Node Activation with Polling Channel Access

Long Yang and J.J. Garcia-Luna-Aceves
Computer Sciences and Engineering
University of California, Santa Cruz
California, Santa Cruz 95064
yanggz, jj@cse.ucsc.edu

***Abstract*— We present a new protocol for collision-free channel access in ad hoc networks called the Node Activation with Polling Access (NAPA) protocol. NAPA assumes a time-slotted channel and operates by having each node elect a transmitting node for each time slot based on the identifiers of the nodes in its two-hop neighborhood. In contrast to prior topology-dependent transmission scheduling schemes (e.g., Node Activation Multiple Access, or NAMA) in which time slots are wasted when nodes selected for transmission have no packets to send, NAPA complements the election of nodes by means of polling and carrier sensing to use time slots allocated to nodes with no data to send. When a node elected for transmission has no packets to send, it polls one or multiple one-hop neighbors, and each neighbor determines if it can transmit during the time slot based on the identifiers of its two-hop neighbors and sensing of the channel. We show that NAPA supports collision-free transmissions, and compare its performance against NAMA.**

I. INTRODUCTION

Medium access schemes of ad hoc networks can be categorized as contention-based or contention-free. A popular example of contention-based schemes is the DCF of 802.11. The main problems associated with contention-based schemes are that their throughput degrades quickly as node density and traffic load increase [4], [6], and that they do not support broadcast traffic efficiently. On the other hand, contention-free access schemes schedule a set of timetables for individual nodes or links, such that the transmissions from the nodes or over the links are conflict-free in the code, time, frequency or space divisions of the channel. The schedules for conflict-free channel access can be established based on the topology of the network, or it can be topology independent.

Recently, Bao and Garcia-Luna-Aceves [1], [2], [3] presented a number of topology-dependent transmission scheduling schemes that operate on the basis of the identifiers of nodes within a two-hop neighborhood of each node. In a nutshell, these schemes consist of: determining the contenders of the entity (node/link) for activation; computing the priority of every considered entity using a hashing algorithm, with the entity ID and current time slot (or any other type of contention period) as the seed; and choosing a subset of the suitable activated entities for data transmissions by comparing their priorities. The Node Activation Multiple Access (NAMA) protocol [1] operates on single-channel networks with omni-directional antennas, and has been shown to approach the performance of UxDMA-NAMA [5] while requiring only two-hop neighbor information. A limitation of NAMA and the other schemes based on two-hop neighbor information is that, although collision-free transmissions are attained, it is possible for a node to be given the opportunity to transmit in a time slot and not have any data to send, which wastes precious bandwidth.

In this paper, we present a simple approach for improving the performance of NAMA, as well as other scheduling schemes aimed at single-channel ad hoc networks with omni-directional antennas. We call our approach Node Activation with Polling Access (NAPA). As in NAMA, each node communicates to its one-hop neighbors the list of its own one-hop neighbors, so that any given node learns the identifiers of its one- and two-hop neighbors over time. Based on the list of its one and two-hop neighbors, each node obtains a priority list of nodes for each time slot, and elects the node with the highest priority to transmit in each time slot. When a node is elected for transmission and has data packets to send, it transmits as in NAMA; otherwise, it polls one or multiple one-hop neighbors using an ordered polling list to let one of them use the time slot. A node can poll a set of more than one neighbor only if those neighbors can all listen to one another. A node that is not selected for transmission in a given time slot listens to the channel. If a node receives a poll and has data to send, it uses its own priority list to determine if it can transmit, and remains silent if it is not the node with the highest priority after the polling node is excluded.

Section II describes the operation of NAPA and discusses why it supports collision-free transmissions when all nodes have consistent two-hop neighborhood data. Section III presents the results of simulations using the Qualnet simulation package comparing NAPA and NAMA; the results show that NAPA provides higher throughput and smaller delays by giving polled nodes the opportunity to transmit.

II. NODE ACTIVATION WITH POLLING

A. Assumptions and Notation

The operation of NAPA assumes that time is slotted and divided into frames. Each frame consists of control time slots and data time slots.

We assume that each node in the wireless ad hoc network is equiped with an omnidirectional antenna, and that the transmission range R is the same for all nodes. Moreover, we assume that a transmitting node does not interfere with other nodes located further than R from it. The one-hop neighbors of node i are all those nodes that are with node i's transmission range.

The set of one-hop neighbor nodes of a node i is denoted by N_i^1. Accordingly, we can define the set of two-hop neighbors of node i as $N_i^2 = \bigcup_{j \in N_i^1} N_j^1$.

B. Operation of NAPA

NAPA consists of two parts: exchanging two-hop neighborhood information among nodes, and selecting which nodes should transmit during each data time slot. Nodes use the control time slots of each frame to exchange their neighborhood information with one another. At the beginning of each data time slot, each node elects a node as the *holder* of the data time slot. The algorithm used for this election is the same as in NAMA, and hence we call it the *node activation algorithm*. Fig.1 shows the node activation algorithm, where the $hash(i,t)$ is used to compute a pseudo random number representing node i's priority at time t.

```
accessOk = TRUE;
$prio_i = hash(i,t)$;
for each $x \in N_i^2$ do
    $prio_x = hash(x,t)$;
    if $prio_x > prio_i$ then
        node $i$ cannot access channel;
        accessOk = FALSE;
        break;
    end if
end for
if acessOk == TRUE then
    node $i$ can access channel;
end if
```

Fig. 1. Node activation scheduling: Electing a data time slot holder at node i for data time slot t

The node activation algorithm implements a random permutation operation. Statistically, each node takes the same share of the common channel in the long run. NAMA, which is based on this election scheme, adapts easily to multihop ad hoc networks. Fig.2 helps to illustrate the basic shortcoming of channel access schemes based solely on node activation. Each edge in the graph shown in the figure represents bidirectional radio connectivity between nodes. Suppose that node X is elected as the holder of time slot t. Whether node X uses the time slot depends on whether X has backlogged data. When node X has no data, this time slot is wasted. This problem is more severe in high-density areas of a network in which several nodes have no data traffic. NAPA alleviates this problem by means of polling.

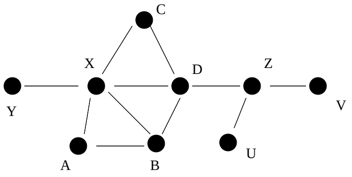

Fig. 2. Example network

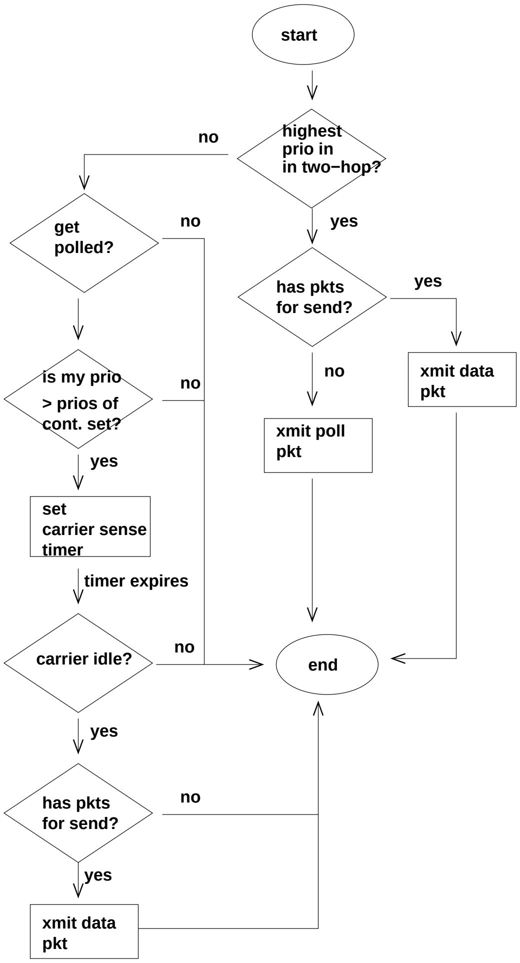

Fig. 3. Operation of NAPA in each data time slot

Because each node receives a list of one-hop neighbors from each of its neighbors, it can determine which of its one-hop neighbors form fully-connected sets of one-hop neighbors. Furthermore, when a node applies the node activation algorithm, it obtains a *node priority list* for a given data time slot.

A data time slot in NAPA is assumed to be long enough for a node to be able to listen at the beginning of the time slot for a polling packet or the presence of carrier in the absence of such a packet, and to turn around to transmit mode if the polled node determines that it can transmit. Figure 3 shows a flow chart of how NAPA operates at each node during a data time slot.

At the beginning of each data time slot, each node runs the node activation algorithm to select the holder of the time slot. If the holder of the time slot has data to transmit, it goes ahead and transmits its packet to the intended receivers. Alternatively, if the holder of the time slot has no packets to transmit, it selects one or more one-hop neighbors to poll, and transmits a short polling packet containing an ordered list of the polled node(s). The list of polled nodes consists of the one-hop neighbor h with the highest priority and all the other one-hop neighbors that are neighbors of h and of each other. All the polled nodes must be one-hop neighbors of each other, so that they can apply carrier sensing on each other's transmission attempts. The list of polled nodes is ordered in descending order based on their priorities. For a polled group $G = \{\ldots, i, j, \ldots\}$, the nodes in this group are ordered according to their priority, $prio.x, x \in G$.

A node that is not the holder of a time slot listens for a polling packet. If the polling packet is received correctly and specifies the receiving node in its list of polled nodes, it proceeds to determine if it can transmit a data packet in response to the poll. A polled node simply remains quiet for the remainder of the time slot if either (a) it has no data packet to send, or (b) it has at least one neighbor with a higher priority than its own according to the node's priority list with the polling node deleted from the list.

A polled node with a packet to transmit and with the highest priority in its node priority list with the polled node deleted listens to the channel for a waiting time period proportional to its position in the list of polled nodes specified in the polling packet. If the node detects carrier before the waiting time period elapses, it remains quiet for the rest of the time slot; otherwise, it transmits its data packet. For a node i in the group of polled node G, the waiting time period is $(rank_i + 1) * CS.interval$, where $CS.interval >$ *propagation delay of signal* and $rank.i$ is the rank of the priority of i in group G. Because the nodes in a polled group can hear each other by the fully-connected set constraint, when a node in this group transmits, other nodes in this group must find the channel busy.

The simplest scheme in NAPA consist of polling a single one-hop neighbor when the holder of the time slot does not have data to send. Our simulation results presented in Section III show that even this simple version of NAPA performs better than NAMA.

C. Analysis

In this section, we study the conflict-free transmission property of NAPA.

Theorem 1: When nodes have consistent two-hop neighborhood information and correct carrier sensing can be enforced, the channel access schedules derived with NAPA are free of conflicts, in the sense that no data packet collides with other packets.

Proof: Data transmissions in NAPA take place during a given time slot after a node determines that it has the highest priority in its two-hop neighborhood fo the given time slot. There are two cases to consider for the case in which node X has the highest priority in its two-hop neighborhood for a given time slot T. In the first case, node X has a data packet to transmit. The proof for this case follows from the correctness of NAMA [1]. In the second case, node X selects a set of one-hop neighbors S_X that are fully connected. Each node $n_i \in S_X$, $i = 1, \ldots, |S_X|$, transmits a data packet only if (a) no node in S_X with a higher priority than node n_i has started to transmit (thus producing carrier) before its waiting time period elapses, and (b) node n_i has the highest priority in its own two-hop neighborhood excluding node X and the nodes in S_X that have not started to trasnmit before n_i's waiting time period elapses. Accordingly, because no node in S_X can start transmitting before its waiting time period elapses and because the order of nodes in S_X is the same for all such nodes, node n_i's transmission cannot collide with the transmission of any node in S_X, and no node in the two-hop neighborhood of n_i that is not in $S_X \cup X$ can also transmit in the same time slot.

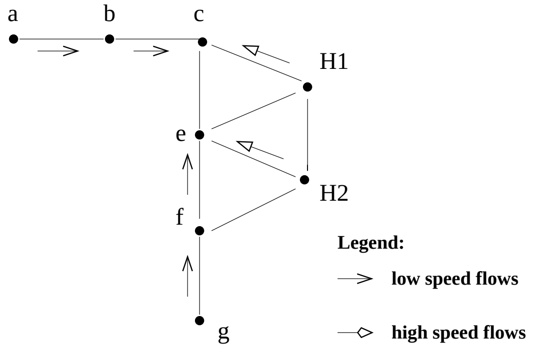

Fig. 4. A network scenario

III. PERFORMANCE COMPARISON

In this section, we compare NAPA and NAMA via simulation. We implemented both protocols using the Qualnet simulator. The slot lengths of NAPA and NAMA are slightly different, because NAPA needs to accommodate the polling packet. The percentage of time slots dedicated to the transmission of two-hop neighborhood information is the same at 7.8% for both protocols. The data packet size is 1148 bytes. The link bandwidth is 2 Mbps. The portion of a data time slot dedicated to polling in NAPA is 6.6%.

In the scenario shown in Fig. 4, there are eight nodes in the network. The simulation results show that, as should be expected, when every node always have data packet to send whenever it is elected as the holder of a time slot, the performance of NAPA is almost the same as NAMA. Simulation runs with ON-OFF flows were used to study the difference in performance between NAPA and NAMA. Simulations last 30 seconds.

In the first experiment, each node has a high speed CBR flow whose destination is one of this node's neighbor nodes, say they

Ave thr (Kbps)		
source	NAPA	NAMA
a	436.4	455.3
b	292.6	290.5
c	205.5	206.8
e	215.6	219.2
f	269.5	264.2
g	324.8	333.6
H1	249.0	247.2
H2	248.1	252.6
total	2241.4	2269.4

Fig. 5. Net throughout comparison of NAPA and NAMA

	Ave thr (Kbps)			delay (s)		
in(Kbps)	NPg	NPs	NM	NPg	NPs	NM
s(102.8)	102.8	102.8	102.8	0.0278	0.0289	0.0345
f(239.2)	239.2	239.2	211.2	0.0282	0.0979	0.581
s(102.8)	102.8	102.8	102.8	0.0285	0.0291	0.0345
f(473.6)	422.0	342.5	243.2	0.278	0.452	0.778
s(102.8)	102.8	102.8	102.8	0.0288	0.0291	0.0345
f(590.2)	447.3	352.9	248.2	0.329	0.491	0.797

Fig. 6. Performance comparison of NAPA and NAMA. Average inout rates of low speed flows are kept at 102.8 Kbps. The left column lists input rates of low speed slow flows, and input rates of high speed fast flows. Three sets of experiment result are listed in this table.

are a-b, b-c, c-H1, H1-e, e-H2, H2-f, f-g and g-f. Other than the delivery ratio and delay, we are interested in the net throughput reached by both NAPA and NAMA. Fig.5 shows that NAPA and NAMA attain much the same throughput when each node always has packets ready to send . The minor differences are due to the fact that NAPA has to allocate a small portion of time in each time slot for the possible dedicated polling packet and some time for carrier sensing operation.

In the second experiment, there are four low-speed flows: a-b, b-c, g-f, f-e, and two high-speed flows: H1-c and H2-e, as shown in Fig.4. We use ON-OFF traffic model for all nodes, attempting to simulate the bursty characteristics of arrival patterns. The lengths of ON period and OFF period obey an exponential distribution. During an OFF period, a node does not generate any data packets. During an ON period, a node generates packets at a constant rate. In all simulation runs, the mean of ON and OFF periods is set at 0.5 sec. As an interesting scenario and for simplicity, we keep the data rates of low-speed flows the same, and data rates of all high-speed flows the same. Moreover, data rates of low speed flows in different experiments are the same.

Fig.6 shows the average throughout and delay of the slow flows and fast flows. Three sets of results of different flow combinations (low speed, high speed) are presented, and we keep the data rate of low-speed flows in all experiments the same. NAPA with group polling is denoted by NPg, and NAPA with a single node polled is denoted by NPs, NAMA is denoted as NM. All input CBR flows are denoted as “in” in the tables. “s” means a low-speed flow, “f” means a high-speed flow.

Fig.6 shows that NAPA attains higher throughput than NAMA by reusing time slots that are otherwise wasted in NAMA. Note that, for those nodes that only transmit low data rate flows, some time slots assigned to them by NAMA can be wasted due to the low data rate and the ON-OFF nature of the traffic. NAPA attempts to reuse some of those time slots. When the data rate of high-speed flows increases, NAPA delivers higher throughout than NAMA. Because nodes get more chances to access the channel, delay is decreased in NAPA. Here we also note that the performance of NPg is different with NPs. Even only polling one neighbor as in NPs provides an improvement in this scenario.

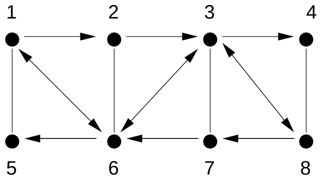

Fig. 7. A multihop scenario

	Tot. thr (Kbps)			delay (s)		
in(Kbps)	NPg	NPs	NM	NPg	NPs	NM
124.6	379.9	348.9	244.6	1.166	1.199	2.069
164.5	336.3	317.8	243.7	1.717	1.730	2.459
244.1	316.2	309.8	249.2	2.078	2.101	2.587

Fig. 8. Total throughput and delay. The leftmost column lists the average rate of each input flow

Next we look at a mix scenario depicted in Fig.7. While multiple flows exist, 1-2-3-4, 1-6-3-8, 8-7-6-5, 8-3-6-1. We are interested in the total throughput attained by the four flows. We use ON-OFF traffic as in the previous experiment. Table.8 lists the total throughput attained for different data rates for ON-OFF traffic. Both NAMA and NAPA do not reach high delivery ratio because both are saturated.

the results show shows that NAPA outperforms NAMA in this multihop scenario. Not surprisingly, NPg attains higher throughout and smaller delay than NPs.

Finally, we look at a random 30-node network in Fig.9. All links are shown by solid lines and the topology is static. We randomly choose 15 nodes to broadcast. We use CBR flows in this simulation and we are interested in the average throughput and delay. From Figure 10 we note that both NAPA and NAMA maintain quite stable throuput during different load situations. NAPA reaches higher throughput than NAMA, while the delay of NAPA is smaller than that of NAMA. Becuase there are 15 nodes that do not broadcast in this simulation scenario, that leaves time slots underutilized in NAMA. In all three cases, NPg outperforms NPs, because NPg simply gives nodes for more opportunities to transmit.

From all the experiments, we observe that NAPA increases throughput while decreasing the average delay compared with NAMA. The simulation-based eveluation shows that even a simple polling scheme without carrier sensing helps to improve the performance of NAMA.

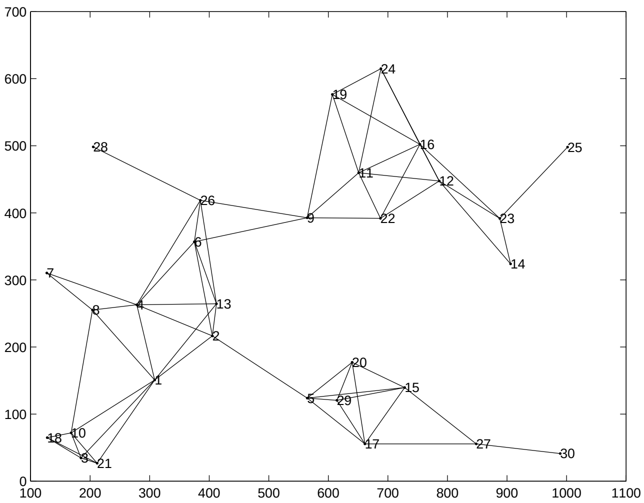

Fig. 9. A random network

input	Ave thr (Kbps)			delay (s)		
in(Kbps)	NPg	NPs	NM	NPg	NPs	NM
229.6	212.0	203.7	173.2	0.405	0.664	1.211
306.1	237.9	214.8	180.2	0.868	1.140	1.362
459.2	242.5	215.1	182.0	1.067	1.219	1.464

Fig. 10. Average throughput and delay

IV. CONCLUSION

In this paper, we propose NAPA as an extension of NAMA. NAPA operates in time-slotted scheme and is a conflict-free scheduling protocol for broadcast. NAPA improves on NAMA by introducing group polling and carrier sensing mechanism into NAMA while keeps the conflict-free property of NAMA. Simulation experiments show that NAPA provides higher slot utilization than NAMA, which results in higher throughput and smaller average delays than NAMA.

V. ACKNOWLEDGEMENT

This work was supported in part by the US Air Force/OSR under grant F49620-00-1-0330 and by the Baskin Chair of Computer Engineering at UCSC

REFERENCES

[1] L. Bao and J.J. Garcia-Luna-Aceves. A new approach to channel access scheduling for ad hoc networks. In *Proc. ACM Seventh Annual International Conference on Mobile Computing an Networking*. Rome, Italy, July 2001.

[2] L. Bao and J.J. Garcia-Luna-Aceves. Distributed dynamic channel access scheduling for ad hoc networks. In *JPDC, Special Issue on Wireless and Mobile Ad Hoc Networking and Computing*. Academic Press, 2002.

[3] L. Bao and J.J. Garcia-Luna-Aceves. Hybrid channel access scheduling in ad hoc networks. In *Proceedings of IEEE Tenth International Conference on Network Protocols (ICNP)*, November 2002.

[4] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal On Selected Areas In Communications*, 18, March 2000.

[5] R. Ramanathan. A unified framework and algorithm for channel assignment in wireless network. *Wireless Networks*, 5(2):81 –94, 1999.

[6] Y. Wang and J.J. Garcia-Luna-Aceves. Performance of collision avoidance protocols in single-channel ad hoc networks. In *Proc. of IEEE Tenth International Confrence on Network Protocols (ICNP)*, 2002.