transmission times, and the probability for packet loss was set to 0.038 for link errors and 0.034 for loss due to buffer overflow resulting in packet discarding. The savings in overhead from reducing the number of feedback messages are significant.

The second advantage of our proposed protocol is a significant reduction in retransmission delay between our enhanced NAK with its prediction capability and a standard NAK protocol. Our proposed protocol saves about 5 min in 1 GB data transfer. Again, this is a significant reduction.

Lastly, we have shown where this protocol can be implemented in the application layer, without affecting the remainder of the protocol stack or requiring additional modifications to the router software to accommodate this new protocol. We also have shown that the proposed protocol permits utilization of links featuring both QoS requirements and links over unreliable connections, depending on the criticality of the data being transmitted. In order to make this protocol effective, we limited the data component of the packets to video and multimedia data. In that case, the characteristics of the data tolerate individual packet loss without significant degradation of service, in contrast to normal textual or numeric data, where time criticality is not as stringent but the loss of a single packet may be critical. Therefore, we propose this protocol in conjunction with standard data transmission protocols that provide semi-timely data transmission that is guaranteed in an end-to-end manner.

DOI: 10.1002/dac

ACKNOWLEDGEMENTS

This work was supported in part by the Ministry of Knowledge Economy, Republic of Korea, under the ITRC support program supervised by the Institute for Information Technology Advancement (IITA) and National Science Council research grant NSC97-2219-E-006-004, Taiwan.

REFERENCES

1. Baek J, Paris J-F. A heuristic buffer management and retransmission control scheme for tree-based reliable multicast. *ETRI Journal* 2005; **27**(1):1–12.
2. Shi SY, Turner JS. Routing in overlay multicast networks. *Proceedings of IEEE INFOCOM*, New York, U.S.A., June 2002; 1200–1208.
3. Banerjee S, Kommareddy C, Kar K, Bhattacharjee B, Khuller S. Construction of an efficient overlay multicast infrastructure for real-time applications. *Proceedings of IEEE INFOCOM*, San Francisco, U.S.A., April 2003; 1521–1531.
4. Baek J, Paris J-F. A scalable recovery tree construction considering packet losses correlation for reliable multicast. *KSII Transactions on Internet and Information Systems* 2008; **2**(2):82–102.
5. Kwon G-I, Byers JW. ROMA: reliable overlay multicast with loosely coupled TCP connections. *Proceedings of IEEE INFOCOM*, Hong Kong, China, March 2004; 385–395.
6. Al-Misbahi HO, Al-Aama AY. The overlay multicast protocol (OMP): a proposed solution to improving scalability of multicasting in MPLS networks. *Proceedings of the International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007)*, Sydney, Australia, August 2007; 79–84.
7. Li Y, Peng S, Chu W. K-tree trunk and a distributed algorithm for effective overlay multicast on mobile ad hoc networks. *Proceedings of the International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN 2008)*, Sydney, Australia, May 2008; 53–58.
8. Morin T. Requirements for multicast in layer 3 provider—provisioned virtual private networks (PPVPNs). *IETF RFC 4384*, April 2007.
9. Akyildiz IF, Wang XA. Survey on wireless mesh networks. *IEEE Communication Magazine* 2005; **43**(9):S23–S30.
10. Stewart R. *et al.* Stream control transmission protocol (SCTP) dynamic address reconfiguration. *IETF Internet Draft,* draft-ieft-tsvwg-addip-sctp-08.txt, June 2004.
11. Koh SJ, Park JS, Kim YJ. Framework of mobile multicast communications over wireless networks. *Proceedings of the International Conference on Ubiquitous Information Technologies and Applications (ICUT 2007)*, Bali, Indonesia, December 2007; 724–729.
12. Birman KP. *et al*. Bimodal multicast. *ACM Transactions on Computer Systems* 1999; **17**(2):41–88.
13. Costello M, McCanne S. Search party: using randomcast for reliable multicast with local recovery. *Proceedings of the IEEE ICC*, Vancouver, Canada, June 1999; 1256–1264.
14. Floyd S. *et al*. A reliable multicast framework for lightweight sessions and application-level framing. *IEEE/ACM Transactions on Networking* 1997; **5**(6):784–803.
15. Guo K, Rhee I. Message stability detection for reliable multicast. *Proceedings of the IEEE ICC*, New Orleans, U.S.A., June 2000; 814–823.
16. Kasera SK, Kurose J, Towsley D. Buffer requirements and replacement polices for multicast repair service. *Proceedings of the Network Group Communication*, Palo Alto, U.S.A., November 2000; 5–14.
17. Luby M, Vicisano L. Compact forward error correction (FEC) schemes. *IEFT RFC 3695*, February 2004.
18. Luby M, Vicisano L, Gemmell J, Rizzo L, Handley M, Crowcroft J. Asynchronous layered coding (ALC) protocol instantiation. *IETF RFC 3450*, December 2002.
19. Whetten B, Taskale G. The overview of reliable multicast transport protocol II. *IEEE Networks* 2000; **14**(1):37–47.
20. Xiao Z, Birman KP, Renesse R. Optimizing buffer management for reliable multicast. *Proceedings of the International Conference on Dependable Systems and Networks*, Washington DC, U.S.A., June 2002; 187–202.
21. Baek J, Kim C, Hong YS. Packet loss patterns adaptive feedback scheduling for reliable multicast. *Journal of Ubiquitous Convergence Technology (JUCT)* 2007; **1**(1):28–34.
22. Baek J, Kanampiu W. A NAK suppression scheme for group communications considering the spatial locality of packet losses. *IJCSNS International Journal of Computer Science and Network Security* 2006; **6**(10):158–167.
23. Baek J, Fisher PS. A comprehensive consideration for overlay multimedia multicast services in wireless mesh networks. *A Technical Report*, Department of Computer Science, Winston-Salem State University, NC, U.S.A., 2008.
24. Jung O, Park J, Kang S. Design requirement of overlay multicast session manager. *Proceedings of the International Conference on Advanced Communication Technology*, Phoenix Park, Korea, February 2005; 269–272.
25. Yong FK, Chee WT, Ramadass S. M-SCTP: transport layer multicasting protocol. *Proceedings of the National Computer Science Postgraduate Colloquium*, Penang, Malaysia, June 2005; 1–4.
26. Easton MC. Model for database reference strings based on behavior of reference clusters. *IBM Journal of Research and Development* 1978; **22**(2):197–202.
27. Mahdavi J, Floyd S. TCP-friendly unicast rate-based flow control. Available from: http://www.psc.edu/networking/papers/tcp_friendly.html [December 2004].

DOI: 10.1002/dac

AUTHORS' BIOGRAPHIES

Jinsuk Baek is Assistant Professor of Computer Science at the Winston-Salem State University (WSSU), Winston-Salem, NC. 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), Korea, in 1996 and 1998, respectively and his Ph.D. in Computer Science from the University of Houston (UH) in 2004. Dr Baek was a post doctorate research associate of the Distributed Multimedia Research Group at the UH. He acted as a consulting expert on behalf of Apple Computer, Inc in connection with Rong and Gabello Law Firm which serves as legal counsel to Apple computer. 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.

Paul S. Fisher is R. J. Reynolds Distinguished Professor of Computer Science at the Winston-Salem State University (WSSU), Winston-Salem, NC. He is the director of High Performance Computing Group at the WSSU. He received his B.A. and M.A. degrees in Mathematics from University of Utah and his Ph.D. in Computer Science from Arizona State University. He has written and managed more than 100 proposal efforts for corporations and DoD involving teams of 1 to 15 people. He worked as consultant to the U.S Army, U.S Navy, U.S Air Force and several companies over the years. In the 1990's he commercialized an SBIR funded effort and built Lightning Strike, a wavelet compression codec, then sold the company to return to academe. His current research interests include wired/wireless communication protocols, image processing and pattern recognition.

Minho Jo received the B.S. degree in industrial engineering from Chosun University, Seoul, South Korea, and the Ph.D. degree in computer networks from the Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, U.S.A. in 1994. He worked as a Staff Researcher with Samsung Electronics, South Korea, and was a Professor at the School of Ubiquitous Computing and Systems, Sejong Cyber University, Seoul. He is now a Research Professor at the Graduate School of Information Management and Security, Korea University, Seoul, South Korea. Prof. Jo is Executive Director of the Korean Society for Internet Information (KSII) and Board of Trustees of the Institute of Electronics Engineers of Korea (IEEK), respectively. He is Founding Editor-in-Chief and Chair of the Steering Committee of KSII Transactions on Internet and Information Systems, General Chair of International Ubiquitous Conference, and Co-Chair of the International Conference on Ubiquitous Convergence Technology. He is Editor of the Journal of Wireless Communications and Mobile Computing, and Associate Editor of the Journal of Security and Communication Networks published by Wiley, respectively. He serves on an Associate Editor of the Journal of Computer Systems, Networks, and Communications published by Hindawi. He served as Chairman of IEEE/ACM WiMax/WiBro Services and QoS Management Symposium, IWCMC 2008. He is Technical Program Committee of IEEE ICC 2008 & 2009 and IEEE GLOBECOM 2008 & 2009 and TPC Chair of CHINACOM 2009 Network and Information Security Symposium. His current interests lie in the area of wireless sensor networks, RFID, wireless mesh networks, security in communication networks, machine intelligence in communications, and ubiquitous and mobile computing.

Hsiao-Hwa Chen is currently a full Professor in Department of Engineering Science, National Cheng Kung University, Taiwan, and he was the founding Director of the Institute of Communications Engineering of the National Sun Yat-Sen University, Taiwan. He received BSc and MSc degrees from Zhejiang University, China, and PhD degree from University of Oulu, Finland, in 1982, 1985 and 1990, respectively, all in Electrical Engineering. He has authored or co-authored over 300 technical papers in major international journals and conferences, five books and several book chapters in the areas of communications, including the books titled 'Next Generation Wireless Systems and Networks' (512 pages) and 'The Next Generation CDMA Technologies' (468 pages), both published by John Wiley and Sons in 2005 and 2007, respectively. He has been an active volunteer for IEEE various technical activities for over 20 years. Currently, he is serving

DOI: 10.1002/dac

as the Chair of IEEE ComSoc Radio Communications Committee, and the Vice Chair of IEEE ComSoc Communications & Information Security Technical Committee. He served or is serving as symposium chair/co-chair of many major IEEE conferences, including VTC, ICC, Globecom and WCNC, etc. He served or is serving as Associate Editor or/and Guest Editor of numerous important technical journals in communications. He is serving as the Chief Editor (Asia and Pacific) for Wiley's Wireless Communications and Mobile Computing (WCMC) Journal and Wiley's International Journal of Communication Systems, etc. He is the founding Editor-in-Chief of Wiley' Security and Communication Networks journal (www.interscience.wiley.com/journal/security). He is also an adjunct Professor of Zhejiang University, China, and Shanghai Jiao Tong University, China. Professor Chen is a recipient of the Best Paper Award in IEEE WCNC 2008.

DOI: 10.1002/dac