SPARM: Spatially Pipelined ACK Aggregation for Reliable Multicast in Directional MAC

Laihyuk Park, Changun Lee, and Sungrae Cho

***Abstract*—The use of a directional antenna is a promising technique for the provision of high speed wireless local area networks such as IEEE 802.11ad. In this paper, we propose an ACK-based reliable multicast protocol for directional antennas referred to as spatially pipelined ACK aggregation for reliable multicast (SPARM). To resolve the problems of ACK implosion and ACK collection latency, the SPARM exploits (1) spatial reuse and (2) pipelining of ACK aggregation. In the SPARM, receivers sequentially aggregate their ACKs while the sender multicasts a data frame to the next beam, both removing the ACK implosion problem and reducing ACK collection latency. In this paper, we prove that the aggregation process does not interfere with the sender multicasting to the next beam. Performance evaluation shows that the proposed SPARM has full reliability and outperforms the existing schemes with respect to the throughput by about 200%.**

***Index Terms*—Directional MAC and reliable multicast**

I. Introduction

THERE has been considerable attention paid recently to very high speed wireless local area networks such as IEEE 802.11ad, through which we can envision more than 1Gbps data service. IEEE 802.11ad enables operation in the 60 GHz frequency band capable of very high throughput using directional antennas. One of the potential services using IEEE 802.11ad WLANs is reliable multicasting. Multicasting service is particularly important in many applications, such as updating software, sending an alarm signal, etc. Especially, future smart devices will proliferate proximity-based social network applications exploiting multicasting services. Although a reliable multicast protocol for directional antennas does not exist in the literature, a significant number of studies have been performed for reliable multicasting in wireless networks [3] [4] [8] based on ACK, NAK, or both.

The slot reservation-based reliable broadcast protocol (SRB) [8] was proposed to add a reliability component to the existing multicast protocol in the IEEE 802.11 MAC. This scheme is based on positive acknowledgments (ACK). In SRB, a sender multicasts a frame to all receivers, and ACKs from all receivers are scheduled in order to avoid the ACK implosion problem. The sender maintains a bitmap table that keeps track of unacknowledged receivers by indicating '1.' To guarantee full reliability, the sender retransmits the original multicast frame with the bitmap so that only the unacknowledged receivers can correct the errors. In SRB, since all receivers need to transmit ACKs, the delay will be significant and thus the throughput might be degraded. This problem is exacerbated as the number of receivers increases.

On the contrary, NAKs are well established as an effective loss advertisement mechanism in multi-hop wireless networks. Cooperative loss recovery for reliable multicast in ad hoc networks (CoreRM) [3] is a NAK-based scheme where the NAK frames are scheduled by random timers to avoid NAK implosion. Since one NAK is sufficient for the sender to be aware that an error has occurred, retransmission of the original frame informs the receivers with later NAK timers, and thus they can suppress their timer. However, NAKs cannot handle the unique cases of all frames being lost at a particular receiver. Since the receiver is not aware that a data frame is expected, it cannot possibly advertise a NAK to request retransmission. For short message types, such as queries consisting of a few frames, the probability that a receiver does not receive any frame in a message is not negligible.

To tackle the above problems, the reliable and efficient multicast protocol (REMP) [4] was proposed. In this scheme, a multicast group leader is elected for transmission of an ACK to the sender if it successfully receives a multicast frame. This ACK transmission is performed only by the leader. In contrast, both the leader and other receivers transmit NAKs if erroneous frames are received. Since their scheme exploits both ACK and NAK, we refer to this scheme as a *hybrid* scheme. However, this scheme does not correct the problem where the leader and most of receivers successfully receive an announcement frame (MTA in [4]) while some other receivers miss the frame or non-leader's NAKs are lost due to wireless channel error while leader's ACK is successfully received.

In this paper, we propose an ACK-based[1] reliable multicast protocol for directional antennas referred to as spatially pipelined ACK aggregation reliable multicast (SPARM). As mentioned earlier, ACK-based schemes suffer from the ACK implosion problem and ACK collection latency. However, our protocol resolves the ACK implosion problem and reduces the latency of ACK aggregation, and thus increases the throughput by exploiting spatial reuse of the directional antennas and pipelining of ACK aggregation. In the wireless sensor network domain, there have been extensive research efforts devoted to data aggregation [6]. However, our scheme is different from the data aggregation technique in that the receivers relay ACKs as long as the receivers do not interfere with the sender. The rest of this paper is organized as follows. The SPARM is described in section II in detail followed by the performance of our scheme in section III. Finally, conclusions are drawn in section IV.

II. SPARM Scheme

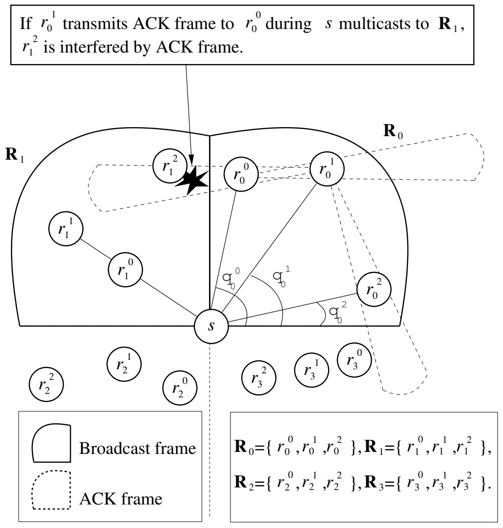
We assume a switched beam antenna system where K beam patterns are ideally non-overlapping. For directional communication, the other beams are blocked when one beam is transmitting or receiving. Therefore, we do not consider simultaneous multicasting from all K beams. Instead, we assume sequential multicasting from beam 0 to $K-1$. As in Fig. 1, we assume a sender s stores angle θ_i^j of receiver r_i^j (the jth receiver at the ith beam of the sender). This information is obtained by direction of arrival (DoA) estimation from a hello frame of r_i^j. Moreover, the sender estimates the relative distance l_i^j to r_i^j by measuring the received power of the hello frame with transmit power information. We further denote the sender's multicast group as $\mathbf{R}_i$ for beam i ($\mathbf{R}_i = \{r_i^0, \cdots, r_i^{n_i-1}\}$), where n_i is the number of receivers in $\mathbf{R}_i$. Besides the above assumptions, multicast receivers use a much narrower beam as in Fig. 1 (e.g., narrow *pencil beam* [2]) than the multicast sender. To provide the narrower beam, an accurate DoA estimation mechanism is required [5].

Manuscript received July 13, 2012. The associate editor coordinating the review of this letter and approving it for publication was Qi Cheng.

This research was supported by Basic Science Research Program through the NRF funded by Ministry of Education, Science and Technology (NRF-2010-0024669).

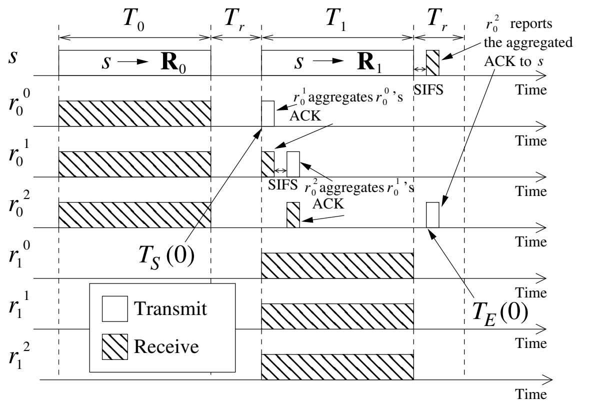
Laihyuk Park, Changun Lee, and Sungrae Cho are with the School of Computer Science and Engineering, Chung-Ang University, South Korea; Email: srcho@cau.ac.kr; Tel: +82-2-820-5766; Fax: +82-2-824-1394.

[1]As mentioned, NAK or hybrid approaches do not guarantee full reliability.



If r_0^1 transmits ACK frame to r_0^0 during s multicasts to $\mathbf{R}_1$, r_1^2 is interfered by ACK frame.
$\mathbf{R}_1$
$\mathbf{R}_0$
Broadcast frame
ACK frame
$\mathbf{R}_0$={ r_0^0, r_0^1, r_0^2 }, $\mathbf{R}_1$={ r_1^0, r_1^1, r_1^2 },
$\mathbf{R}_2$={ r_2^0, r_2^1, r_2^2 }, $\mathbf{R}_3$={ r_3^0, r_3^1, r_3^2 }.

Fig. 1. An exemplary network topology ($K = 4$).



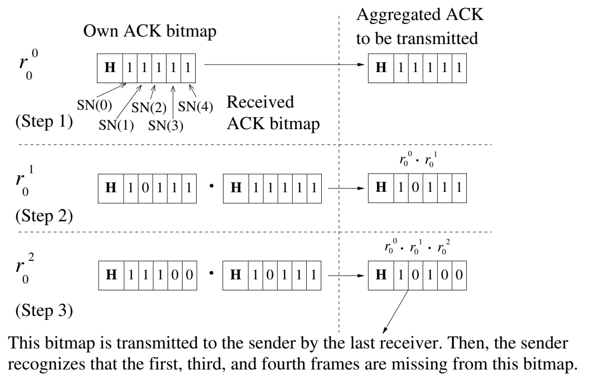
T_0
T_r
T_1
T_r
r_0^2 reports the aggregated ACK to s
$s \rightarrow \mathbf{R}_0$
$s \rightarrow \mathbf{R}_1$
SIFS
r_0^1 aggregates r_0^0's ACK
SIFS
r_0^2 aggregates r_0^1's ACK
$T_S(0)$
$T_E(0)$
Transmit
Receive
Time

Fig. 2. The SPARM scheduling for the scenario in Fig. 1 (T_0 and T_1 are multicasting periods for beam 0 and 1, respectively; and T_r is the reporting duration of the aggregated ACKs).

A. SPARM Scheduling

In the SPARM scheme, s multicasts data frame(s) in circular fashion from $\mathbf{R}_0$ to $\mathbf{R}_{K-1}$. After multicasting data frame(s), s needs to receive ACKs, since SPARM is an ACK-based scheme. Our question is when would be the optimum time to transmit ACKs? The ACK transmission can be done after data multicasting is completed up to $\mathbf{R}_{K-1}$. However, it will be more efficient for the receivers to transmit ACKs while the sender is multicasting as long as the receivers do not interfere with the multicast sender, resulting in reduced multicasting time. Therefore, in the SPARM scheme, $\mathbf{R}_i$ aggregates[2] ACKs while s multicasts a data frame at beam $(i+1)\%K$ for i=$\{0,1, \cdots, K-1\}$ in order to reduce ACK collection latency. As in Fig. 2, s multicasts a data frame to $\mathbf{R}_0$ during T_0. Then, s switches its beam to 1 and multicasts a data frame to $\mathbf{R}_1$ for T_1 during which $\mathbf{R}_0$ aggregates ACKs (a detailed description of Fig. 2 is presented later in this section).

Since there can be multiple outstanding multicast frames that are not acknowledged yet, the sender needs to keep track of them. Each receiver maintains a bitmap for multicast frames where the bitmap is used to indicate whether or not the ith frame is successfully received ("1" indicates "success"). By ANDing these bitmaps,[3] the last receiver will be aware of which frames are successfully received at all receivers. Fig. 3 shows the ACK aggregation process using a bitmap for the scenario in Fig. 1. A bitmap frame consists of a bitmap header $\mathbf{H}$ and a bitmap. $\mathbf{H}$ contains the first sequence number (SN) of the bitmap and the number of outstanding frames. Also, the ith bit of the bitmap indicates whether or not the ith frame is successfully received. As in Fig. 3, r_0^1 will recognize the first frame is missing by ANDing its own bitmap and the bitmap transmitted from r_0^0. Likewise, r_0^2 will recognize that the first, third and fourth frames are missing by ANDing its own bitmap and the bitmap transmitted from r_0^1. Most ACK-based reliable multicast protocols, including SRB [8], maintain a bitmap table where each bit indicates whether or not the corresponding receiver successfully received a multicast frame. However, SPARM does not maintain a bitmap for receivers, and therefore our bitmap scheme is scalable and independent of the number of receivers.



Own ACK bitmap
Aggregated ACK to be transmitted
r_0^0
H 1 1 1 1 1
SN(0) SN(1) SN(2) SN(3) SN(4)
Received ACK bitmap
(Step 1)
r_0^1
H 1 0 1 1 1 · H 1 1 1 1 1
$r_0^0 \cdot r_0^1$
H 1 0 1 1 1
(Step 2)
r_0^2
H 1 1 1 0 0 · H 1 0 1 1 1
$r_0^0 \cdot r_0^1 \cdot r_0^2$
H 1 0 1 0 0
(Step 3)
This bitmap is transmitted to the sender by the last receiver. Then, the sender recognizes that the first, third, and fourth frames are missing from this bitmap.

Fig. 3. ACK aggregation using bitmap (we assume the first sequence number of the bitmap is 0; and ACK aggregation is scheduled from r_0^0 to r_0^1 and then from r_0^1 to r_0^2).

The order of ACK aggregation requires additional considerations. Suppose r_0^1 and r_0^0 are the initial and the second receivers to aggregate their ACKs, respectively, in Fig. 1. Then, r_0^1 will transmit its ACK to r_0^0. As can be observed from Fig. 1, r_0^1 will interfere with r_1^2 while s is multicasting at $\mathbf{R}_1$. Thus, the order of ACK aggregation should be carefully designed. If we schedule the ACK aggregation such that r_0^0 to r_0^1, r_0^1 to r_0^2, and r_0^2 to s, there will be no interference at $\mathbf{R}_1$. Of course, the ACK aggregation from r_0^2 to s has to be delayed until the multicasting to $\mathbf{R}_1$ finishes.

Fig. 2 shows the SPARM scheduling for the scenario in Fig. 1. During the time when s multicasts a frame to $\mathbf{R}_1$, r_0^0, r_0^1, and r_0^2 in $\mathbf{R}_0$ aggregate their ACKs. The last receiver of aggregation (r_0^2) waits until s finishes transmission to $\mathbf{R}_1$. After multicasting from s to $\mathbf{R}_1$, r_0^2 reports the aggregated ACKs to s. As can be seen from this example, the sender receives only one ACK report, and thus our scheme is independent of the number of receivers. This feature of the SPARM resolves the ACK implosion problem.

In the SPARM, this aggregation scheduling is determined by the header of the sender's data frame. To make a decision on the order of ACK aggregation, s needs to consider the following two policies.

Policy 1: The aggregation schedule should be sorted in descending order of angles, i.e., if $\theta_i^a > \theta_i^b$, r_i^b aggregates r_i^a's ACK. For example, ACK aggregation from r_0^0 to r_0^1 does not interfere with $\mathbf{R}_1$ as shown in Fig. 1. Otherwise, ACK aggregation from r_0^1 to r_0^0 interferes with r_1^2 in $\mathbf{R}_1$.

[2]Sending ACKs directly to the multicast sender may cause interference with the sender multicasting a data frame to another beam. Therefore, instead, receivers cooperatively merge their ACKs as long as they do not interfere with the sender. Once merged, the ACKs are transmitted directly to the sender when the sender finishes its multicasting to another beam (e.g., in T_r as in Fig. 2). Here, we define this merging process as aggregation.

[3]Implementation of ACK aggregation is practically performed by a bit-wise AND operation of bitmaps in the SPARM scheme.

Policy 2: If $\theta_i^a = \theta_i^b$, the aggregation schedule should be sorted in ascending order of distance l_i^j from the sender. In Fig. 1, r_1^0 to r_1^1 have the same angles and $l_1^0 < l_1^1$. In this case, ACK aggregation from r_1^0 to r_1^1 does not interfere with s and $\mathbf{R_2}$ while aggregation from r_1^1 to r_1^0 does.

We will prove **Policy 1** and **Policy 2** do not interfere with the sender's transmission in *Lemma 1* and 2, respectively.

Lemma 1: If $\theta_i^a > \theta_i^b$, the aggregation from r_i^a to r_i^b does not interfere with multicasting from s to $\mathbf{R}_{i+1}$.

Proof: We assume s is located at the origin in the Cartesian coordinate system. We further assume θ_B is the beam width where $\theta_B = 2\pi/K$. Without loss of generality, suppose that the region of $\mathbf{R}_i$ is located between the x-axis and a linear function of $y = (\sin\theta_B/\cos\theta_B)x$, where $x > 0$ and $y > 0$. For the special case of $K = 4$, the region of $\mathbf{R}_i$ is located in the first quadrant. The linear function (either $y = (\sin\theta_B/\cos\theta_B)x$ or y-axis) is a border line between $\mathbf{R}_i$ and $\mathbf{R}_{i+1}$. We denote $\overrightarrow{r_i^a r_i^b}$ as the aggregation from r_i^a to r_i^b, and geometrically $\overrightarrow{r_i^a r_i^b}$ is realized as a vector representation. Then, we need to show that $\overrightarrow{r_i^a r_i^b}$ does not pass the region of $\mathbf{R}_{i+1}$ in order to prove *Lemma 1*.

Considering the polar coordinate system, let r_i^a be located at two polar coordinates (l_i^a, θ_i^a). Since r_i^a can be converted to two Cartesian coordinates $(l_i^a\cos\theta_i^a, l_i^a\sin\theta_i^a)$, we need to consider the direction from r_i^a to r_i^b for the following two cases:

- Case 1: $l_i^a\cos\theta_i^a < l_i^b\cos\theta_i^b$ and
- Case 2: $l_i^a\cos\theta_i^a \geq l_i^b\cos\theta_i^b$.

In Case 1, r_i^b is on the right of r_i^a. If $\overrightarrow{r_i^a r_i^b}$ is directed downward, $\overrightarrow{r_i^a r_i^b}$ does not pass the border line between $\mathbf{R}_i$ and $\mathbf{R}_{i+1}$. Therefore, the aggregation from r_i^a to r_i^b does not interfere with multicasting from s to $\mathbf{R}_{i+1}$.

Now, if $\overrightarrow{r_i^a r_i^b}$ is directed upward, the slope of $\overrightarrow{r_i^a r_i^b}$ should be less than θ_B to avoid passing the border line. We transform the $\overrightarrow{r_i^a r_i^b}$ to a linear function. Then, the y-intercept (ζ) of the linear function is given by

$$\zeta = l_i^a\sin\theta_i^a - \frac{l_i^a\cos\theta_i^a(l_i^b\sin\theta_i^b - l_i^a\sin{\theta_i}^a)}{l_i^b\cos\theta_i^b - l_i^a\cos\theta_i^a}. \quad (1)$$

If $\zeta \leq 0$, the slope of $\overrightarrow{r_i^a r_i^b}$ can be larger than θ_B. Therefore, the sufficient condition that $\overrightarrow{r_i^a r_i^b}$ does not pass $\mathbf{R}_{i+1}$ is $\zeta > 0$. In order to be $\zeta > 0$, the following condition should be satisfied:

$$\frac{\sin\theta_i^a}{\sin\theta_i^b} \cdot \frac{\cos\theta_i^b}{\cos\theta_i^a} > 1. \quad (2)$$

Of course, (2) is satisfied since $\sin\theta_i^a > \sin\theta_i^b$ and $\cos\theta_i^b > \cos\theta_i^a$ when $(\pi/2) > \theta_i^a > \theta_i^b > 0$.

In Case 2, the x coordinate of r_i^a is greater than or equal to that of r_i^b. In this case, $\overrightarrow{r_i^a r_i^b}$ is never directed upward since $(\pi/2) > \theta_i^a > \theta_i^b > 0$. To avoid passing $\mathbf{R}_{i+1}$ in Case 2, ζ should be negative; otherwise $\overrightarrow{r_i^a r_i^b}$ can pass $\mathbf{R}_{i+1}$. To be $\zeta < 0$, (2) should be satisfied[4]. As described in Case 1, (2) is true. Therefore, the aggregation from r_i^a to r_i^b does not interfere with multicasting from s to $\mathbf{R}_{i+1}$. ∎

Lemma 2: If $\theta_i^a = \theta_i^b$ and $l_i^a < l_i^b$, aggregation from r_i^a to r_i^b does not interfere with multicasting from s to $\mathbf{R}_{i+1}$.

[4]It seems that the inequality sign should be inverted. However, (2) is still true since the condition for Case 2 (i.e., the relation between $l_i^a\cos\theta_i^a$ and $l_i^b\cos\theta_i^b$) also has been inverted.

```
for $i = 0$ to $K-1$ do {For all beams}
    Perform ACK aggregation scheduling at beam $i$
    SN$_i \Leftarrow 0$;
end for
while $\exists i$ such that $|$TX_BUFF$_i| \neq 0$ do
    for $i = 0$ to $K-1$ do {For all beams}
        Switch its beam to $i$
        Compute $T_S(i)$ and $T_E(i)$
        for $j = 0$ to $\Omega_i - 1$ do {Retransmissions}
            Insert $T_S(i)$ and $T_E(i)$ in the frame at TX_BUFF$_i[j]$
            Transmit the frame at TX_BUFF$_i[j]$
        end for
        for $j = \Omega_i$ to $|$TX_BUFF$_i| - 1$ do {New transmissions}
            Place SN$_i$ to the frame at TX_BUFF$_i[j]$
            Insert $T_S(i)$ and $T_E(i)$ in the frame at TX_BUFF$_i[j]$
            Transmit the frame at TX_BUFF$_i[j]$
            SN$_i \Leftarrow ($SN$_i + 1)\%$SN$_{\max}$;
        end for
        Switch $i$ to $(i-1)\%K$ {To receive bitmap at the previous beam}
        while $T_{ack}$ do
            if received ACK then
                Remove the acknowledged frames from TX_BUFF$_{(i-1)\%K}$
            end if
        end while
    end for
end while
```

Fig. 4. The SPARM algorithm at the sender.

Proof: We represent $\overrightarrow{r_i^a r_i^b}$ as a linear function $y = \theta_i^a x$ since $\theta_i^a = \theta_i^b$. If $\overrightarrow{r_i^a r_i^b}$ passes the origin, it does interfere with the sender. Hence, $\overrightarrow{r_i^a r_i^b}$ cannot pass the origin. Therefore, l_i^a should be less than l_i^b. ∎

B. Sender and Receiver Behavior

Fig. 4 shows the pseudo code of the sender's behavior. SN$_i$ is the last sequence number of outstanding frames at beam i, and Ω_i is the number of unacknowledged frames at TX_BUFF$_i$, where TX_BUFF$_i$ is the transmit buffer array for beam i. When a sender is turned on, the sender collects the receivers' information (i.e., DoA value, received power, etc.) from the hello frames. Then, the sender performs ACK aggregation scheduling for each beam by the SPARM policies and sets each beam's SN$_i$ to 0. After the ACK aggregation scheduling, the sender is ready to multicast its data frame.

Before transmitting multicast frames, the sender computes $T_S(i)$ and $T_E(i)$ where $T_S(i)$ and $T_E(i)$ are the start time of the ACK aggregation process for the first receiver in $\mathbf{R}_i$ and the report time that the last receiver in $\mathbf{R}_i$ transmits its bitmap frame to the sender, respectively, as in Fig. 2. $T_S(i)$ and $T_E(i)$ are given by $T_S(i) = T_C + |$TX_BUFF$_i| \times T_{data} + T_{ack}$ and $T_E(i) = |$TX_BUFF$_{(i+1)\%K}|) \times T_{data} + T_S(i)$, respectively, where T_{data} is the frame transmission time; T_{ack} is the ACK transmission time; T_C is the current system time. Of course, besides $T_S(i)$ and $T_E(i)$, multicast frames contain the order of ACK aggregation (*ordered list of receivers*) for $\mathbf{R}_i$.

If there are any unacknowledged outstanding frames ($\Omega_i > 0$), the sender multicasts those frames with $T_S(i)$ and $T_E(i)$. Also, if there are new frames from the upper layer ($|$TX_BUFF$_i| > \Omega_i$), the sender multicasts those frames with SN$_i$ as well as $T_S(i)$ and $T_E(i)$, and increments SN$_i$. Once the sender finishes multicasting to $\mathbf{R}_i$, the ACK aggregation at $\mathbf{R}_{(i-1)\%K}$ can be completed if no errors occurred in aggregation. Therefore, the sender switches to beam $(i-1)\%K$ and receives a bitmap frame from the last receiver in $\mathbf{R}_{(i-1)\%K}$ during T_{ack}. If the bitmap indicates a frame is successfully received, then the sender removes the corresponding frame from the TX_BUFF$_{i-1}$.

The receiver operation is as follows. When a receiver is turned on, the receiver associates with the sender by a registration frame. When the receiver receives a multicast frame successfully, the receiver marks the SN$_i$th bitmap to 1 as in Fig. 3. Each receiver is informed of its order of aggregation

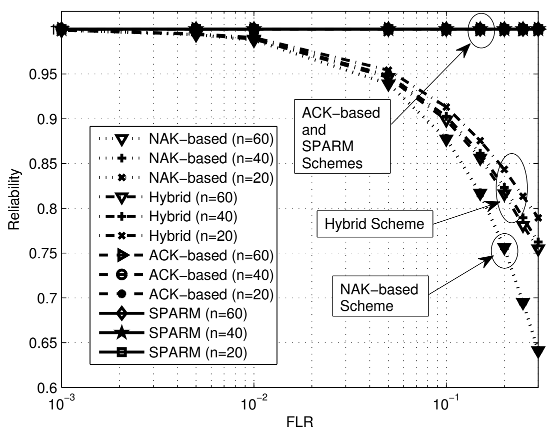


Fig. 5. Reliability vs. FLR.

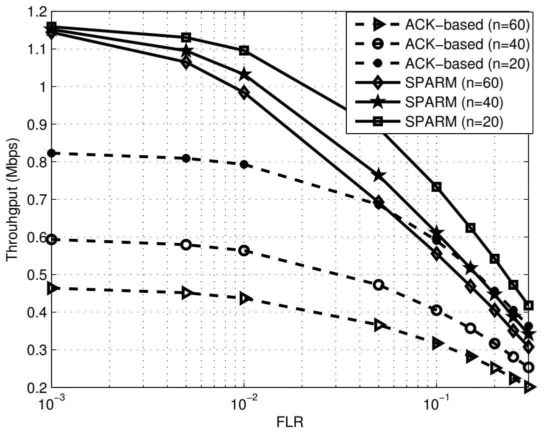


Fig. 6. Throughput vs. FLR

by the multicast frame. If the receiver is the first receiver in the ACK aggregation schedule, the receiver relays its bitmap to the next receiver in the ACK aggregation schedule at $T_S(i)$ (e.g., step 1 in Fig. 3). If the receiver is the last receiver in the ACK aggregation schedule, the receiver aggregates its bitmap with a relayed bitmap and transmits the aggregated bitmap to the sender at $T_E(i)$ (e.g., step 3 in Fig. 3). Otherwise, the receiver aggregates its bitmap with a relayed bitmap and relays the bitmap to the next receiver in the ACK aggregation schedule (e.g., step 3 in Fig. 3).

III. Performance Evaluation

In this section, we evaluate the performance of the proposed scheme compared with the ACK-based [8], NAK-based [3], and hybrid [4] schemes. We modify the above original schemes to support sequential multicasting. For performance evaluation, we use the OPNET modeler and randomly generate 20 to 60 nodes in a 200m $\times$ 200m square area. The simulation parameters are chosen to be similar to [1]. The data rate is assumed to be 10 Mbps. The frame sizes for the data frame and ACK frame are set to be 1024 and 16 bytes (including a 2-byte bitmap), respectively. The average interarrival time of the data frames is assumed to be 0.1 sec and K is assumed to be 4. By varying the frame loss rates, we measure the reliability and throughput defined below.

- **Reliability:** The number of successful frames divided by the number of frames transmitted excluding retransmissions
- **Throughput:** The average number of frames (in bits) per second that are successfully transmitted (i.e., all receivers are acknowledged).

Fig. 5 shows the reliability versus the frame loss rate (FLR) where the FLR varies from 0 to 0.3. As shown in the figure, all schemes except the NAK-based and hybrid schemes guarantee full reliability (i.e., all receivers successfully received the transmitted frames). However, the reliability of the NAK-based and hybrid schemes sharply decreases as the FLR increases. In the NAK-based scheme, this situation occurs when the receivers do not transmit NAK since the receivers are not aware of the loss of the frames. Unreliability also occurs in the hybrid scheme if the leader receives a frame successfully while some frames sent to the other receivers are lost. Further degradation is observed as n increases because the number of receivers that the leader receiver takes charge of is proportional to n. For these reasons, we exclude both NAK-based and hybrid schemes in the remaining performance evaluation.

Fig. 6 shows the throughput when the FLR varies from 0 to 0.3. As shown in the figure, the throughput decreases as the FLR increases for both the ACK-based scheme and SPARM. The throughput decreases more sharply in SPARM than in the ACK-based scheme because the ACK collection latency increases more sharply in the SPARM than in the ACK-based scheme. However, the ACK collection latency of the SPARM cannot be longer than that of the ACK-based scheme since ACKs are aggregated while source multicasting in the SPARM scheme. As a result, the throughput of the SPARM is 200% higher than that of the ACK-based scheme. Also, our scheme is less sensitive to n than the ACK-based scheme. This is because the sender in the SPARM receives only one bitmap frame whereas multiple ACKs are received in the ACK-based scheme.

IV. Conclusion

In this paper, we proposed a SPARM protocol, where receivers sequentially aggregate their ACKs while the sender multicasts a data frame to the next beam, thereby resolving the ACK implosion problem and improving the throughput.

The performance evaluation shows that the proposed SPARM has full reliability and outperforms the existing schemes with respect to the throughput by about 200%. This gain, however, has to be paid for with the complexity associated with the required DoA estimation mechanism. Nevertheless, cost effective pencil beam antenna technology is emerging [7].

References

[1] O. Bazan and M. Jaseemuddin, “Routing and Admission Control for Wireless Mesh Networks with Directional Antennas,” *in Proc. of IEEE WCNC,* April 2009.

[2] K. Chin, “SpotMAC: A Pencil-Beam MAC for Wireless Mesh Networks,” *in Proc. of IEEE ICCCN,* August 2007.

[3] M. Huang, G. Feng, and Y. Zhang, “Cooperative Loss Recovery for Reliable Multicast in Ad Hoc Networks,” *Int. Journal of Comm., Network, and Sys. Science,* vol. 3, no. 1, January 2010.

[4] W. Lim, D. Kim, and Y. Suh, “Design of Efficient Multicast Protocol for IEEE 802.11n WLANs and Cross-Layer Optimization for Scalable Video Streaming,” *IEEE Trans. on Mobile Computing,* vol. 11, no. 5, May 2012.

[5] M. Lin and L. Yang, “Blind Calibration and DOA Estimation With Uniform Circular Arrays in the Presence of Mutual Coupling,” *IEEE Antennas and Wireless Propagation Letters*, Vol. 5, No. 1, December 2006.

[6] S. Palazzo, F. Cuomo, and L. Galluccio, “Data Aggregation in Wireless Sensor Networks: A Multifaceted Perspective,” *Sensor Networks Where Theory Meets Practice*, 2009.

[7] A. Rennings, B. Zhou, A. Al-Bassam, Z. Chen, S. Otto, K. Solbach, and D. Erni, “MetaBeam: A High-Gain Pencil-Beam Array Antenna based on corporately fed CRLH Meta-Lines for the 24 GHz ISM Band,” *in Proc. of GeMiC*, March 2012.

[8] V. Srinivas and L. Ruan, “An Efficient Reliable Multicast Protocol for 802.11-based Wireless LANs,” *in Proc. of IEEE WoWMoM,* June 2009.