BTRB: Beam Table-based Reliable Broadcast for Directional Antennas

Laihyuk PARK†**, Jeongseok YU**†**, Chan-Gun LEE**†**, *and* Sungrae CHO**†**, *Members***

SUMMARY Directional antennas provide numerous benefits, such as higher gains, increased transmission range, and lower interferences. In this paper, we propose a reliable broadcast protocol for directional antenna referred to as beam table-based reliable broadcast for directional antennas (BTRB). The BTRB employs (1) ACK-based scheme to provide full reliability; (2) spatio-temporal ACK combination to resolve the problems of ACK implosion and transmission delay; and (3) beam table caching to represent spatial relationship among destination nodes in the broadcast group. Performance evaluation has shown that the proposed BTRB shows full reliability and outperforms existing reliable broadcast schemes with respect to transmission delay by about 55%.
key words: *Directional MAC, reliability, broadcast, ACK combination.*

1. Introduction

Directional antenna is a promising technology for high data rate service such as IEEE 802.11ad to support many applications including VoD streaming, wireless display, and group communications such as conference rooms, lecture halls, etc. For these applications, reliable broadcasting is an essential technique to be developed. While a significant number of studies have been proposed for reliable broadcasting in the omni-directional domain [3–5], the directional counterpart has been considered as a relatively undeveloped area. Multihop broadcasting approaches have been proposed in [2, 6] for directional antennas. However, they did not provide the reliability for their broadcasting service. Few reliable broadcast techniques for directional antennas have been studied in [7,8]; however, they have been limited to apply the existing omni-directional protocols or methods (i.e., BMMM [8] and ARB [7]) to directional environment.

Reliable broadcast technique has been traditionally categorized as ACK, NAK, or both. In the ACK-based approach [5, 8], a source node needs a positive acknowledgment (ACK) from all destinations to guarantee full reliability. To avoid ACK implosion problem, a source node schedules destinations' ACK transmission [5] or explicitly polls each destination [8]. In the ACK-based scheme, however, the ACK collection delay will be significant and thus the throughput might be degraded because all destinations need to send ACK. Similar to the ACK-based approach, the NAK-based scheme [3] schedules NAK by random timers to avoid NAK implosion. However, NAKs cannot handle the unique cases of all frames being lost at a particular destination. Since the destination is not aware that a data frame is expected, it cannot possibly advertise a NAK to request retransmission. To tackle the above problems, hybrid schemes [4, 7] (using both ACK and NAK) are proposed. In [4], non-leader destinations use the NAK-based scheme while the leader uses ACK-based scheme. In [7], erroneous destinations use NAKs while successful destinations use ARB. However, these schemes do not correct the problem where the leader (or destination) receives a frame successfully while that frame to the other destinations is lost.

As mentioned in the above, NAK or hybrid approaches do not guarantee full reliability. Therefore, we propose an ACK-based approach referred to as beam table-based reliable broadcast for directional antennas (BTRB). Aforementioned, ACK-based schemes suffer from ACK implosion problem and transmission delay. By employing beam table caching and spatio-temporal ACK combination, our BTRB resolves the above problems. The rest of this paper is organized as follows. The BTRB is described in detail in section 2 followed by simulation results in section 3. Finally, conclusions are drawn in section 4.

2. BTRB Scheme

We consider a single hop environment, where a source (denoted by s) broadcasts a frames to M destinations (d_i, $0 \leq i < M$) using its N disjoint beams (for 360° coverage). We denote $\mathbf{D}_n$ as the group of the destinations which are located in the source's nth beam ($n \in \{0, 1, \cdots, N-1\}$). For example, $\mathbf{D}_0 = \{d_2, d_4, d_5, d_{12}\}$ as shown in Fig. 1. We assume that the broadcasting is scheduled sequentially to cover all destinations from 0th to $(N-1)$th beam, one beam to another. For directional communication, the other beams are blocked when one beam is sending or receiving. Therefore, we do not consider simultaneous broadcasting with all N beams. Also, let $B_{i,j}$ denote the beam number of node i towards node j, e.g., $B_{d_2,d_5} = 3$ since d_2 uses its beam 3 towards d_5 as shown in Fig. 1. Here, nodes i and j can be broadcast source or destinations. If node i cannot send to node j since there is no link or no information about j, $B_{i,j}$ is set to -1, e.g., $B_{d_2,d_7} = -1$ since d_2 has no information about d_7 as shown in Fig. 1.

Initially, a node is not aware of the other nodes' beam information, i.e., node i does not have any infor-

[a]The authors are with School of Computer Science and Engineering, Chung-Ang University, 221 Heukseok, Dongjak, Seoul 156-756 South Korea; Email: srcho@cau.ac.kr; Tel: +82-2-820-5766; Fax: +82-2-824-1394.

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

DOI: 10.1587/trans.E0.??.1

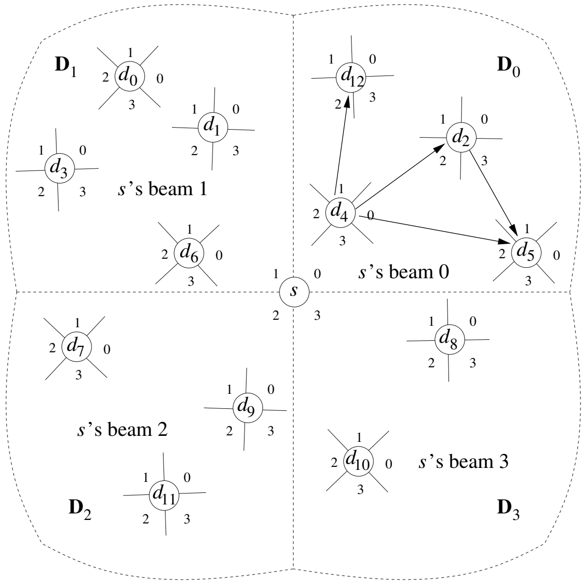


Fig. 1 An exemplary network topology (N=4 and M=13).

Table 1 The source's beam table of Fig. 1.

sender	receiver													
	s	d_0	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}
s	-1	1	1	0	1	0	0	1	2	3	2	3	2	0
d_0	3	-1	0	0	3	0	-1	3	3	-1	3	3	3	0
d_1	3	-1	-1	3	2	3	3	-1	2	3	2	-1	2	3
d_2	2	1	-1	-1	2	2	3	2	-1	2	2	2	2	1
d_3	3	0	0	0	-1	3	3	-1	3	3	3	-1	3	0
d_4	3	2	2	0	2	-1	0	2	2	-1	-1	3	3	1
d_5	2	2	2	-1	2	2	-1	2	2	2	2	2	2	2
d_6	0	1	1	0	2	-1	0	-1	2	-1	3	3	3	0
d_7	0	1	1	-1	1	0	-1	0	-1	0	0	0	-1	0
d_8	1	1	1	0	1	1	0	1	1	-1	2	-1	2	1
d_9	0	-1	0	0	1	-1	0	1	1	0	-1	3	-1	0
d_{10}	1	1	1	1	2	1	0	1	2	-1	2	-1	2	1
d_{11}	0	1	-1	0	1	0	-1	0	1	0	-1	0	-1	0
d_{12}	2	1	2	-1	2	2	-1	2	2	-1	2	-1	2	-1

For instance, the value (at sender=d_2 and receiver=d_5) is 3 since d_2 uses its beam 3 towards d_5; while the value (at sender=d_2 and receiver=d_7) is -1 since d_2 has no information about d_7 as shown in Fig. 1.

mation about $B_{j,i}$ ($\forall j \neq i$ and nodes i and j are in communication range). However, as network is sufficiently established, we assume each node can have the beam information of other nodes by exchanging or overhearing Hello or data frames among them. Node i sending these frames should include (1) list of $B_{i,j}$'s and (2) list of $B_{j,i}$'s in the frames where nodes i and j are in communication range. For instance, d_2 needs to inform s the list of $B_{d_2,j}$'s (the shaded row in Table 1) and the list of B_{j,d_2}'s (the shaded column in Table 1).

Based on this beam information, a node can form a beam table as shown in Table 1. We refer this mechanism as a *beam table caching*. Let $\mathbf{S}$ and $\mathbf{R}$ denote sets of senders and receivers, respectively inside the corresponding node's communication range. The beam table is described by matrix $\mathbb{B}_{\mathbf{S}}^{\mathbf{R}} = (B_{x,y}, x \in \mathbf{S}, y \in \mathbf{R})$.

2.1 ACK Combination

In the BTRB scheme, s broadcasts data frame in circular fashion from $\mathbf{D}_n$ to $\mathbf{D}_{(n+1)\%N}$. For instance, suppose that s finishes broadcasting to $\mathbf{D}_0$ as in Fig. 1. Then, s switches its beam to 1 and broadcasts its data frame to $\mathbf{D}_1$. At this instant (when s is broadcasting frame at beam 1), it will be efficient that the destinations in $\mathbf{D}_0$ combine[a] ACKs of the previous transmission, in order to reduce transmission delay. In our scheme, ACK combination is defined by bit-wise AND operation of bitmaps of destinations in $\mathbf{D}_n$. The bitmap is stored at each destination, and used to indicate which frames[b] are successfully received. If the kth bit of a bitmap is set 1, this implies the k'th frame is successfully received, otherwise, '0' is used. To share the bitmaps among destinations, bitmap frames are transmitted where the bitmap frame consists of a bitmap header and a bitmap. The bitmap header contains the first sequence number (SN) of the bitmap and the number of outstanding frames. By the result of bit-wise AND operation of destinations' bitmaps, we can observe which frames are successfully received at all destinations in $\mathbf{D}_n$.

In the BTRB scheme, ACK (bitmap) combination of $\mathbf{D}_i$ should not interfere to s's broadcasting to $\mathbf{D}_{(i+1)\%N}$. Suppose d_2 sends its bitmap to d_4 for ACK combination while s is broadcasting at $\mathbf{D}_1$ as in Fig. 1, d_2 will cause interference to s and $\mathbf{D}_1$. We can prevent this situation based on beam table as in Table. 1. Thus, the order of ACK combination requires to be carefully designed to avoid interference. If we schedule the ACK combination as follows: d_4 to d_2, d_2 to d_5, d_5 to s, and d_{12} to s, there will be no interference caused to s and $\mathbf{D}_1$. Of course, ACK combination from d_5 to s has to be delayed until broadcasting to $\mathbf{D}_1$ finishes. This exemplary order of ACK combination will be explained in the rest of this section.

In the BTRB, this scheduling is determined by source node. To make the order of ACK combination in $\mathbf{D}_n$, s needs to consider the following steps.

Step 1: From the source's beam table, the source node extracts $\mathbb{B}_{\mathbf{D}_n}^{\mathbf{D}_n}$ and $\mathbb{B}_{\mathbf{D}_n}^{(\mathbf{D}_{((n+1)\%N)} \cup s)}$. Here, $\mathbb{B}_{\mathbf{D}_n}^{\mathbf{D}_n}$ is the beam information for communication among $\mathbf{D}_n$ while $\mathbb{B}_{\mathbf{D}_n}^{(\mathbf{D}_{((n+1)\%N)} \cup s)}$ is the beam information of $\mathbf{D}_n$ that should not be used for ACK combination since using these beams can interfere to s's broadcasting. Table 2 (a) and (b) show $\mathbb{B}_{\mathbf{D}_0}^{\mathbf{D}_0}$ and $\mathbb{B}_{\mathbf{D}_0}^{(\mathbf{D}_1 \cup s)}$ for the scenario in Table 1, respectively.

Step 2: The source computes a *candidate link matrix* ($\mathbb{A}_n$) which indicates the beam information of $\mathbf{D}_n$ that can be used to combine ACKs during s broadcasts to $\mathbf{D}_{(n+1)\%N}$. $\mathbb{A}_n$ is defined by

$$\mathbb{A}_n = \mathbb{B}_{\mathbf{D}_n}^{\mathbf{D}_n} \ominus \mathbb{B}_{\mathbf{D}_n}^{(\mathbf{D}_{((n+1)\%N)} \cup s)} \quad (1)$$

[a] Instead of sending ACK directly to the source, destination nodes cooperatively combine their ACKs sequentially in the BTRB scheme.

[b] Considering the transmission window, there might be outstanding frames that are not acknowledged.

Table 2 The exemplary tables for BTRB as scenario in Table 1.

Sender	(a) $\mathbb{B}_{\mathbf{D}_0}^{\mathbf{D}_0}$.				(b) $\mathbb{B}_{\mathbf{D}_0}^{(\mathbf{D}_1 \bigcup s)}$.					(c) $\mathbb{A}_0$.			
	Receiver				Receiver					Receiver			
	d_2	d_4	d_5	d_{12}	s	d_0	d_1	d_3	d_6	d_2	d_4	d_5	d_{12}
d_2	-1	2	3	1	2	1	-1	2	2	-1	-1	3	-1
d_4	0	-1	0	1	3	2	2	2	2	0	-1	0	1
d_5	-1	2	-1	2	2	2	2	2	2	-1	-1	-1	-1
d_{12}	-1	2	-1	-1	2	1	2	2	2	-1	-1	-1	-1

which is the process where each sender eliminates beam numbers in $\mathbb{B}_{\mathbf{D}_n}^{\mathbf{D}_n}$ that match any beam numbers in $\mathbb{B}_{\mathbf{D}_n}^{(\mathbf{D}_{((n+1)\%N)} \bigcup s)}$. Here, the elimination result should be marked as '-1'. As an example, Table 2 (a) has {-1, 2, 3, 1} for sender d_2. Now, since Table 2 (b) has {2, 1, -1, 2, 2} for d_2, {-1, 2, 3, 1} has to be changed to {-1, -1, 3, -1} as in Table 2 (c). From $\mathbb{A}_0$, we can obtain an ACK combination topology as in Fig. 1 (see the arrows such that connect from d_2 to d_5 and from d_4 to d_2, d_5, and d_{12}).

Step 3: The source node then determines the order of ACK combination as follows. We denote $\mathbf{C}_n^{l_n}$ is a path connecting one destination to another in $\mathbf{D}_n$ where $l_n \in \{0, 1, \cdots, L-1\}$ and L is the number of paths from $\mathbb{A}_n$. The source chooses the ACK combination order set ($\mathbf{C}_n$) given by

$$\mathbf{C}_n = \underset{\forall l_n}{\arg\max} |\mathbf{C}_n^{l_n}| \tag{2}$$

where $|f|$ is the length of set f. For instance, $\mathbf{C}_0 = \{d_4, d_2, d_5\}$ for scenario in Table 1.

Step 4: Depending upon the network conditions, some of the destination's Hello messages might be lost for significant duration. In that case, beam table information will not be available. Those destinations have no way to combine ACKs. Therefore, those nodes have to unicast ACK to the source in our scheme. We denote $\mathbf{U}_n$ is the set of unicast destinations in $\mathbf{D}_n$ and computed by $\mathbf{U}_n = \mathbf{D}_n - \mathbf{C}_n$. Of course, the order of unicast from the destinations in $\mathbf{U}_n$ has to be scheduled in accordance of increasing order of node ID.

2.2 BTRB Protocol

We denote SN_n is the last sequence number of outstanding frames at beam n and $\mathbf{Q}_n$ is the transmission queue for beam n. Suppose that the source already determined ACK combination order as in 2.1. Then, the source is ready for broadcasting its data frame.

The source needs to inform the destinations in $\mathbf{D}_n$ of when the ACK combination starts and ends. Let Δ_n^S and Δ_n^E denote the start time of ACK combination process for the first destination in $\mathbf{C}_n$ and the report time that the last destination in $\mathbf{C}_n$ sends its bitmap frame to the source, respectively. Let Δ^C denote the current system time before the source sends its broadcast frames to $\mathbf{D}_n$, Δ^{data} denote the frame transmission time, and Δ^{ack} denote the ACK transmission time. After the source broadcasts to $\mathbf{D}_n$ ($|\mathbf{Q}_n| \cdot \Delta^{data}$), the source waits for the ACKs from $\mathbf{D}_{(n-1)\%N}$ during $(|\mathbf{U}_{(n-1)\%N}| + 1) \cdot \Delta^{ack}$. Because ACK combination in $\mathbf{D}_n$ should be processed while the source broadcasts at beam $(n+1)\%N$, Δ_n^S should be the instant when the source starts broadcasting to $\mathbf{D}_{(n+1)\%N}$. Therefore, Δ_n^S is given by

$$\Delta_n^S = \Delta^C + |\mathbf{Q}_n|\Delta^{data} + (|\mathbf{U}_{(n-1)\%N}| + 1)\Delta^{ack} \tag{3}$$

Similarly, Δ_n^E is given by

$$\Delta_n^E = |\mathbf{Q}_{(n+1)\%N}|\Delta^{data} + \Delta_n^S. \tag{4}$$

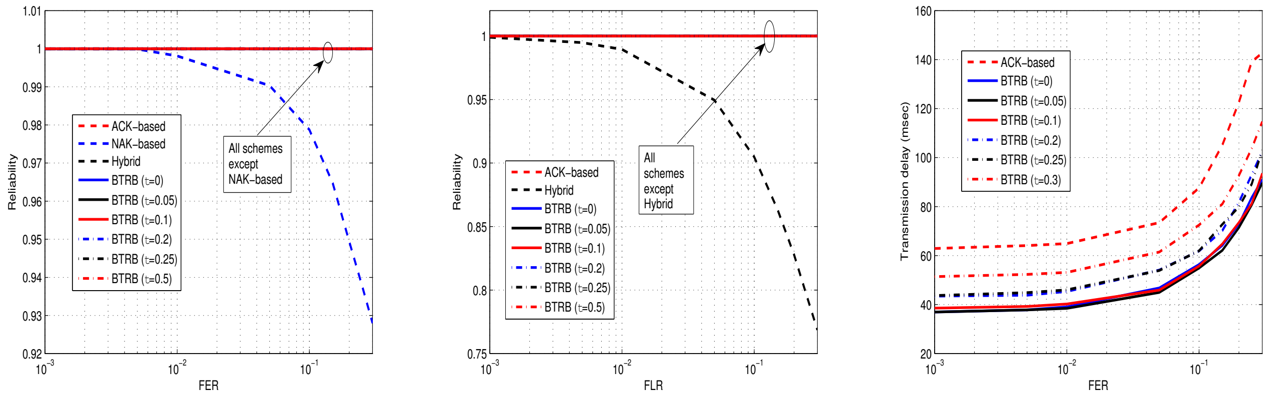
In our scheme, the broadcast frame to $\mathbf{D}_n$ includes SN_n, Δ_n^S, Δ_n^E, $\mathbf{C}_n$, and $\mathbf{U}_n$ so that the destinations in $\mathbf{D}_n$ can process ACK combination.

Now, the source behavior is as follows. The source starts from $n = 0$. If there are any unacknowledged outstanding frames ($|\mathbf{Q}_n| > 0$), the source broadcasts those frames. Once the source finishes broadcasting to $\mathbf{D}_n$, the ACK combination at $\mathbf{D}_{(n-1)\%N}$ might be completed if the source sent broadcast frames previously. Therefore, source switches to beam $(n-1)\%N$ and receives a bitmap frame from the last destination in $\mathbf{C}_{(n-1)\%N}$ during Δ^{ack}. Then, the source waits for unicast ACKs during $(|\mathbf{U}_{(n-1)\%N}|) \cdot \Delta^{ack}$. If the bitmap and unicast ACKs indicate broadcasted frames are successfully received, the source removes the corresponding frames from the $\mathbf{Q}_{n-1}$. Then, the source switches to beam $(n+1)\%N$.

The destination operation is as follows. When the destination receives a broadcast frame successfully, its marks the corresponding bit of bitmap to '1.' Each destination is informed of its order of unicast or combination by the broadcast frame. If its order is xth destination in the unicast group, it unicasts its ACK to the source at $\Delta_n^E + x \cdot \Delta^{ack}$. If it is the first destination in the ACK combination group, it relays its bitmap to the next destination in the ACK combination group at Δ_n^S. If it is the last destination in the ACK combination group, it combines its bitmap with a relayed bitmap and sends the combined bitmap to the source at Δ_n^E. Otherwise, it combines its bitmap with a relayed bitmap and relays the bitmap to the next destination in the ACK combination group.

3. Performance Evaluation

In this section, we evaluate performance of the proposed scheme compared with the ACK-based [5], NAK-based [3], and hybrid [4] schemes. We modify the above original schemes to support sequential broadcasting. For performance evaluation, we use the OPNET modeler with random topology. The simulation parameter are chosen to be similar to [1]. We assume the number of the source and destinations are 1 and 60, respectively. The data rate is assumed to be 10 Mbps. The frame sizes for data frame and ACK frame are set to be 1024 and 2 bytes, respectively. The average interarrival time of data frames is assumed to be 0.1 sec and N is assumed to be 4. In our simulation scenarios, we assume some destinations can lose their beam tables for realistic situations and the percent of these destinations is denoted by τ. With this setup, we measure the reliability and the transmission delay of all schemes



(a) reliability vs. FER. (b) reliability vs. FLR. (c) The transmission delay(msec) vs. FER.

Fig. 2 Performance Evaluation.

defined by

- **Reliability:** the number of successful frames divided by the number of frames transmitted excluding retransmissions.
- **Transmission delay:** the average delay from the broadcast data to the corresponding ACK received.

Fig. 2(a) shows the reliability versus the frame error rate (FER) where the FER varies from 0 to 0.3. As shown in the figure, all schemes except the NAK-based scheme guarantee full reliability (i.e., no errors are generated). However, the reliability of the NAK-based scheme sharply decreases as the FER increases. This situation occurs since the NAK can be corrupted and the sender may not receive any NAK even if there are errors at receivers. It can be expected to have more degradation with more receivers or higher FER. Thus, we exclude the NAK-based scheme in the remaining performance evaluation since this scheme does not guarantee full reliability.

Since unreliability may arise not only because of frame errors, but also frame losses, we evaluate the impact of frame loss. Fig. 2(b) shows the reliability versus the frame loss rate (FLR) where the FLR varies from 0 to 0.3. As shown in the figure, only the hybrid scheme does not provide full reliability. Unreliability can occur for the hybrid scheme if a broadcast frame is received at the leader receiver successfully, but it is lost at some of the non-leader receivers; and thus they do not transmit NAK. With the same reason as the above, we exclude the hybrid scheme in the remaining performance evaluation.

Fig. 2(c) shows the transmission delay when the FER varies from 0 to 0.3. As shown in the figure, the ACK transmission delay increases as the FER increases for both ACK-based and our schemes, and the delay of our scheme ($\tau = 0$) is less than the ACK-based scheme by 55% at FER=0.3. Delay of our scheme increases as τ increases. It is because the number of unicast destinations increases. Although $\tau = 0.3$, the delay of our scheme is still less than ACK-based scheme by 21%. As the worst case, we can expect the delay performance of our scheme is the same as the ACK-based scheme with $\tau = 1$.

4. Conclusion

In this paper, we propose the BTRB protocol to solve the problems of ACK implosion and transmission delay by ACK combination. To avoid interference of the ACK combination process to the source's broadcasting, the ACK combination is scheduled based on beam table caching. Performance evaluation has shown that the proposed BTRB shows full reliability and outperforms existing reliable broadcast schemes with respect to transmission delay by about 55%.

References

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

[2] F. Dai *et al.*, "Efficient broadcasting in ad hoc wireless networks using directional antennas," *IEEE Trans. Parallel and Distributed Systems,* vol.17, no.4, pp.335-347, Apr. 2006.

[3] M. Huang *et al.*, "Cooperative Loss Recovery for Reliable Multicast in Ad Hoc Networks," *Intl. J. of Communications, Network, and System Science,* pp. 72–78, 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 TMC,* Vol. 27, No. 8, October 2009.

[5] V. Srinivas and L. Ruan, "An Efficient Reliable Multicast Protocol for 802.11-based Wireless LANs," *in Proc. of IEEE WOWMOM,* June 2009.

[6] C. C. Shen *et al.*, "Directional broadcast for mobile ad hoc networks with percolation theory," *IEEE Trans. Mobile Computing,* Vol.5, No.4, pp.317–332, April 2006.

[7] Y. Utsunomiya *et al.*, "A medium access control protocol with retransmission using NACK and directional antennas for broadcasting in wireless ad-hoc networks," *in Proc. European Wireless Conference,* 2004.

[8] R. M. Yadumurthy *et al.*, "Reliable MAC broadcast protocol in directional and omni-directional transmissions for vehicular ad hoc networks," *in Proc. of ACM international workshop on Vehicular ad hoc networks,* pp.10–19, 2005.