A Reliable Broadcast Algorithm with Selected Acknowledgements in Mobile Ad Hoc Networks

Wei Lou and Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
Email:{wlou, jie}@cse.fau.edu

Abstract— **Mobile ad hoc networks (MANETs) suffer from transmission contention and congestion because of the broadcast nature of radio transmission. The broadcast operation, as a fundamental service in MANETs, will cause the broadcast storm problem if the forward nodes are not carefully managed. It is a major challenge to reduce broadcast redundancy while still providing high delivery ratio for each broadcast packet in a dynamic environment. In this paper, we propose a simple broadcast algorithm to provide high delivery ratio. Among the 1-hop neighbors of the sender, only selected forward nodes will send acknowledgements to confirm their receipt of the packet. Forward nodes are selected in such a way that all the sender's 2-hop neighbors are covered. Moreover, no acknowledgment is needed from non-forward 1-hop neighbors, each of which is covered by at least two forward neighbors. The sender waits for the acknowledgements from all of its forward nodes. If not all acknowledgments are received, the sender will resend the packet until the maximum number of retries is reached. Simulation results show that the algorithm has high delivery ratio and low end-to-end delay for a broadcast operation.**

I. INTRODUCTION

A mobile ad hoc network (MANET) consists of a group of mobile nodes, and it enables communications between participating nodes without the burden of any base stations. When two nodes that are out of one another's transmission range intend to communicate with each other, they need the support of the intermediate nodes for relaying the messages. Broadcast operation has the most significant role in MANETs because of the broadcasting nature of the radio transmission, that is, when a sender transmits a packet, all nodes in the sender's transmission range will be affected by this transmission. The advantage of this nature is that one packet can be received by all neighbors; the disadvantage is that it will interfere with the sending and receiving of other transmissions, creating issues referred to as *exposed terminal* and *hidden terminal problems*.

Blind flooding, where each node forwards once and only once, makes every node a forward node. If the forward nodes are not carefully managed, they will cause the *broadcast storm problem* [9]. To reduce contention, a subset of nodes is used to forward the broadcast message and the remaining nodes are still covered (i.e., they are adjacent to forward nodes). Basically, forward nodes form a *connected dominating set* (CDS). A *dominating set* (DS) is a subset of nodes such that every node in the graph is either in the set or is adjacent to a node in the set. If the subgraph induced from a DS of the network is connected, the DS is a CDS. Finding a *minimum CDS* in a given graph is NP-complete; in a unit disk graph, it has also been proved to be NP-complete [7].

Since MANETs suffer from transmission contention and congestion that are results of the broadcasting nature of radio transmission, it is a major challenge to provide a reliable broadcasting under such dynamic MANETs. We aim to reduce broadcast redundancy by decreasing the number of the forward nodes yet still provide high delivery ratio for each broadcast packet in a dynamic environment. As pointed out by other researchers, providing total reliability for broadcasting in a dynamic MANET is impractical and unnecessary when the physical communication channels are prone to errors. Usually, acknowledgments (ACKs) are used to ensure broadcast delivery. However, the requirement of sending ACKs in response to the receipt of a packet for all receivers may cause channel congestion and packet collision, which is called *ACK implosion* [3]. Our goal is to make a sensible reduction in ACKs without sacrificing broadcast delivery ratio. Specifically, we propose a simple reliable broadcast algorithm that only requires selected forward nodes among the 1-hop neighbors to send ACKs to confirm their receipt of the packet. Forward nodes are selected in such a way that all sender's 2-hop neighbor nodes are covered. Moreover, no ACK is needed for non-forward 1-hop neighbors, each of which is covered by at least two forward neighbors, one by the sender itself and one by one of the selected forward nodes. The sender waits for the ACKs from all of its forward nodes. If not all ACKs are received, it will resend the packet until the maximum times of retry is reached. If the sender fails to receive all ACKs from the forward nodes, it assumes that the non-replied forward nodes are out of its range and chooses other nodes to take their roles as forward nodes. Simulation results show that the algorithm has high delivery ratio and low end-to-end delay for a broadcast operation.

II. RELATED WORK

Our work is related to the following two aspects:

A. Neighbor-Designating-Based Broadcasting

In [14], Wu and Dai proposed a generic distributed broadcast scheme in which a CDS is constructed for a particular broadcast and dependent on the location of the source and the

progress of the broadcast process. Each node v determines its status and the status of some of its neighbors under a current local view. Two categories of broadcasting approaches, called *self-pruning* and *neighbor-designating* broadcasting approaches, are classified. We are interested in the class of neighbor-designating approach, where a node can determine its neighbor's forwarding/non-forwarding status. In [12], Qayyum et al proposed selected multipoint relays (MPRs) as forward nodes. The MPRs are selected from 1-hop neighbors to cover the entire set of 2-hop neighbors. A relaxed neighbor-designating requirement is applied in [4]: if an MPR first receives a broadcast packet from a neighbor that is not its designator, it does not forward this packet. Lim and Kim [5] provided a dominant pruning algorithm (DP) that excludes the coverage of the forwarded node from the current node's 2-hop neighbor set. Suppose u is the last forwarded node and v is designated as the next forward node, v selects its forward node set from $N(v) - N(u)$ to cover 2-hop neighbor set $N_2(v) - N(u) - N(v)$. [6] and [11] provide further extensions.

B. Reliable Broadcasting

A reliable communication needs some feedback from receivers. The basic schemes for reliable communication can be classified as *sender initiated* and *receiver initiated* approaches [13]. In the sender initiated approach, the receiver returns a positive ACK to the sender for each message it receives. The sender maintains all records for each receiver to confirm the success of the delivery. Only missing packets are retransmitted by the sender, either to individual requested receivers, or to all receivers. The drawback of this scheme is that the sender may become the victim of collisions when simultaneous ACKs return. Moreover, the amount of records that the sender must maintain may also grow large. In the receiver initiated approach, the receiver is responsible for reliable delivery. Each receiver maintains receiving records and requests retransmission via a negative acknowledgement (NACK) when errors occur. The problem of the receiver initiated approach is the long end-to-end delay since the sender cannot terminate a broadcast operation until it receives feedback from receivers. Therefore, it can be applied only when the sender has many packets to be sent.

Most reliable broadcast protocols come from the routing protocol proposed by Merlin and Segall [8]: The source starts a broadcast operation by sending a message to all its neighbors and waiting for the ACKs from its neighbors. When it receives all these ACKs, it sends the message asking the neighbors to propagate the message one more hop to their own neighbors. The neighbors of the source forward the message to their neighbors and send the ACKs back to the source when they receive all ACKs from all their own neighbors, and so forth. The scheme incurs too much communication overhead and needs stable linkages for MANETs.

A flooding-based reliable broadcast protocol proposed by J. J. Garcia and Zhang [2] allows the nodes that received the broadcast packet to forward the packet without further notice from the sender. Alagar and Venkatesan [1] also proposed a

reliable broadcast protocol based on flooding. The protocol works as follows: The source broadcasts the message to its 1-hop neighbors. When a node receives the message, it sends an ACK back to the sender. If the message is a new one, the node retransmits the message; otherwise, it drops the message. If the sender does not receive an ACK from any of its neighbors for a predefined period, it re-sends the message. In the case that some links happen to be broken, a handshake process is provided; this process requires that the two neighbor nodes exchange all of the messages they have thus far so as to keep all records identical. The obvious drawback of these flooding-based protocols is that the flooding may easily introduce the broadcast storm problem. The ACK implosion problem may worsen the broadcast storm problem.

Pagani and Rossi [10] proposed a cluster-based reliable broadcast protocol in MANETs: A forwarding tree, which is rooted from the clusterhead of source to each clusterhead, is constructed to forward the broadcast packet. The packet is forwarded down the tree from the root to the leaf nodes; the ACKs are first collected by each clusterhead in each cluster and then up the tree from the leaves to the root. The algorithm changes to flooding when the topology change of the network becomes high.

All the above reliable broadcast algorithms require each receiver to send ACKs in response to the receipt of a packet. These ACKs may become another bottleneck of channel congestion and lead to ACK implosion problem.

III. A RELIABLE BROADCAST ALGORITHM

A. Basic Idea

A reliable broadcast operation requires the packet be disseminated to all nodes in the network. But the interference of the transmission of neighbors and the movement of the nodes may cause the failure of some nodes to receive the broadcast packet. Therefore, the sender needs to retransmit the packet to increase the delivery ratio of the transmission.

The proposed reliable broadcast algorithm works as follows: When a source broadcasts a packet, it selects a subset of 1-hop neighbors as its forward nodes to forward the broadcast based on a greedy approach. The selected forward nodes must cover all the nodes within 2 hops of the source. After the forward nodes receive the broadcast packet, they need to send back ACKs to confirm their receipt. Each forward node records the packet, computes its forward nodes and re-broadcasts the packet. The sender waits for a predefined duration to receive ACKs from its forward nodes. If the sender does not receive all ACKs from its forward nodes during this duration, it assumes that a transmission failure has happened for this broadcast and that the packet needs to be resent. If the sender fails to receive ACKs from all its selected forward nodes after it sends the packet a threshold number of times, the sender assumes the forward nodes that do not reply are out of its transmission range and stops further attempts.

A node may fail to receive the broadcast packet from its neighbors. If the node that misses the packet is a non-forward node, it still has a chance to receive the broadcast packet since

each node is covered by at least two neighboring forward nodes. Also, the missed packet does not cause other missing propagations in the network. On the other hand, if it is a forward node that misses the broadcast packet, this miss may propagate through the network since the neighbors of this forward node will also miss the packet. We apply the following extension to improve the performance of the algorithm: When a sender u fails to receive an ACK from its forward node v after maximum number of retries, u re-selects alternative forward nodes to cover the set which is supposed to be covered by v.

The algorithm that requires only the selected forward nodes send ACKs, which is commonly used for nodes sending NACKs to inform the sender of the missing packet, can avoid the ACK implosion problem. Also, the algorithm guarantees that each node is covered by at least two transmissions so that a missing packet caused by a single collision can be avoided. Moreover, the algorithm does not suffer the disadvantage of the receiver-initiated approach that needs a much longer time to detect a missed packet.

B. Forward Node Set Selection Process

We describe a MANET as a unit disk graph $G = (V, E)$, where the node set V represents a set of wireless mobile nodes and the edge set E represents a set of bi-directional links between the neighboring nodes. Two nodes are considered neighbors if and only if their geographic distance is less than the transmission range r. We use $N_k(v)$ to represent the neighbor set of v, where nodes in the set are not further than k hops from v. $N_k(v)$ includes v itself. ($N_1(v)$, 1-hop neighbor set, can be simply represented as $N(v)$.) If S is a node set, $N(S)$ is the union of the neighbor sets of every node in S, that is, $N(S) = \cup_{w \in S} N(w)$.

Neighboring nodes exchange their 1-hop neighbor set information, therefore, each node v has its 2-hop neighbor set information $N_2(v)$. The forward node set selection process executes at each forward node to determine its own forward node set: A node u selects its forward node set from its 1-hop neighbor set $N(u)$ to cover all the nodes in its 2-hop neighbor set $N_2(u)$. Therefore, each node v in $N(u)$ can be one of two cases: 1) v is a forward node, it will actively reply an ACK when it receives the broadcast packet. 2) v is not a forward node, it will not actively reply an ACK, but it is adjacent to at least two nodes that will locally broadcast the broadcast packet: one is u and the other is the forward node that covers v. Therefore, v has at least two chances to correctly receive the broadcast packet.

The forward nodes are selected based on the following greedy algorithm:

In the sample network shown in Figure 1, $N(1) = \{1, 2, 3, 4, 6\}$ and $N_2(1) = \{1, 2, 3, 4, 5, 6, 7\}$. When using the FNSSP, sender node 1 selects nodes $2, 3$ and 4 as its forward nodes. Node 3 is selected because there is no node in $N(1)$ to cover it.

Algorithm 1 Forward Node Set Selection Process (FNSSP)

1: The forward node set F is initialized to be empty.
2: Add in F the node that covers the largest number of 2-hop neighbors that are not yet covered by current F. A tie is broken by node ID.
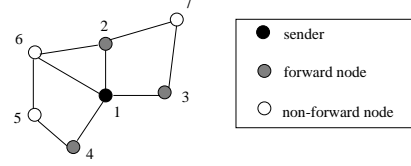3: Repeat step 2 until all 2-hop neighbors are covered.

Fig. 1. A sample network where the sender 1 uses the FNSSP to select its forward nodes.

C. Broadcast With Selected Acknowledgements

The broadcast algorithm is described as a set of event-driven rules. We assume a broadcast process starts from source s. The following symbols are used:

- $F(v)$: the forward node set of node v.
- $U(v)$: the uncovered 2-hop neighbor set of node v.
- Cnt_v: the number of times the packet has been sent by node v.
- T_v: a timer at node v for acknowledgement.
- $P(s, v, F(v))$: a broadcast packet from source s, attaching $F(v)$, and forwarded by node v.
- $ACK(P(s), u, v)$: an ACK sent from v for the broadcast packet $P(s, u, F(u))$.
- $WAIT\_FOR\_ACK$: the bound on the timer for acknowledgment.
- MAX_RETRY: the bound on the number of broadcast retries.

When a broadcast process starts from s, it uses the FNSSP algorithm to select its forward node set $F(s)$, and then piggybacks $F(s)$ with the packet and broadcasts the packet to its 1-hop neighbor set $N(s)$.

For a node v that receives a new broadcast packet from an upstream sender u for the first time, v initializes its uncovered 2-hop neighbors $U(v) = N_2(v)$. If v is a forward node (i.e., v is in $F(u)$), it sends back an ACK to u and also computes its forward nodes to relay the packet. Node v updates $U(v)$ by excluding $N(u)$ and $\cup_{\forall x \in F(u), id(x) < id(v)} N(x)$, that is, $U(v) = U(v) - N(u) - \cup_{\forall x \in F(u) \wedge id(x) < id(v)} N(x)$. If the updated $U(v)$ is empty, then v does not need to forward the packet again. $N(u)$ is excluded from $U(v)$ because u has sent the packet and its neighbors can receive the packet. The reason $\cup_{\forall x \in F(u) \wedge id(x) < id(v)} N(x)$ can be excluded from $U(v)$ is that the nodes in this union are scheduled to be covered by some forward nodes whose ID is less than v without mutual exclusion since nodes in $F(u)$ are totally ordered by node ID.

The sender u broadcasts the packet and waits for a duration $WAIT\_FOR\_ACK$ to receive ACKs from the forward nodes. If u received an ACK from its forward node v, v will be removed from $F(u)$. If u does not receive ACKs from all of its forward nodes during this duration, it assumes the transmission failure has happened for this broadcast packet

Algorithm 2 Broadcast with Selected ACKs (SBA)

For new packet starts from source s
  use the FNSSP algorithm to find $F(s)$ to cover $N_2(s)$.
  $Cnt_s := 0$
  $T_s := WAIT\_FOR\_ACK$
  broadcast$P(s, s, F(s))$ to $N(s)$.

When v receives a packet $P(s, u, F(u))$ from u
  if P is a new one **then**
    $U(v) = N_2(v)$
  end if
  if $v \in F(u)$ **then**
    send $ACK(P(s), u, v)$ to u.
    if P has never been sent **then**
      $U(v) = U(v) - N(u) - \cup_{\forall x \in F(u) \wedge id(x) < id(v)} N(x)$
      if $U(v) \neq \phi$ **then**
        use the FNSSP algorithm to find $F(v)$ to cover $U(v)$,
        $Cnt_v := 0$
        $T_v := WAIT\_FOR\_ACK$
        broadcast $P(s, v, F(v))$ to $N(v)$.
      end if
    end if
  else
    $U(v) = U(v) - N(u) - \cup_{\forall x \in F(u)} N(x)$
  end if

When v receives a message $ACK(P(s), v, w)$ from w
  if $w \in F(v)$ **then**
    $F(v) = F(v) - w$
  end if

When timer T_v is expired
  if $(F(v) \neq \phi) \wedge (Cnt_v < MAX\_RETRY)$ **then**
    $Cnt_v := Cnt_v + 1$
    $T_v := WAIT\_FOR\_ACK$
    broadcast $P(s, v, F(v))$ to $N(v)$.
  end if

---

and the packet needs to be resent. u will resend the packet which attaches the remaining $F(u)$ that includes all un-replied forward nodes. When the forward nodes that are listed in $F(u)$ receive the packet, they have to resend ACKs to u. If $F(u)$ is not empty, the sender u will continue to resend the packet until the MAX_RETRY limit is reached. When this happens, u assumes the forward nodes that do not acknowledge are out of its transmission range and stops further broadcast attempts.

A node v that is not selected as a forward node, is covered by forward nodes at least twice so that its reliability can be improved. When it receives a broadcast packet from node u, v only updates its $U(v) = U(v) - N(u) - \cup_{\forall x \in F(u)} N(x)$. If $U(v)$ becomes empty, v does not need to forward the packet even if it may be selected as a forward node later.

IV. SIMULATIONS

A. Simulation Model

In order to analyze the performance of the proposed algorithm, we ran the simulation under the following network model: The working space is 100×100. Different numbers of nodes (range from 20 to 100) are randomly placed in this area. Each node has the same transmission range r and two neighbor nodes have a bi-directional link if their distance is

less than r. The network is generated with a fixed average node degree 6. The generated network that is not connected is discarded. We assume that the network is not error free and transmission collision and contention will happen even if the MAC layer schemes are applied. Also, due to the movement of the nodes, two nodes that are once neighboring may move out of range and miss the transmission. For simplicity, we use a constant probability p to describe all the affects that cause a broadcast transmission failure. The simulations are executed when p are 5% and 30%, respectively. We suppose that the ACK is sent via unicast and that it is reliable.

We assume that the network will delay t_b unit time for one-hop broadcast operations and the sender waits t_r unit time to resend the broadcast packet if transmission errors occur. The values of t_b and t_r affect the behavior of a node's retransmission. If the value of t_r is compared to t_b, the sender will resend a second copy of the broadcast packet once the first one is missed. In this case, the receiver will have the best chance to receive a broadcast packet from a shortest path from the source. If, on the other hand, the value of t_r is several times that of t_b, when a transmission error occurs, a node is likely to receive the message from another path by some other intermediate nodes' relaying. We run the simulation with $t_b = 2$ and $t_r = 10$. The simulation is repeated until the 99% confidence interval of the result is within $\pm 5\%$.

B. Results and Analysis

We compare the performances of the following algorithms through simulations to see the benefits and losses of the proposed reliable broadcast algorithm:

(a) *Blind flooding* (BF): Each node forwards the packet when it first receives the packet. No ACK mechanism.

(b) *Dominant pruning* (DP) [5]: Described in Section 2, no ACK mechanism.

(c) *Broadcast with selected acknowledgements* (BSA): Described in Section 3, forward nodes send ACKs.

(g) *AV reliable broadcast* (AVR) [1]: Described in Section 2, each receiver sends ACK.

We measure the following five metrics:

(a) *Broadcast delivery ratio*   Broadcast delivery ratio is the ratio of the nodes that received the broadcast packet to the number of the network. Figure 2 shows the broadcast delivery ratios when p are 5% and 30%. We can see that SBA and AVR maintain very good delivery ratio ($> 98\%$) even when $p = 30\%$. But the delivery ratio of DP decreases sharply when p increases. The delivery ratio of BF also drops remarkably when p is high.

(b) *Broadcast forwarding ratio*   Broadcast forwarding ratio is the fraction of the total number of nodes in the network that retransmit the broadcast packet at least once. Figure 3 shows that the broadcast forwarding ratios of AVR with $p = 5\%$ and $p = 30\%$ are almost 1 since every node forwards the packet when it first receives the packet. The BSA keeps almost the same value of broadcast forwarding ratio as the value of p increases and its broadcast forwarding ratio ranges from 0.6 to 0.7. p effects performance of the BF and
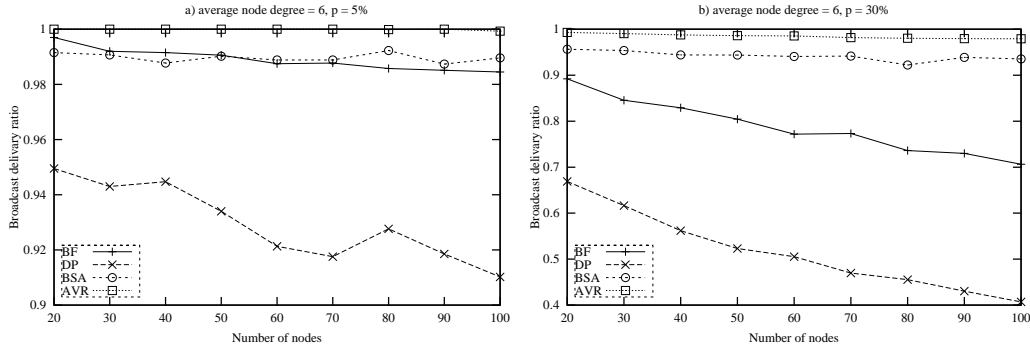
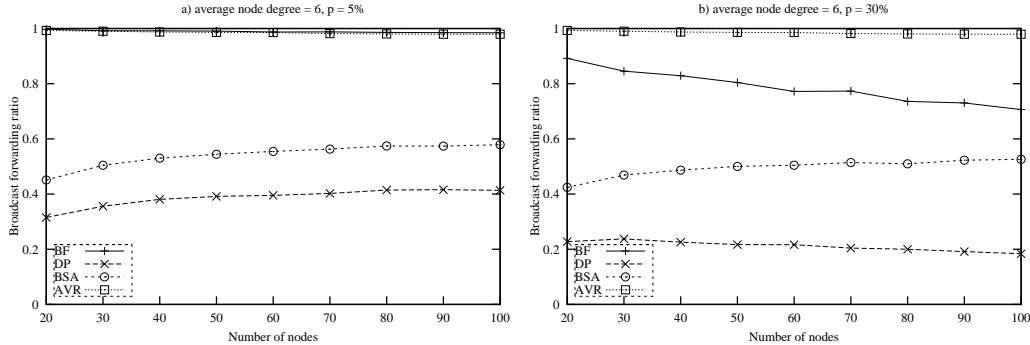Fig. 2. Broadcast delivery ratio: a) $p = 5\%$ and b) $p = 30\%$.

Fig. 3. Broadcast forwarding ratio: a) $p = 5\%$ and b) $p = 30\%$.

DP significantly, since these two algorithms have no reliable mechanism. When p is high, more receiving nodes miss the broadcast packet, resulting in the low broadcast delivery ratio.

(c) *Broadcast retransmission ratio*　Broadcast retransmission ratio is the average retransmission times of a forward node for a single broadcast operation. Figure 4 shows that the broadcast retransmission ratios of both BF and DP are 1 because they have no resend mechanism. The ratio of the BSA is less than that of the AVR. The increase of p causes the increase of the ratio for the BSA. The reason is that when p increases the sender needs more retries to successfully deliver a packet.

(d) *Broadcast end-to-end delay*　Broadcast end-to-end delay measures the period from the time the source broadcasted the packet to the time the last node receives the packet or no nodes re-send the packet for one broadcast operation. Figure 5 shows that with the ACK mechanism, the broadcast operation leads to more end-to-end delay. Also, higher p causes longer end-to-end delay for both BSA and AVR, but the delay for BSA is shorter than that for AVR.

(e) *ACK retransmission ratio*　ACK retransmission ratio is the ratio of the total number of ACKs to the number of the forwarding nodes. Only BSA and AVR have this metric. Figure 6 shows that the ACK retransmission ratio of the BSA is significantly less than that of AVR due to the ACK implosion problem.

From this simulation, we can see that using the ACK mechanism can greatly improve the delivery ratio, but result in

the increase of the broadcast retransmission ratio and broadcast end-to-end delay. Also, the simulation shows that the BSA retains the same delivery ratio as the AVR and outperforms the AVR in all other measured metrics.

V. CONCLUSION

In this paper, we have proposed a simple broadcast algorithm that provides high delivery ratio while suppressing broadcast redundancy. This is achieved by requiring only some selected forward nodes of 1-hop neighbor nodes to confirm their receipt of the packet. The forward node set selection process provides some redundancy so that retransmissions can be remarkably suppressed when the transmission error is considered. Primitive simulation results show that the proposed broadcast algorithm has the same high broadcast delivery ratio as the AVR and outperforms the AVR in forwarding ratio, broadcast retransmission ratio, broadcast end-to-end delay and ACK retransmission ratio for a broadcast operation. Our future work is to use more precise wireless transmission models, such as the IEEE 802.11 MAC model used in the ns-2 test-bed, to simulate the algorithm. Since the increase of the transmission of broadcast message and ACK message will greatly increase the collision of the transmission, the transmission error rate p is location dependent and time variant. The performance of the algorithm will be more vulnerable to the changes.
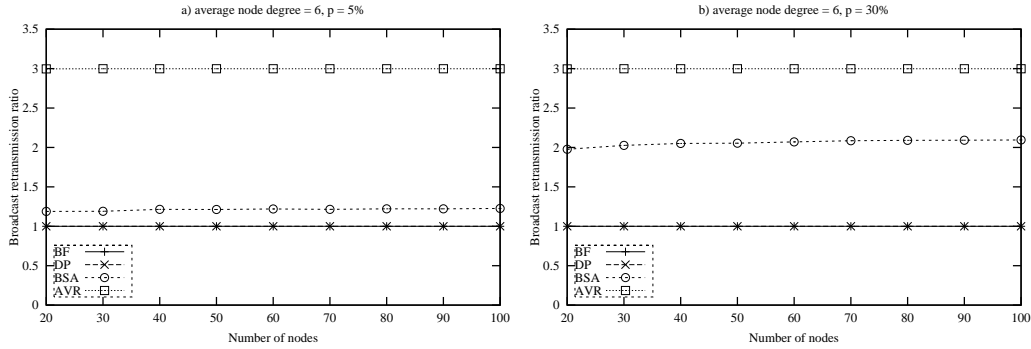
ACKNOWLEDGMENT

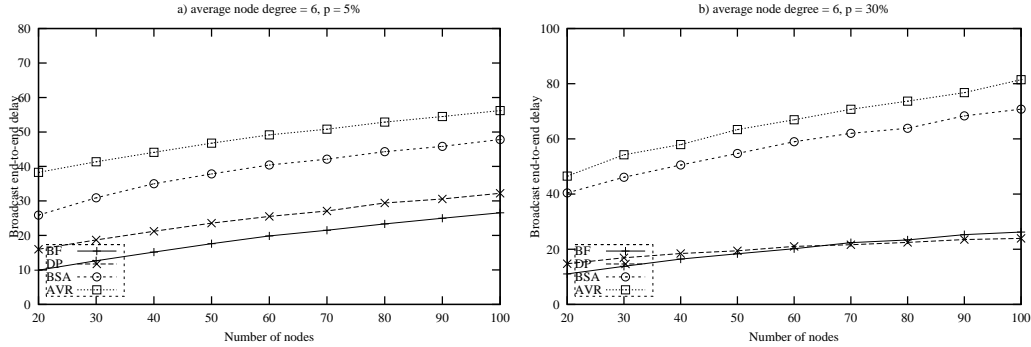Fig. 4. Broadcast retransmission ratio: a) $p = 5\%$ and b) $p = 30\%$.

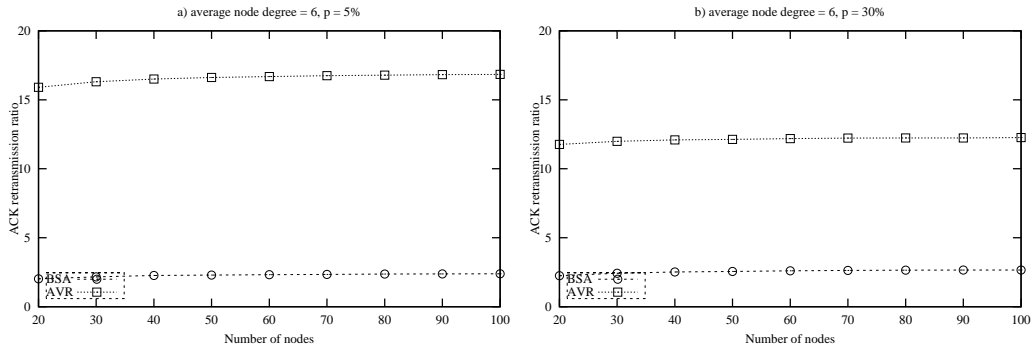Fig. 5. Broadcast end-to-end delay: a) $p = 5\%$ and b) $p = 30\%$.

Fig. 6. ACK retransmission ratio: a) $p = 5\%$ and b) $p = 30\%$.

REFERENCES

[1] S. Alagar, S. Venkatesan, and J. Cleveland. Reliable broadcast in mobile wireless networks. *Proc. of Military Communications Conference (MILCOM'95)*, pages 236–240, 1995.

[2] J. J. Garcia-Luna-Aceves and Y. X. Zhang. Reliable broadcasting in dynamic network. *Proc. of 1996 IEEE Int'l Conf. on Communications(ICC'96)*, 3:1630–1634, 1996.

[3] M. Impett, M. S. Corson, and V. Park. A receiver-oriented approach to reliable broadcast ad hoc networks. *Proc. of Wireless Communications and Networking Conference (WCNC'2000)*, 1:117–122, 2000.

[4] P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol. draft-ietf-manet-olsr-07.txt, 2002.

[5] H. Lim and C. Kim. Flooding in wireless ad hoc networks. *Computer Communications Journal*, 24(3-4):353–363, 2001.

[6] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Trans. on Mobile Computing*, 1(2):111–123, 2002.

[7] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59–68, 1995.

[8] P. M. Merlin and A. Segall. A failsafe distributed routing protocol. *IEEE Trans. on Communications*, 27(9):1280–1288, 1979.

[9] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. *Proc. of ACM/IEEE MOBICOM'99*, pages 151–162, 1999.

[10] E. Pagani and G. P. Rossi. Providing reliable and fault tolerant broadcast delivery in mobile ad hoc networks. *Mobile Networks and Applications*, 4:175–192, 1999.

[11] W. Peng and X. Lu. Efficient broadcast in mobile ad hoc networks using connected dominating sets. *Journal of Software*, 1999.

[12] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast message in mobile wireless networks. *Proc. of 35th Hawaii Int'l Conf. on System Sciences (HICSS-35)*, pages 3898–3907, 2002.

[13] D. Towsley, J. Kurose, and S. Pingali. A comparison of sender-initiated and receiver-initiated reliable multicast protocols. *IEEE Journal On Selected Areas in Communications*, 15(3):398–406, 1997.

[14] J. Wu and F. Dai. A generic distributed broadcast scheme in ad hoc wireless networks. *Proc. of ICDCS 2003*, pages 460–468, 2003.