

# STUDY ON WINDOW-BASED RELIABLE MULTICAST PROTOCOLS FOR WIRELESS LANS

Huei-Wen Ferng and Pin-Chieh Yeh

Department of Computer Science and Information Engineering  
National Taiwan University of Science and Technology  
Taipei 106, Taiwan  
E-mail: hwferng@mail.ntust.edu.tw

**Abstract** - Two reliable multicast protocols for wireless local area networks (LANs), i.e., leader-based protocol with a sliding window (LBPW) and leader-based protocol with a sliding window and  $n$ -fold acknowledgement reduction (LBPR( $n$ )), are proposed in this paper based on a previously proposed leader-based protocol (LBP). Firstly, the pipelining technique is added into LBP to form LBPW so as to make frame transmission efficient. Then, an acknowledgement reduction scheme is incorporated into LBPW to form LBPR( $n$ ) to alleviate the ACK/NAK implosion problem. Through numerical examples done by a simulation approach, we demonstrate that these two reliable multicast protocols outperform LBP.

**Keywords** - Multicast, Reliable Multicast, Wireless LAN, Window-based.

## I. INTRODUCTION

As networks dramatically grow, e.g., the pervasion of the world wide web (WWW) in the Internet, networks have made our daily lives versatile and convenient. To satisfy the needs of different applications, network transmission techniques have been developed not only for the unicast transmission technique, but also for the multicast transmission technique. In particular, the multicast technique provides an efficient means of data dissemination with wide-spread applications which may be run on the multicast backbone (Mbone) using the IP multicast [3] technique. However, original IP multicast is not adequate for some specific applications, such as stock quote dissemination, shared whiteboard [2], web cache updates, and distributed interactive system (DIS) [10] etc. due to the stringent requirement on reliability requested by these applications. Therefore, how to incorporate the reliability into the multicast technique becomes a hot issue to accommodate such applications. Issue of reliable multicast previously focused on wired networks, e.g., [4]. Recently, researchers have also gradually shifted the reliable multicast issue to wireless networks [1], [5], [6]. In this paper, we shall pay attention to the infrastructure-based wireless LAN.

This work was supported in part by the National Science Council, Taiwan under Contract NSC 90-2213-E-011-096.

To achieve reliability, automatic repeat request (ARQ) [9] and forward error correction (FEC) [11] are frequently employed in the literature. For ARQ, it is frequently used in the non-real-time applications, such as data dissemination, employing a feedback mechanism with ACKs/NAKs, while FEC is suitable in the real time applications, utilizing the error correction code to correct possible errors. Since wireless LANs mainly carry data; hence, we adopt ARQ in this paper. However, two major problems, i.e., the ACK/NAK implosion [7] and media access are confronted when using ARQ in wireless LANs. To solve the above mentioned problems, Kuri and Kasera [6] proposed three protocols: delayed feedback-based protocol (DBP), probabilistic feedback-based protocol (PBP), and LBP. As in most wireless LANs, e.g., IEEE 802.11 wireless LAN, data frames are ready to be sent after the exchange of request-to-send/clear-to-send (RTS/CTS) messages between a sender/access point (AP) and receivers. For DBP, a random timer is set for each group member (GM) once hearing the RTS sent by the AP and a CTS is sent by a GM only when no other CTS is heard. That is, the CTS suppression is employed. For PBP, it uses a random probabilistic scheme to determine whether a CTS should be sent or not; if a collision occurs, the above procedure proceeds again until the AP successfully receives a CTS. As for LBP, an elected leader rather than all GMs should take the responsibility of feedback messages. Once some other members are not ready, they send a not-clear-to-send (NCTS) to collide the possible CTS sent by the leader. Then the previous procedure should be performed again until a CTS is successfully received by the AP. As far as the phase of data frame transmission is concerned, any member who received an erroneous data frame contends the channel and sends back a NAK to the AP for both DBP and PBP, while the leader sends an ACK or a NAK according as receiving status of the data frame and other members just send NAKs when an erroneous data frame is received for LBP. Kuri and Kasera [6] showed that LBP is the best among the three protocols. In this paper, we develop reliable multicast protocols LBPW and LBPR( $n$ ) for wireless LANs based on LBP. The LBPW improves the transmission efficiency of LBP by incorporating a sliding window. Furthermore, LBPR( $n$ ) mitigates the ACK/NAK

implosion problem through the acknowledgement reduction over LBPW.

The remainder of the paper is organized as follows. In Section II, we describe protocols LBPW and LBPR( $n$ ). In Section III, we examine the performance of these two protocols through numerical experiments done by a simulation approach. Finally, Section IV concludes the paper.

## II. DESCRIPTION OF PROPOSED PROTOCOLS

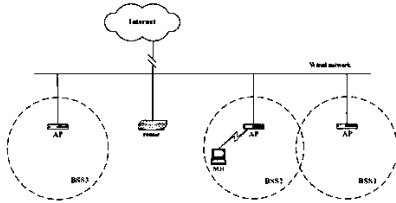


Fig. 1. The framework of a wireless LAN.

The scenario to be addressed is a wireless LAN whose channel is shared using a media access control (MAC) protocol (see Fig. 1.). In the wireless LAN, the infrastructure mode is assumed. Therefore, the basic network architecture consists of an AP and several mobile hosts. Several basic network architectures then jointly form the *wireless portion* of the wireless LAN. Considering a multicast group in the wireless LAN with one sender and several group members, the sender first transmits frames towards APs; then any AP forwards these frames to the group members within its service area. Therefore, The communication links from the sender to all group members may be logically split into two links: one from the sender to APs, the other from APs to group members. In the following, reliable multicast protocols LBPW and LBPR( $n$ ) are designed for the basic network architecture. Such an arrangement can enhance the scalability of reliable multicast protocols since the error recovery will be confined to a *local* area. It also shortens the end-to-end latency due to the local error recovery. In the following discussions, we just treat the AP as a sender with group members within its service area. Now these two protocols are described as follows.

### A. LBPW

As above mentioned, LBPW incorporates the sliding window scheme (a pipelining technique) into LBP to improve the transmission efficiency of LBP. Hence the AP can send a batch of data frames at a time. This results in less times of contention and shorter frame waiting time. Let a slot time  $t_{st}$  be the sum of processing time  $t_{pc}$ , transmission time  $t_{tr}$  required to transmit an ACK or a NAK, and propagation time  $t_{pp}$ . For simplicity, each data frame has a fixed frame length of  $f_l$  times of an ACK (or a NAK). If the window size of LBPW is  $W_S$ , then after at least one exchange of RTS/CTS, the AP can contiguously transmit at

most  $W_S$  frames to all GMs and the corresponding ACKs are then sent from the leader in response to these frames if there is no frame error. We then called the time period between the time transmitting the first RTS and the time receiving the last ACK a *cycle*. Like LBP, we assume that a leader can be elected using a leader selection algorithm [6] in LBPW (and LBPR( $n$ )). As for the resultant protocol for LBPW in a cycle, it can be depicted in two phases, i.e., phase of control message exchange and phase of data transfer, which are described as follows:

#### Phase of RTS/CTS exchange:

Event PCW<sub>1</sub> – AP to GMs (starting in slot  $k$ ): Send an RTS to all GMs.

Event PCW<sub>2</sub> – Leader/GMs to AP (in slot  $k + 1$ ):

*Leader*: Send a CTS if it is ready to receive data frames; otherwise, do nothing.

*Other GMs*: Send an NCTS if it is *not* ready to receive data frames; otherwise, do nothing.

#### Phase of data frames transfer:

Event PTW<sub>1</sub> – AP to GMs (in slot  $k + 2$ ): If a CTS was received by the AP in slot  $k + 1$ , start to transmit contiguously available  $n_a$  ( $\leq W_S$ ) data frames with labels, say, 1, 2, ...,  $n_a$ ; otherwise, go back to event PCW<sub>1</sub>.

Event PTW<sub>2</sub> – Leader/GMs to AP (during slot  $k + 2 + \lceil (f_l * t_{tr} * n_a + t_{pc} + t_{pp}) / t_{st} \rceil$  and slot  $k + 1 + \lceil (f_l * t_{tr} * n_a + t_{pc} + t_{pp}) / t_{st} \rceil + n_a$ ):

*Leader*: If the leader received the  $i$ th frame correctly, it sends an ACK in slot  $k + 1 + \lceil (f_l * t_{tr} * n_a + t_{pc} + t_{pp}) / t_{st} \rceil + i$ ; otherwise, it sends a NAK.

*Other GMs*: If the  $i$ th frame was received with error bits by any GM, it sends a NAK in slot  $k + 1 + \lceil (f_l * t_{tr} * n_a + t_{pc} + t_{pp}) / t_{st} \rceil + i$ ; otherwise, it does nothing.

In fact, the first phase is rendered from LBP, but the second phase is different from that of LBP since the pipelining technique of data transfer is utilized in this phase to improve the transmission efficiency.

Based on feedbacks from GMs, the AP should make a decision about whether these frames are required to retransmit or not. We note that the AP faces the following three cases when receiving feedbacks from GMs: i) an ACK is received; ii) nothing is received; iii) a collision of frames caused by an ACK and NAKs is detected. For the first case, the corresponding frame is indeed correctly received by all GMs, while for the remaining two cases, the frame is either not correctly received by the leader or other GMs. Hence, frames without feedbacks of the first case are retransmitted in the next cycle with other queued frames.

### B. LBPR( $n$ )

Another reliable multicast protocol LBPR( $n$ ) which is a modified version of LBPW to reduce the number of ACKs/NAKs by a factor of  $n$  via aggregating the receiving

status of at most  $n$  frames using a bit map put into an ACK sent by the leader or a NAK sent by other GMs once at least one frame is required to retransmit. The parameter  $n$  is called the reduction ratio since every  $n$  frames (if available) are acknowledged by the leader or other GMs using only one ACK or NAK. For the leader, the aggregated bit map is used to denote the receiving status of these  $n$  frames; but for other GMs, they do nothing when these  $n$  frames are correctly received or just reply a NAK when at least one frame is required to retransmit. Based on the above mechanism, the ACK/NAK implosion problem is further alleviated since less ACKs or NAKs are generated compared to LBPW (or LBP). From the above description, we note that as the reduction ratio  $n$  increases, it may improve the performance of LBPR( $n$ ) due to the amount reduction, while higher reduction ratio may cause more correctly received frames to be retransmitted once a collision is detected. Hence, it is necessary to select a proper reduction ratio to gain better system performance. The detailed algorithm for LBPR( $n$ ) in a cycle is described as follows:

**Phase of RTS/CTS exchange:** The operation is same as that in LBPW.

**Phase of data frames transfer:**

Event PTR<sub>1</sub> – AP to GMs (in slot  $k + 2$ ): If a CTS was received by the AP in slot  $k + 1$ , start to transmit contiguously available  $n_a$  ( $\leq W_S$ ) data frames with labels, say, 1, 2, ...,  $n_a$ ; Otherwise, go back to event PCW<sub>1</sub>.

Event PTR<sub>2</sub> – Leader/GMs to AP (during slot  $k + 2 + \lceil (f_l * t_{tr} * n_a + t_{pc} + t_{pp}) / t_{st} \rceil$  and slot  $k + 1 + \lceil (f_l * t_{tr} * n_a + t_{pc} + t_{pp}) / t_{st} \rceil + \lceil n_a / n \rceil$ ):

*Leader:* Send an acknowledgement in the bit map manner including the receiving status for at most  $n$  frames at a time. Hence  $\lceil n_a / n \rceil$  times of acknowledgements are required to send during slot  $k + 2 + \lceil (f_l * t_{tr} * n_a + t_{pc} + t_{pp}) / t_{st} \rceil$  and slot  $k + 1 + \lceil (f_l * t_{tr} * n_a + t_{pc} + t_{pp}) / t_{st} \rceil + \lceil n_a / n \rceil$ .

*Other GMs:* Break the  $n_a$  frames into  $\lceil n_a / n \rceil$  subsegments each including exactly  $n$  frames except the last one. If one of frames for subsegment  $i$  was received with error bits by any GM, it sends a NAK directly in slot  $k + 1 + \lceil (f_l * t_{tr} * n_a + t_{pc} + t_{pp}) / t_{st} \rceil + i$ ; otherwise (i.e., all frames for the  $i$ th subsegment are correctly received by the GM), it does nothing.

We note that the AP faces the following two cases when receiving feedbacks from GMs: i) an ACK with bit map is received; ii) a collision of frames caused by an ACK and NAKs is detected. For the first case, only indicated erroneous frames are retransmitted, but all frames are retransmitted for the second case.

### III. NUMERICAL EXPERIMENTS AND DISCUSSIONS

As mentioned in Section II, the minimal wireless LAN consisting of an AP and several mobile hosts is considered

in the following numerical experiments. For simplicity, the operation of the MAC protocol is neglected and perfect time synchronization is assumed such that each operation works on a slotted time axis. Moreover, only one multicast group in the wireless LAN is considered. For each group member, it is assumed to be *always ready* to receive data frames from the AP, i.e., NCTS will not be used under such an assumption. As for the characteristics of the wireless channel, we assume that data frames may be received in error but never get lost over the channel, while control frames, such as RTS, CTS, ACK, NAK are always correctly received. Since the propagation time for the wireless LAN is negligible compared to the transmission time, it is simply set to zero. Also, the processing time is set to zero for simplicity. Hence, only the effect caused by frame transmission time and frame queuing time in an infinite buffer endowed at the AP for temporary queuing of frames are considered here. To model the simulation system, we assume that frames to be sent to all group members are generated according to a Poisson process with rate  $\lambda$  per slot and a uniform distribution of batch size  $U(b_{\min}, b_{\max})$ , where  $\lambda = 1/710$ ,  $b_{\min} = 5$ , and  $b_{\max} = 15$ . In the following experiments, parameters to be adjusted include at least number of group members  $n_{GM}$ , frame error probability (FEP), frame length (in slots). If there is no error correcting code is employed, bit error rate (BER) relates to FEP via the relation  $FEP = 1 - (1 - BER)^{F_l}$ , where  $F_l$  represents the frame length in bits. We note that the above relation can be approximated as  $FEP \approx F_l * BER \propto f_l * BER$  if BER is quite small. In the following numerical experiments, we set  $f_l$  to 10 and 20 for the comparison purpose.

In these numerical experiments, performance metrics to be measured are cost, average queuing delay, average queue length, and exposure when comparing LBPW to LBP. Here, cost and exposure are defined as follows.

- Cost: the average time lasting since the AP contends the channel until the AP ascertains that all group members correctly receive the frame.
- Exposure: ratio of the number of mobile hosts actually receiving the frame and the number of mobile hosts who do need the frame.

We note that the reciprocal of the cost is proportional to the throughput and the exposure measures the effect of annoyance when a frame is sent. To compare the performance among LBP, LBPW, and LBPR( $n$ ), the following metric is further gauged.

- Number of ACKs or NAKs: the average number of ACKs or NAKs required until a frame is correctly received.

Now let us begin the discussion on numerical experiments. In Fig. 2, the average cost for various window size  $W_S$  varying among 1, 2, 3, 5, 10 is shown under  $FEP = 0.05$ . From this figure, one can easily find that the increase of the window size  $W_S$  causes a lower cost, i.e., higher throughput.

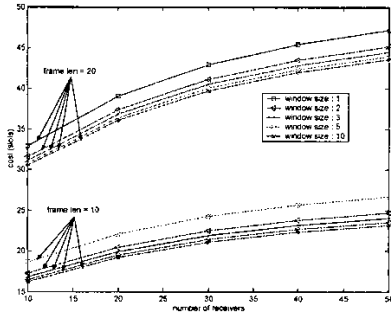


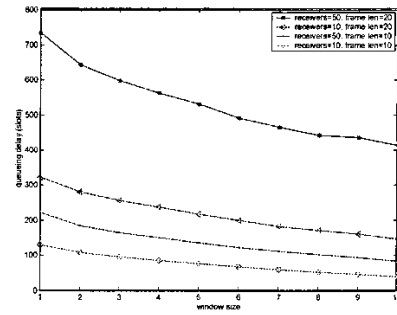
Fig. 2. Cost vs. number of group members of LBPW.

The reductions of costs when  $W_S = 2$  and  $W_S = 10$  compared to those of LBP (i.e., LBPW when  $W_S = 1$ ) are 4.3 % and 7.0 %, respectively, when  $f_l = 20$  and  $n_{GM} = 10$  and 7.3 % and 13.3 %, respectively, when  $f_l = 10$  and  $n_{GM} = 10$ . These results evidently show that LBPW with large window size, say 10, performs much better than LBP. We also see from Fig. 2 that the cost increases as the number of group members goes up. In Fig. 3, other performance measures including queueing delay, average queue length, and exposure are exhibited for various window size, number of group members, and frame length. From Fig. 3(a)/Fig. 3(b), we see that the queueing delay/the average queue length goes down as the window size increases or the number of group members decreases. This phenomenon is quite crystal-clear because the increase of the window size raises the transmission efficiency due to the pipelining effect and it is easier to handle the error recovery for a small group. As for the exposure shown in Fig. 3(c), it is not affected by the increase of the window size but it increases as the group grows up. From the above observations, we see that the attainable transmission efficiency coming from the pipelining technique can be over 10 %.

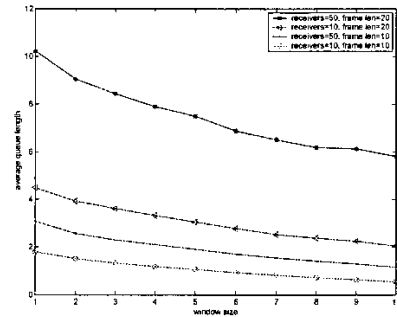
We now further examine the performance of LBPR( $n$ ). In the following experiments, we fix  $n_{GM} = 50$  and  $W_S = 12$ . Let us first look at the average number of ACKs/NAKs required to successfully transmit a frame in Table 1 which reveals that LBPR( $n$ ) achieves ACKs/NAKs reduction approximately by a factor of  $n$  compared to LBP or LBPW. As shown in Table 2, we first note that as FEP decreases, the exposure of LBP/LBPW increases because most of group

Table 1  
The number of ACKs/NAKs for different protocols with  $f_l = 10$ ,  $n_{GM} = 50$  and,  $W_S = 12$ .

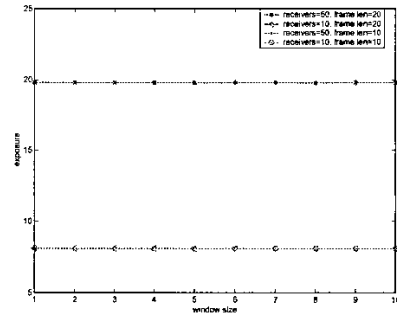
FEP	LBP/LBPW	LBPR(2)	LBPR(3)	LBPR(4)
0.1	2.695	1.231	0.775	0.559
0.05	2.255	1.086	0.712	0.529
0.01	1.507	0.741	0.492	0.373
0.001	1.154	0.601	0.411	0.321



(a) Queueing delay.



(b) Average queue length.



(c) Exposure.

Fig. 3. Performance measures vs. window size of LBPW.

members will successfully receive a frame and only a small portion of group members will receive that frame in error. For LBPR( $n$ ), the situation gets worse once a NAK is sent by a group member since the group member requests retransmissions not only for the erroneous frames but also for other correctly received frames for itself. As the reduction ratio increases, the exposure gets further worse as shown in Table 2. From the above discussions, we see that although LBPR( $n$ ) reduces the number of ACKs/NAKs, it causes a larger value of exposure. Hence the reduction ratio  $n$  should be properly chosen. In the following, we further investigate cost and queueing delay of LBPR( $n$ ). Table 3 shows the performance of cost. We see that LBPR( $n$ ) performs better than LBPW due to the saving of ACKs/NAKs. The pairs of

Table 2  
Exposure for different protocols with  $f_i = 10$ ,  $n_{GM} = 50$   
and,  $W_S = 12$ .

FEP	LBP/LBPW	LBPR(2)	LBPR(3)	LBPR(4)
0.1	12.944	14.915	16.365	17.498
0.05	19.798	22.762	24.452	26.001
0.01	40.448	66.406	82.927	93.648
0.001	48.703	98.829	140.099	195.277

reduction ratio and percentage of cost reduction compared to LBPW are (2, 12.7), (3, 18.8), (4, 22.8) when FEP = 0.1 and  $f_i = 20$ ; they are (2, 7.5), (3, 10.5), (4, 12.2) when FEP = 0.01 and  $f_i = 20$ . Hence the cost reduction is more obviously when FEP is high. As for the queuing delay, it is clear that a longer frame length causes longer queuing delays as shown in Table 4. From Table 4, we also observe the followings: i) as the reduction ratio increases, longer queuing delays are incurred since more rates of retransmission are incorporated; ii) LBPR(2) or LBPR(3) may have less queuing delays than LBPW when FEP is kept quite low; iii) queuing delays of LBP are longer than LBPW and LBPR( $n$ ).

#### IV. CONCLUSIONS

This paper proposes two types of reliable multicast protocols LBPW and LBPR( $n$ ) for the multi-access wireless LAN. Through numerical comparisons, we demonstrate that: i) the cost of LBPR( $n$ ) is lower than that of LBPW which is subsequently lower than LBP. The attainable cost reduction of LBPW compared to LBP can be over 10 %. As the window size goes up, the reduction can be further improved; ii) both LBPW and LBPR( $n$ ) perform better than LBP in respect to queuing delay. Further distinguishing the performance of LBPW and LBPR( $n$ ), we note that LBPW mostly performs better than LBPR( $n$ ) for  $n \geq 3$ , while LBPR(2) performs better than LBPW when the frame loss probability is low; iii) as for the exposure metric, LBPW is the same as LBP and smaller than LBPR( $n$ ). For larger  $n$ , the exposure of LBPR( $n$ ) becomes higher. Based on the above observations, we suggest LBPW and LBPR(2) to be employed to fulfill the reliability of multicast protocols in the wireless LAN.

#### REFERENCES

- [1] G. Anastasi, A. Bartoli, and F. Spadoni, "A reliable multicast protocol for distributed mobile systems: design and evaluation," *IEEE Trans. Parallel and Distributed Systems*, vol. 12, pp. 1009-1022, 2001.
- [2] J. L. Bakker and H. J. Batteram, "Design and evaluation of the distributed software component framework for distributed communication architectures," in *Proc. IEEE EDOC '98*, pp. 282-288, Nov. 1998.
- [3] S. E. Deering, "Host extensions for IP multicasting," *IETF RFC 1112*, Aug. 1988.

Table 3  
Cost for different protocols with  $n_{GM} = 50$  and  $W_S = 12$ .

FEP, $f_i$	LBP	LBPW	LBPR(2)	LBPR(3)	LBPR(4)
0.1, 20	56.350	51.958	45.423	42.170	40.130
0.05, 20	47.161	43.516	39.755	38.040	36.775
0.01, 20	31.515	29.146	26.973	26.072	25.597
0.001, 20	24.131	22.311	21.740	21.506	21.384
0.1, 10	31.861	27.570	23.712	21.811	20.693
0.05, 10	26.652	23.088	20.700	19.641	18.929
0.01, 10	17.813	15.475	14.005	13.430	13.136
0.001, 10	13.639	11.835	11.294	11.084	10.978

Table 4  
Queuing delay for different protocols with  $n_{GM} = 50$  and  
 $W_S = 12$ .

FEP, $f_i$	LBP	LBPW	LBPR(2)	LBPR(3)	LBPR(4)
0.1, 20	1414.629	728.583	803.760	968.985	1181.323
0.05, 20	730.912	358.289	370.832	382.980	407.798
0.01, 20	293.969	110.027	126.457	146.793	158.826
0.001, 20	187.207	63.667	62.679	65.441	68.943
0.1, 10	298.151	96.212	83.881	84.450	86.798
0.05, 10	220.335	63.884	57.902	57.250	58.606
0.01, 10	121.789	27.002	27.268	29.041	30.777
0.001, 10	85.380	18.764	17.293	17.433	17.506

- [4] S. Floyd, V. Jacobson, C. G. Liu, S. McCanne, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 784-803, Dec. 1997.
- [5] T. G. Harrison, C. L. Williamson, W.L. Mackrell, and R.B. Bunt, "Mobile multicast (MoM) protocol: multicast support for mobile hosts," in *Proc. ACM/IEEE Mobicom '97*, pp. 151-160, 1997.
- [6] J. Kuri and S. K. Kasera, "Reliable multicast in multi-access wireless LANs," *Wireless Networks*, vol. 7, pp. 359-369, 2001.
- [7] C. Metz, "Reliable multicast: when many must absolutely positively receive it," *IEEE Internet Computing*, pp. 9-13, Jul.-Aug. 1998.
- [8] I. Nikolaidis and J. J. Harms, "A logical ring reliable multicast protocol for mobile nodes," in *Proc. IEEE ICNP '99*, pp. 106-113, 1999.
- [9] B. W. On and M. S. Park, "A reliable multicast protocol in networks with mobile hosts," in *Proc. DOA '00*, pp. 27-35, 2000.
- [10] J. M. Woodyard and D. C. Reif, "Long haul participation in a distributed interactive simulation demonstration," in *Proc. IEEE NAECON '95*, vol. 2, pp. 810-816, 1995.
- [11] S. W. Yuk and D. H. Cho, "Parity-based reliable multicast method for wireless LAN environments," in *Proc. IEEE VTC '99*, pp. 1217-1221, 1999.