Mobile Networks and Applications 7, 493–502, 2002

An Adaptive Generalized Transmission Protocol for Ad Hoc Networks

ANDREW D. MYERS, GERGELY V. ZÁRUBA and VIOLET R. SYROTIUK
Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75083, USA

Abstract. Wireless networking and group communication in combination allows groups of dispersed mobile users to collaborate. This paper presents AGENT, a medium access control (MAC) protocol that unifies point-to-point and multi-point transmission services to facilitate group communication in ad hoc networks. Analysis and experiments performed in a simulated ad hoc network demonstrate that AGENT exhibits reliable and stable performance with high spatial bandwidth reuse. Moreover, variation in the proportion of point-to-point and multi-point traffic is shown to have little impact on the overall performance of AGENT. Comparison with the other tested MAC protocols reveals that the performance of AGENT is superior, achieving higher channel utilization and lower access delay.

Keywords: medium access control (MAC), mobile ad hoc networks (MANETs), unified unicast/broadcast primitives

1. Introduction

Wireless technology presents users with instantaneous communication and ubiquitous computing capabilities regardless of their current location. In contrast to a cellular network, an ad hoc network consists of a group of nodes that collectively form a multi-hop wireless network. Ad hoc nodes directly exchange packets across shared communication channels without the aid of any communications infrastructure. Due to the limitations of wireless transmission, packet exchanges between distant nodes must be relayed through intermediate nodes in a hop-by-hop fashion. Consequently, an ad hoc network must display a high level of self-organization and adapt to fluctuations in network connectivity. These characteristics enable ad hoc networks to support the rapid deployment of temporary communication and information access solutions.

In the most general sense, multi-point (or group) communication involves multiple participants exchanging information. Some examples include the exchange of audio and video streams during a video conferencing session, and the sharing of text and graphics in many computer-supported collaborative work (CSCW) applications. Effective group communication allows users to collaboratively interact in natural and intuitive ways. This type of interaction is essential to many of the proposed applications of ad hoc networks, such as battlefield coordination and disaster relief [16], that involve the close cooperation of large numbers of users.

Most recent research efforts have focused on developing network and transport layer protocols for group communication in ad hoc networks [1,3,7,8]. However, such multi-hop transmission services ultimately rely on effective, single-hop packet transmissions that are controlled by the medium access control (MAC) layer. For example, most routing protocols typically require broadcast transmission services to exchange connectivity or location information; multicast transmission services allow a multicast routing protocol to forward a packet along a tree or mesh; and virtual circuit services and single source-destination routing is more efficiently supported by unicast transmission services. In practice, high-layer protocols use a combination of these single-hop services. Thus, a MAC protocol that effectively supports this suite of single-hop transmission services is required.

This paper introduces AGENT, an adaptive MAC protocol for ad hoc networks that provides a unified set of effective single-hop transmission services. AGENT employs a hybrid design that combines an allocation and contention based protocol. The allocation protocol gives each node guaranteed access to the channel, providing access delay bounds and preventing instability. The contention protocol allows nodes to claim idle slots, thus obtaining spatial bandwidth reuse. The protocol features efficient support for unicast, multicast, and broadcast packet transmissions, and incorporates prioritized channel access to ensure cooperation between the allocation and contention components.

The rest of this paper is organized as follows. Section 2 presents a brief overview of MAC protocols that emphasizes point-to-point and multi-point transmission service support. In section 3, we motivate the need for a protocol offering these services and describe the operation our AGENT protocol. Analysis of AGENT is presented in section 4, where we examine its performance and reliability. We then evaluate the performance of AGENT in a simulated ad hoc network and compare it to other MAC protocols in section 5. Finally, we summarize our conclusions and outline future research in section 6.

2. Previous work

The carrier sense multiple access (CSMA) protocol was one of the earliest contention protocol designs for ad hoc networks [14]. Developed by Kleinrock and Tobagi, CSMA addresses the half-duplex nature of wireless communication by having nodes first listen for channel activity before transmitting their packets. However, CSMA suffers from hidden terminal interference since a node is unable to determine the status of the channel at its destinations. This leads to more frequent packet collisions, reducing the performance of the protocol.

To address this problem, Karn introduced a collision-avoidance mechanism involving a request-to-send/clear-to-send (RTS/CTS) control packet exchange between a source node and its intended destination [10]. Before transmitting its packet, a source node transmits a RTS expecting a CTS response. If the RTS is successfully received, the destination responds with a CTS; otherwise it remains silent. Furthermore, each RTS/CTS control packet contains the length of time needed to transmit the packet. Thus, any node that overhears this handshake refrains from accessing the channel for the specified duration. Once the CTS is received, the source node is free to transmit its packet.

Over the years, many variations and combinations of these two basic techniques have been proposed, including the commercially available IEEE 802.11 standard [17]. However, many of these contention protocols are subject to instability, i.e., throughput breakdown, at high traffic loads. Furthermore, the point-to-point nature of the collision-avoidance mechanism limits its effectiveness in supporting reliable multi-point transmissions. In practice, multi-point packets are typically transmitted using CSMA-style channel access, with similar reception probabilities. This can be made more reliable by using a "repeated unicast" scheme, i.e., sending a copy of a packet to each addressed neighbor. However, our analysis and simulation results demonstrate the inefficiency of such a technique [6].

The time division multiple access (TDMA) protocol was one of the earliest allocation protocol designs for ad hoc networks. In this case, time is divided into fixed size slots which are then organized into a synchronous frame. Each node is assigned one unique slot per frame in which it is given exclusive access to the channel, thus unicast, multicast and broadcast packets are easily accommodated. The length of a TDMA frame is proportional to N, where N is the number of nodes in the network. Shorter frame lengths can be achieved via more complex slot assignments or dynamic slot assignments. In [4], Chlamtac and Faragó realized a static frame length that scaled logarithmically with N and quadratically with the maximum node degree. In [15], Zhu and Corson dynamically determine the frame length according to the local network topology. A static frame length bounds access delay, yet transmission concurrency is limited by global network parameters that are typically unknown and time varying. While dynamic slot assignment schemes can overcome these limitations, the need for a contention protocol to reorganize transmission schedules can lead to instability.

One of the first hybrid MAC protocols was developed by Sharp, Grindrod, and Camm in [12]. The protocol features a combination of both TDMA and CSMA channel access schemes. Each node is permanently assigned a certain number of TDMA slots in which it has priority to access the channel. If a slot is not used by the assigned node, other nodes may attempt channel access at a random instant after the start of the slot. To alleviate hidden terminal interference, nodes are not permitted to access time slots allocated to nodes exactly two hops away from them. The main disadvantage of this protocol is that nearly half of each idle slot is lost accommodating randomization. Furthermore, reliable multicast or broadcast cannot be assured in an idle slot.

The collision avoidance time allocation (CATA) protocol, developed by Tang and Garcia-Luna-Aceves in [13], directly addresses the hidden terminal problem in [12] by replacing CSMA with a collision-avoidance mechanism. It also features support for both point-to-point and multi-point packet transmissions, as well as on-demand slot reservations. However, in CATA there are no permanent slot assignments, and access to each slot is resolved through contention. Consequently, instability can arise in situations where network load and node connectivity are high, as shown in section 5.

To summarize, each of the above protocols are not well suited to support efficient group communication in ad hoc networks. They either lack the reliable multi-point transmission services needed by higher layer protocols, inefficiently use the bandwidth resources, or suffer from protocol instability. Our AGENT protocol addresses each of these shortcomings.

3. An adaptive generalized transmission protocol (AGENT)

3.1. Model and notation

We represent an ad hoc network as an undirected graph $G = (V, E)$, where V is the set of $|V| = N$ nodes and E is the set of bidirectional, wireless links. We assume that the network can be embedded in a two-dimensional convex area A. For two nodes i and j, $\text{dist}(i, j)$ is a function that returns the Euclidean distance separating them, and $\text{link}(i, j)$ is a logical function that returns *true* if $\text{dist}(i, j) \leqslant r$, where r is the transmission range of each node, and *false* otherwise. $H'(i) = \{k \in V | \text{link}(i, k)\}$ is the set of one-hop neighbors of node i. We assume that communication is perfectly synchronized, half-duplex, and that the simultaneous arrival of two or more packets at a node results in a collision, i.e., no capture.

3.2. Motivation for AGENT

We first explore the difference in concurrency that arises in the network between point-to-point and multi-point transmissions. For a source node s to successfully unicast a packet to a destination d, all nodes in $H'(d) - \{s\}$ must not transmit concurrently to prevent collision at d. For s to successfully broadcast a packet, all nodes in $\bigcup_{d \in H'(s)} H'(d) - \{s\}$ must not transmit concurrently. Spatial bandwidth reuse, and therefore MAC protocol performance, is directly dependent on the number of nodes that must not transmit concurrently, i.e., the neighborhood size, in a given transmission attempt. Thus, we begin by approximating average neighborhood sizes.

Geometrically, the probability that two nodes are neighbors is $\pi r^2/A$. Thus, the average number of nodes in a one-hop neighborhood, η', is approximated by

$$\eta' = \frac{\pi r^2 N}{A}. \tag{1}$$

For any two such connected nodes, the cumulative distribution function of the distance x separating them is given by $F(x) = x^2/r^2$, where $0 \leqslant x \leqslant r$. The probability distribution function is $f(x) = F(x)\,\mathrm{d}/\mathrm{d}x = 2x/r^2$, and the expected value of x is $\mathrm{E}[f(x)] = 2/(3r)$.

The radius of the average two-hop neighborhood is approximately $2/(3r)$ more than the expected value of x. Thus, the probability that a node resides in this area is $16\pi r^2/(9A)$, and the average number of nodes in a two-hop neighborhood, η'', is approximated by

$$\eta'' = \frac{16\pi r^2 N}{9A} = \frac{16}{9}\eta'. \tag{2}$$

In order to express the tradeoff between unicast and broadcast, let us consider a saturated network where each node has a packet to send in every slot. The probability of a successful transmission, p_{succ}, is then $\tau(1-\tau)^{1-\eta}$, where τ is the probability that a node transmits in a slot and η is the neighborhood size. Through differentiation we find that p_{succ} is maximized when $\tau = 1/\eta$. Thus,

$$p_{\text{succ}} = \frac{1}{\eta}\left(1 - \frac{1}{\eta}\right)^{1-\eta}. \tag{3}$$

The inverse of (3) yields the average time needed for a successful transmission, i.e., the average access delay. Those situations where a single broadcast is more effective than repeated unicast occur when

$$\eta' D_{\text{u}} - D_{\text{b}} \geqslant 0, \tag{4}$$

where D_{u} and D_{b} denote the respective average access delay of unicast and broadcast. We find that broadcast is more effective than repeated unicast when $\eta' \geqslant 16/9$. Thus, except for extremely sparse networks, unicast MAC protocols cannot effectively support multi-point communication; yet broadcast is not effective for unicast transmissions due to reduced concurrency. Therefore, a MAC protocol that effectively supports both point-to-point and multi-point traffic in a unified manner is justified.

3.3. AGENT protocol description

Underlying AGENT is a TDMA allocation protocol in which node i is assigned a unique slot s_i, $1 \leqslant s_i \leqslant N$, in a frame of length N. This guarantees each node access to the channel once per frame, bounding delay under high loads and dense connectivity. To take advantage of the potential for spatial bandwidth reuse, AGENT uses signalling similar to that used in collision-avoidance protocols. In order to facilitate transmission concurrency, each TDMA slot is subdivided into a priority, contention and transmission interval (see figure 1). The priority interval is used to signal nodes about activity in an assigned slot; the contention interval gives nodes an opportunity to use a slot provided that transmission will not interfere with that of the slot owner; and the transmission interval is used to transmit a unicast, multicast, or broadcast packet.

To gain access to the transmission interval of a slot s, a source node i first transmits a RTS control packet. The RTS is either sent at the beginning of the priority interval, if $s = s_i$, or at the beginning of the contention interval, otherwise. Reception of a RTS in the priority interval elicits a CTS response from a destination. Notice that in the case of a multi-point packet, there will be a collision of responses at i. This is not a concern, since the purpose of these CTS responses is to inform the neighbors of each destination of i's intention to use its assigned slot. On the other hand, reception of a RTS in the contention interval will generate a CTS response only when it is associated with a unicast packet. Any node that detects a collision among RTS control packets will reply with a not-clear-to-send (NCTS) control packet.

Once the initial control signalling is finished, a node can determine its eligibility to transmit its packet p in the transmission interval. If $s = s_i$, then source node i is granted permission to transmit p without restriction. Otherwise, the following rules must be applied:

1. If any control signalling is detected in the priority interval, then i must withhold the transmission of p to avoid conflict with the owner of s.
2. If a NCTS response is received in the contention interval, then multiple source nodes are contending for s, and i must withhold the transmission of p to avoid collision.
3. If p is a unicast packet and a corresponding CTS is received, then i may transmit p.
4. If p is a multi-point packet and no signalling response is received in the contention interval, then i may transmit p.

Any failure to transmit p in an unassigned slot is resolved by a *backoff algorithm* that is based on the exponential backoff scheme developed in [9]. Using local network feedback, a node decreases τ by a factor of $1/c$ when it detects a failed transmission attempt in a slot. A failed transmission attempt occurs when a negative control response is received, e.g., a NCTS response, or a collision is detected in the contention interval. On the other hand, τ is increased by a factor of c when a slot remains idle. In [9], $c = 2$ was shown to yield optimal results.

For example, consider the five node network of figure 2. The current slot is assigned to node 3, which has a multicast packet addressed to nodes 1 and 2, and node 4 has a unicast packet addressed to node 5. Then 3 sends a RTS at the beginning of the priority interval (figure 2(a)) to which 1 and 2 respond with a CTS (figure 2(b)). Node 4 sends a RTS at the start of the contention interval (figure 2(c)), and 5 sends a CTS response (figure 2(d)). At this point, 3 is free to send its multicast packet in the transmission interval, since this is its assigned slot. However, 4 must refrain from sending its unicast packet, since it detected the RTS of 3 (see figure 2(a)) in the priority interval.

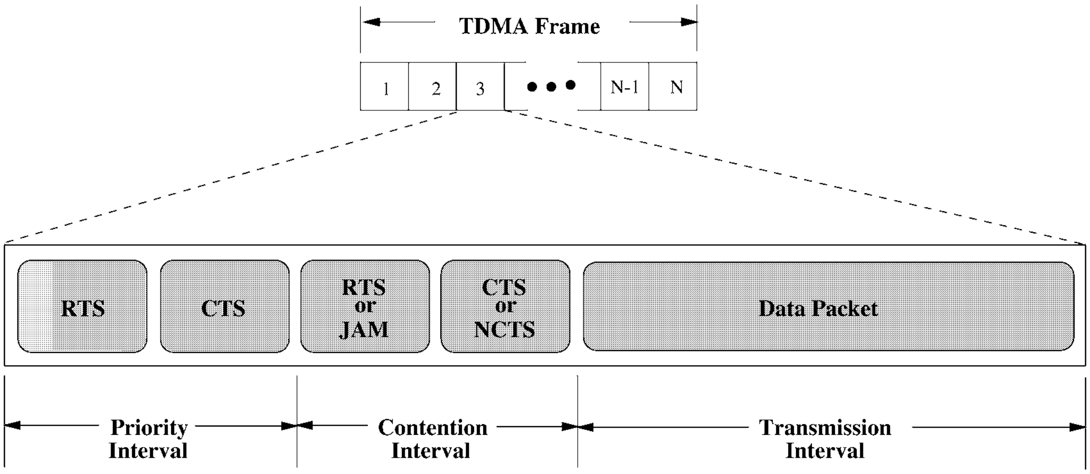

Figure 1. Frame and slot structure of the AGENT protocol.

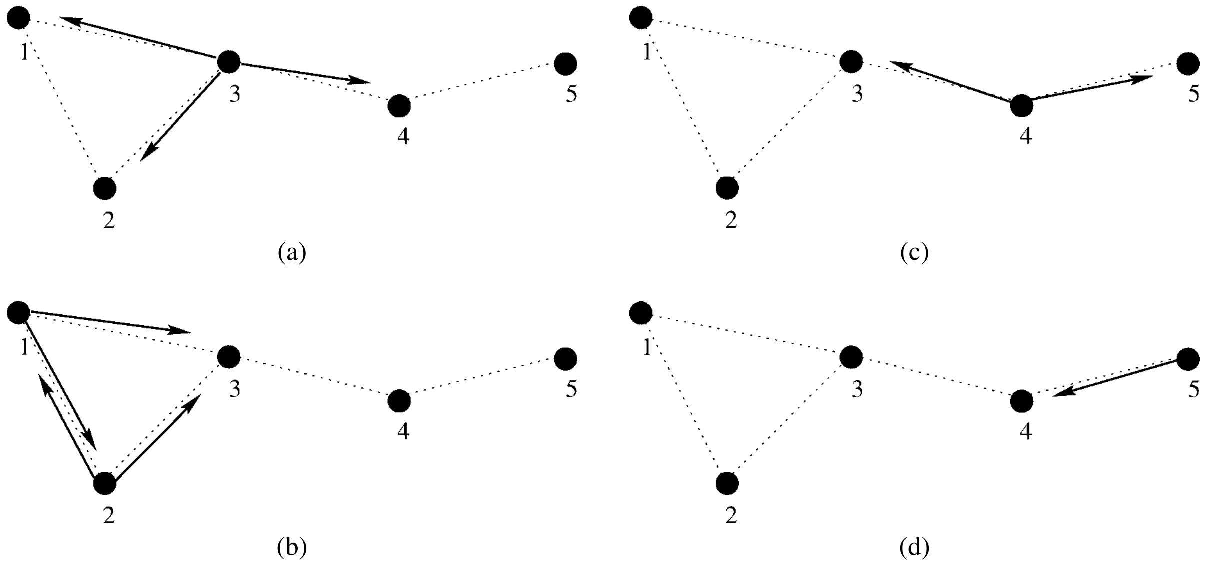

Figure 2. Example of AGENT signalling.

Notice that the concurrent transmissions of nodes 3 and 4 would not cause any interference at any destination node in the above scenario. This increased concurrency can be accommodated by amending the first rule to read: "If any *CTS control signalling* is detected in the priority interval, i must withhold the transmission of p to avoid conflict with the owner of s".

We can further enhance the protocol by eliminating unnecessary control signalling. Specifically, a node that is attempting to send a packet in an unassigned slot can immediately evaluate the amended rule at the end of the priority interval. If it evaluates to true, then there is no reason to send a RTS in the contention interval. This will reduce the number of control packets sent, and increase energy savings.

One further amendment is needed to handle special cases involving multi-point packets. For example, if the roles of nodes 4 and 5 are switched in figure 2, then following scenario arises. Node 5 is unable to determine whether 3 is using its assigned slot, and will send a RTS in the contention interval. According to AGENT, the proper response of 4 is a CTS, only this would lead to a collision at 4 in the transmission interval. If 4 does not send a CTS, then 5 would not send its unicast packet. However, with a multi-point packet, 5 would still attempt packet transmission. To avoid this ambiguity, a NCTS response is needed. Thus, the node sending a RTS in the priority interval also sends a jamming RTS (JAM) at the start of the contention interval. This will cause a collision with any incoming RTS control packets, and elicit the proper NCTS response.

Figure 3 presents the specification of the AGENT protocol in pseudocode. For ease of presentation we assume each destination is in range of the source node. Each node maintains a queue of packets to transmit and a transmission probability (τ). Each packet header contains a source and destination identifier and a type field which is *unicast*, *multicast*, or *broadcast*. The destination contains one or more node addresses (for unicast and multicast, respectively) or a broadcast address. Nodes execute *Send*(*packet*, *interval*) and

```
AGENT() {
  for each slot s do {
    if queue $\neq$ empty then {
      if s = assigned then {
        Send(RTS, priority);
        Send(JAM, contention);
        Send(PKT, transmission);
      }
      else if Recv(RTS, priority) = success and
            RTS.dest = this_node then {
        Send(CTS, priority);
        Recv(pkt, transmission);
      }
      else if Recv(CTS, priority) = idle then {
        if Contend($\tau$) then {
          Send(RTS, contention);
          case RTS.type
            unicast:
              if Recv(CTS, contention) = success then
                Send(pkt, transmission);
              else $\tau \leftarrow \tau/2$;
            multicast, broadcast:
              if Recv(NCTS, contention) = idle then
                Send(pkt, transmission);
              else $\tau \leftarrow \tau/2$;
          end case
        }
        else Passsive();
      }
    }
    else Passive();
  }
} /* end AGENT() */

Passive() {
  if Recv(RTS, priority) = success and
   RTS.dest = this_node then {
   Send(CTS, priority);
   if Recv(RTS, contention) = collision then
     Send(NCTS, contention);
   Recv(pkt, transmission);
  }
  else {
   status $\leftarrow$ Recv(RTS, contention);
   case status
     success:
       if RTS.type = unicast and
        RTS.dest = this_node then {
        Send(CTS, contention);
        Recv(pkt, transmission);
       }
       else Recv(pkt, transmission);
     collision:
       Send(NCTS, contention);
       $\tau \leftarrow \tau/2$;
     idle:
       $\tau \leftarrow 2\tau$;
   end case
  }
}/* end Passive() */
```

Figure 3. AGENT specification.

Recv(*packet*, *interval*) to transmit and receive packets in the specified interval, i.e., *priority*, *contention*, or *transmission*. *Recv* also sets a return code of *success*, *collision*, or *idle* to indicate whether the packet reception was successful or not, or that the channel was idle during the interval, respectively. The statement *Contend*(τ) returns *true* if the node may attempt transmission in an unassigned slot, and *false* otherwise.

4. Analyses of AGENT

4.1. Performance analysis

In this section, we present an approximate analytical framework for evaluating the performance of AGENT. To simplify our presentation, we assume that the network topology is static and that the traffic load distribution is homogeneous.

With AGENT, there are two cases that need to be analyzed. For a given node, a slot is either assigned or unassigned. Let φ be the probability that a node has a packet to transmit, and let η denote the neighborhood size. The probability that a node transmits its packet in its assigned slot is φ/N, since there is a $1/N$ probability that a slot is assigned. A node may also attempt to transmit in an unassigned slot. The probability that a node contends for an unassigned slot can be expressed as $\tau(1-\varphi/N)^{\eta}$, where $0 \leqslant \tau \leqslant 1$ is the probability that a node transmits in a slot. Then the probability that its contention is successful, ϑ, is

$$\vartheta = \eta\left(1-\tau\left(1-\frac{\varphi}{N}\right)^{\eta}\right)^{\eta-1}\tau\left(1-\frac{\varphi}{N}\right)^{\eta}. \tag{5}$$

By combining the probabilities associated with the assigned and unassigned slots, we can approximate a node's average throughput as $T_{\text{node}} = \varphi/N + \vartheta$.

Using differentiation, we find that the maximum average throughput of a node occurs when the parameter τ of (5) equals $1/(\eta(1-\varphi/N)^{\eta})$, yielding an approximate upper bound on the throughput performance of a node using the AGENT protocol:

$$T_{\text{node}} = \frac{\varphi}{N} + \left(1-\frac{1}{\eta}\right)^{\eta-1}. \tag{6}$$

By substituting (1) and (2) for η, we obtain the respective pure unicast and pure broadcast average throughput performance.

4.2. Reliability analysis

There are a few scenarios in which all of the destinations do not receive a multi-point packet transmission. These scenarios are not unique to AGENT as similar situations arise in other MAC protocols [13,15]. The source of the problem is the combination of half-duplex communication and the lack of a positive control response for multi-point transmissions.

Since AGENT guarantees each node collision-free transmission in its assigned slot, these failure scenarios arise only when nodes contend in an unassigned slot. Referring back to figure 2, assume that nodes 3 and 4 simultaneously send a RTS for broadcast in the contention interval. Since nodes 3 and 4 are transmitting, they cannot directly detect the collision. Furthermore, nodes 1, 2 and 5 each receive a RTS, and therefore do not respond. Consequently, both nodes 3 and 4 incorrectly conclude that there is no contention for the unassigned slot, and broadcast their packets in the transmission interval. Clearly, neither 3 nor 4 receives the other's broadcast packet.

AGENT fails in the above scenario because there is no "witness" to the RTS collision. To calculate the likelihood of such a failure, we compute the probability that there are no witnesses of a RTS collision between two neighboring nodes i and j.

Let I be the area of intersection between the transmission radii of i and j when $\mathrm{dist}(i, j) = 2/(3r)$. Let γ represent the probability that a node k resides in I. Using the circle intersection formulas, we find that $\mathrm{E}[\gamma] \approx 2.24r^2/A$. Let $P(\kappa)$ be the probability that there are exactly κ contending nodes in I:

$$P(\kappa) = \binom{N-2}{\kappa} (\gamma\tau)^{\kappa} (1-\gamma)^{N-2-\kappa}.$$

Then AGENT fails when there are $0, 1, 2, \ldots,$ or $N - 2$ such contending nodes in I. Therefore, the probability that a broadcast fails to be received by all neighbors, P_{fail}, is

$$P_{\mathrm{fail}} < \mathrm{e}^{\gamma(2-N)} \sum_{\kappa=1}^{N-2} \frac{1}{\kappa!} \bigl((N-2)\gamma\tau\bigr)^{\kappa}. \tag{7}$$

Notice that as the number of nodes N is increased, P_{fail} decreases rapidly. In [6], our analysis has shown that P_{fail} remains less than 4%.

5. Performance evaluation

The primary goal of our simulation experiments is to collect baseline performance measures of AGENT. We also compare these results with that of other existing MAC protocols within the same simulation framework. Due to limited space, we restrict our evaluation to include the CATA protocol [13], chosen for its similar design and features.

5.1. Simulation model

Using a discrete event simulator, we model an ad hoc network consisting of 100 mobile nodes operating in a two-dimensional plane that measures 10 km per side. Each node is equipped with a simulated radio device that transmitted at a rate of 1 Mbps to a distance of 1 km. All communication occurs on a single perfect channel, i.e., no channel noise, with a free-space propagation model.

Node movement is simulated using the mobility model defined in [11]. With this model, a node's movement is characterized by three parameters: $\langle \overline{\lambda}, \overline{\mu}, \overline{\sigma}^2 \rangle$, where $\overline{\lambda}$ is the average time traveled in a single direction with a constant speed; $\overline{\mu}$ is the average speed; and $\overline{\sigma}^2$ is the speed variance. In our simulations we use $\langle 60^{-1}$ s, 5 m/s, 8.33 m/s$\rangle$ to correspond with pedestrian movement characteristics.

Network traffic is introduced according to a Poisson arrival process with a mean arrival rate of λ packets per second, which are uniformly distributed among the nodes. Each packet is 512 bytes in length, and is either a unicast or broadcast packet. This represents a worst-case traffic scenario since the interaction between these two traffic types is the most volatile. The control packets are 32 bytes in length. We arbitrarily select two different traffic scenarios – one consisting of a 80% unicast and 20% broadcast (80/20) traffic mixture, and the other consisting of a 60% unicast and 40% broadcast (60/40) traffic mixture.

Since we are only interested in MAC layer performance, no specific transport, network, or data link protocols are introduced. To accommodate packet addressing, we assume that each node has perfect knowledge of its neighbors.

5.2. CATA implementation

CATA allows nodes to compete for synchronous time slots, and supports point-to-point and multi-point packet transmissions. It also features slot reservations that maintains collision-free access for extended periods of time. Since multi-point communication is the focus of this paper, this feature was disabled. The contention and reservation scheme is based on a RTS/CTS handshake, and slots are organized into a synchronous frame. We use a static frame length equal to the number of nodes. Each time slot is subdivided into five *mini-slots*. The first four mini-slots (CMS1–CMS4) are used to secure and reserve time slots through the exchange of short control packets. The last mini-slot (DMS) is used for the transmission of a data packet. Unlike AGENT, nodes do not have a dedicated slot.

For a given source node s and time slot t, CATA operates as follows. Regardless of the packet type, s must first determine whether or not the current slot has been previously reserved. To reserve a slot, all nodes that received data in slot t in the preceding frame send a slot reservation (SR) packet in CMS1. In addition, each source node that wishes to maintain a reservation sends a RTS and not-to-send (NTS) packets in CMS2 and CMS4, respectively.

If no SR packet is detected in CMS1, then source node s contends for slot t by sending its own RTS in CMS2. Reception of a unicast RTS causes a node to respond with a CTS in CMS3, and s can transmit its data packet in the DMS. Reception of a multicast or broadcast RTS in CMS2 causes a node to remain silent during CMS3 and CMS4; otherwise it sends a NTS in CMS4 to indicate a potential problem for multi-point transmissions. Detection of a clear channel in CMS4 allows source node s to transmit a multi-point packet in the DMS. Any unsuccessful slot contention is handled by a backoff algorithm; since no specific algorithm was specified in [13], we use a binary exponential backoff algorithm.

5.3. Simulation results

In this set of experiments, we measure how effectively each protocol utilizes the channel capacity and the average access delay for both unicast and broadcast packets. The results are presented in figures 4–15. Each data point represents the statistical average of several simulation trials, and lies within a

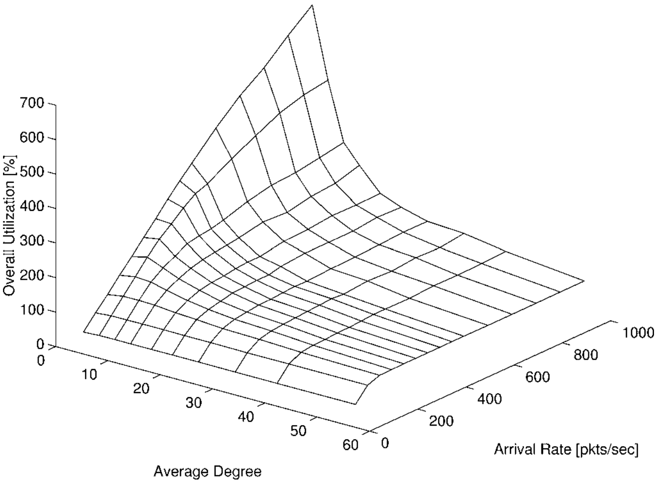

Figure 4. AGENT utilization (80/20).

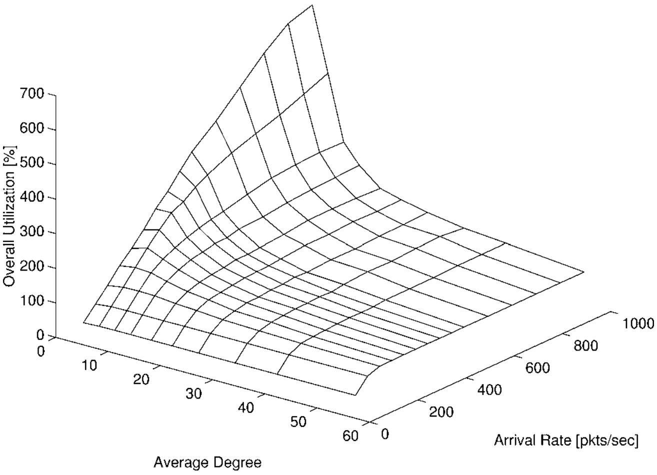

Figure 5. AGENT utilization (60/40).

90% confidence interval. Each graph shows the associated performance metric as a function of the average node degree (η'), and traffic arrival rate (λ), measured in packets per second.

5.3.1. *Channel utilization*

Figures 4 and 5 depict the channel utilization of AGENT with a 80/20 and 60/40 traffic mixture, respectively. The potential for concurrent transmission (i.e., spatial bandwidth reuse) is inversely proportional to the network connectivity. When the average node degree is less than 20, we find that the channel utilization of AGENT exceeds the channel capacity, reaching a maximum of 700% (i.e., an average of seven concurrent packet transmissions are occurring in each slot). This demonstrates that the contention-based component protocol is capable of spatially reusing TDMA slots, resulting in increased bandwidth efficiency. As network connectivity is increased, the contention level for each slot rises, and channel utilization begins to drop. However, the underlying TDMA protocol prevents instability under high traffic loads and network connectivity. Thus, AGENT operates at near channel capacity under such conditions.

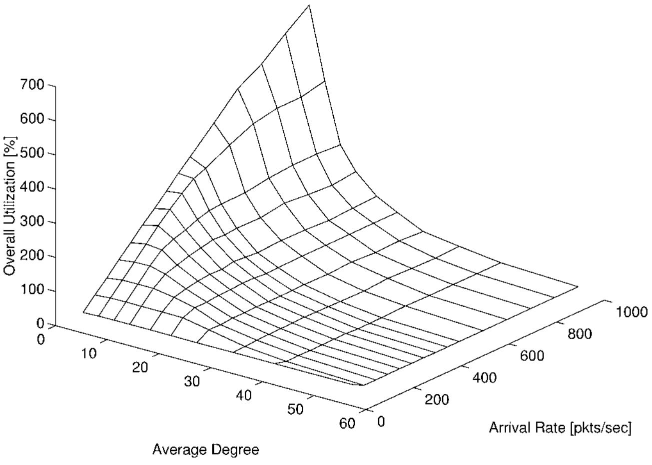

Figure 6. CATA utilization (80/20).

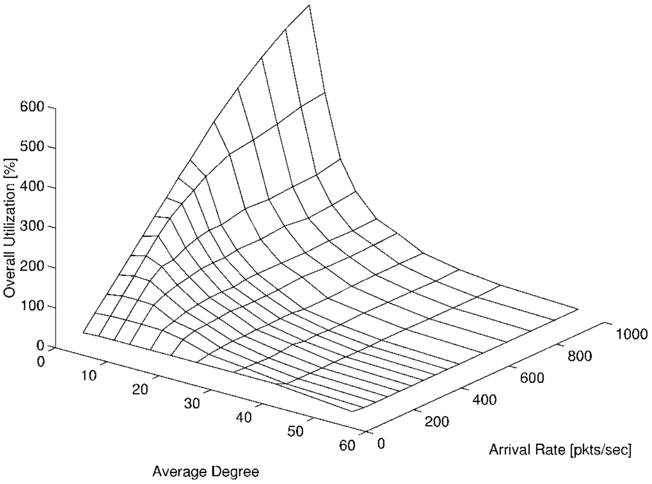

Figure 7. CATA utilization (60/40).

Comparing figures 4 and 5 we find that there is a slight degradation in channel utilization when the proportion of broadcast traffic is increased. This is expected since the number of nodes involved (i.e., remaining silent) in a broadcast is greater than in a unicast transmission. However, the lack of significant performance degradation indicates the absence of bias among unicast and broadcast packet transmissions. This fact is emphasized when we examine the associated access delays.

Figures 6 and 7 depict the channel utilization of CATA with a 80/20 and 60/40 traffic mixture, respectively. In figure 6, we find the utilization of CATA comparable to AGENT when the network connectivity is sparse ($\eta' \leqslant 10$). As before, the contention protocol is successful in spatially reusing the available time slots, leading to a channel utilization that exceeds capacity. However, the absence of permanent slot assignments forces all channel access to be decided through contention. Rising contention levels naturally increase the number of unsuccessful slot contentions that must be resolved by the backoff algorithm. This introduces additional packet delay, and reduces the number of packets sent in each slot. Consequently, the channel utilization of CATA begins

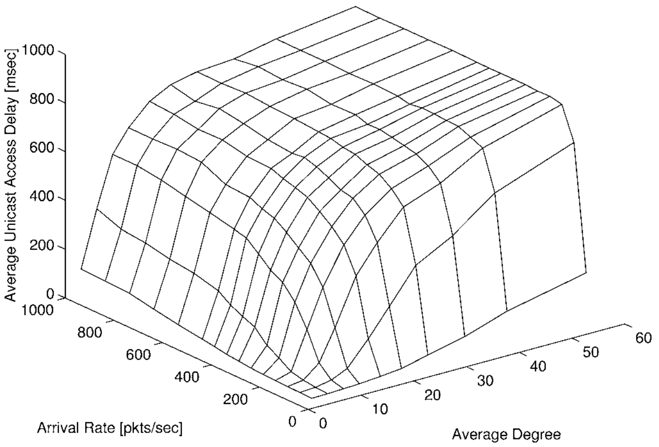

Figure 8. Average unicast access delay of AGENT (80/20).

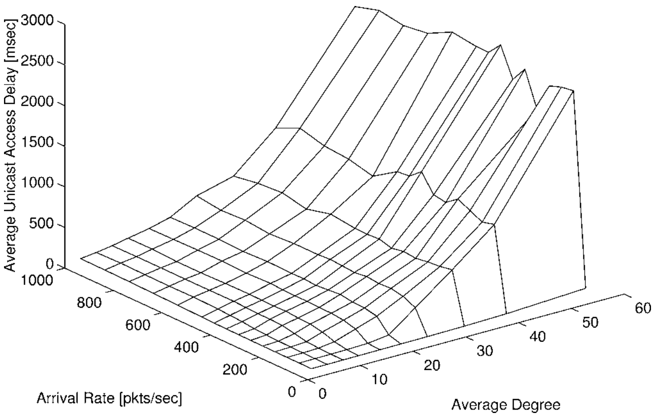

Figure 10. Average unicast access delay of CATA (80/20).

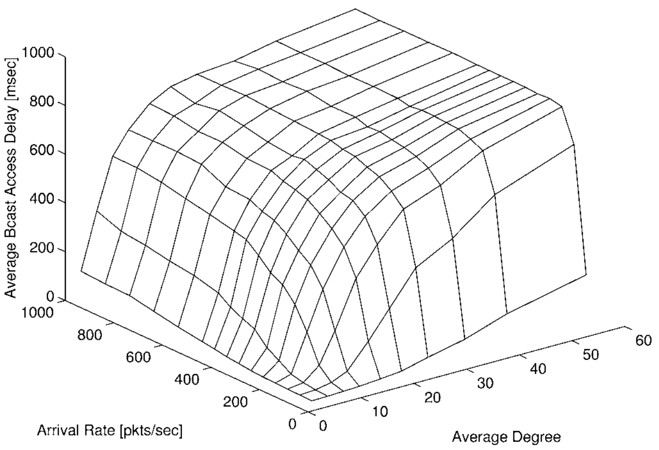

Figure 9. Average broadcast access delay of AGENT (80/20).

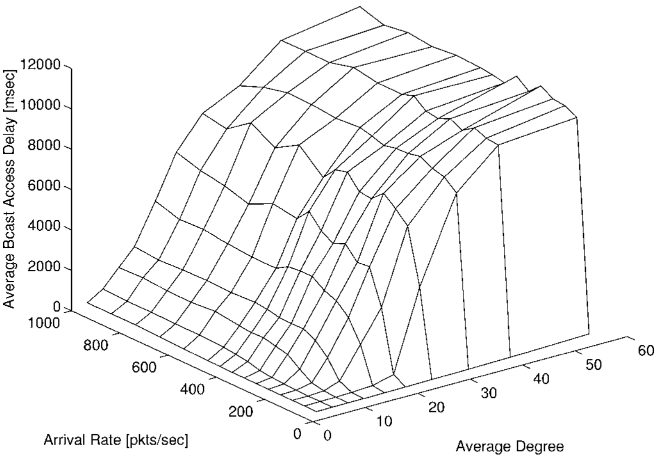

Figure 11. Average broadcast access delay of CATA (80/20).

to quickly deteriorate as η' is increased, and eventually drops well below full capacity.

Comparing figures 6 and 7 we see that there is a more pronounced performance degradation when the proportion of broadcast traffic is increased. The maximum utilization is 700% in figure 6, and reduced to 600% in figure 7. Examining the protocol, we find that the responses for unicast and broadcast RTS control packets are sent at different times. It is then possible to receive a positive unicast response followed by a negative broadcast response. For example, referring back to the network in figure 2, let node 1 transmit a broadcast RTS and node 3 a unicast RTS to 4 in CMS2. Then 3 will receive a CTS in CMS3 while 1 receives a NTS in CMS4. In this case, the unicast succeeds and the broadcast fails. This preferential treatment of unicast increases the delay of broadcast packets. Consequently, increasing the amount of broadcast traffic reduces the overall channel utilization.

5.3.2. Access delay

Figures 8 and 9 depict the respective average unicast and broadcast access delay of AGENT with a 80/20 traffic mixture, from which we see no significant difference in the delay experienced by unicast and broadcast packets. This confirms that AGENT has no bias towards either packet type. Furthermore, the maximum delay associated with either packet type remains asymptotically bounded by 1000 ms, which corresponds to the frame length used. The same results are evident in figures 12 and 13 which show the respective average unicast and broadcast access delay of AGENT with a 60/40 traffic mixture.

Figures 10 and 11 depict the respective average unicast and broadcast access delay of CATA with a 80/20 traffic mixture. We see that the average broadcast access delay is up to 4 times higher than its unicast delay, clear evidence of CATA's partiality towards unicast transmissions. As the proportion of broadcast traffic is increased (see figures 14 and 15) we find the unicast delay relatively unchanged, while the broadcast delay actually decreases. With a 60/40 traffic mixture, there is less opportunity to favor unicast packets, and thus broadcast packets are successfully transmitted with increased frequency. The irregularity present in the CATA access delay curves indicate a high degree of delay variation. Large delay variations typically lead to increased packet jitter which negatively impact the performance quality of multimedia applications.

In comparison to AGENT, there is a significant increase in the average access delay of CATA for both packet types. With unicast traffic, the average access delay of CATA ranges from a few milliseconds to nearly 3 s. This wide delay range limits the ability of high level services to estimate link/path qual-

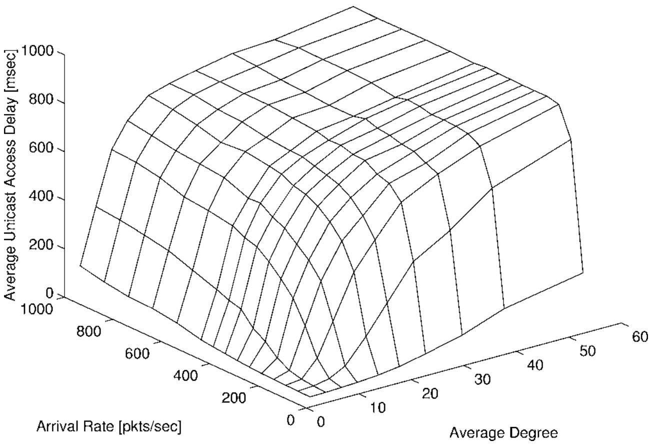

Figure 12. Average unicast access delay of AGENT (60/40).

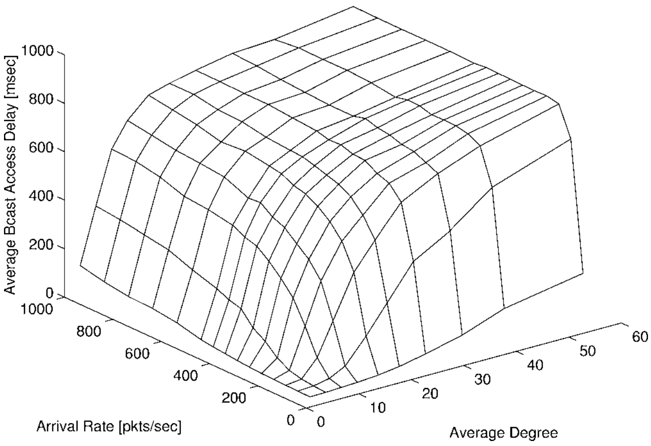

Figure 13. Average broadcast access delay of AGENT (60/40).

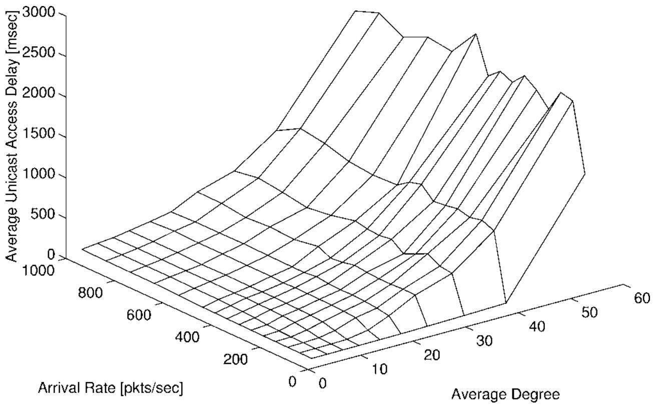

Figure 14. Average unicast access delay of CATA (60/40).

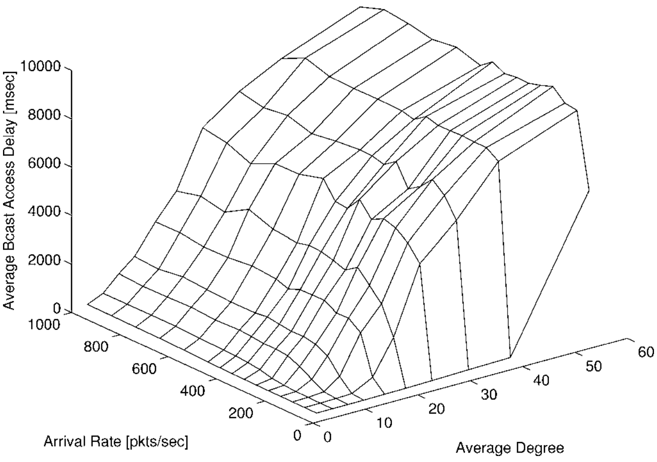

Figure 15. Average broadcast access delay of CATA (60/40).

ity. With broadcast traffic, the delay range is even wider with a maximum exceeding 10 s. This can also negatively impact the performance of higher layer services. For example, proactive routing protocols typically use broadcast transmissions to periodically exchange updated connectivity information. The introduction of large broadcast delays reduces the frequency of these updates, diminishing the effectiveness of the routing protocol to find valid paths.

With AGENT, the access delay of both unicast and broadcast traffic is bounded. Consequently, worst case end-to-end delay can easily be computed for unicast traffic, and the use of broadcast for periodic information exchange is much less volatile. However, the use of an allocation protocol introduces synchronization and timing issues that were not studied in this performance evaluation. Many of these timing issues can be alleviated through the application of global timing services, such as the Global Positioning System (GPS). This requires additional hardware to be added to each node, resulting in an increase per unit cost and reduced battery life.

6. Conclusions

This paper presented AGENT, an adaptive, generalized transmission protocol for ad hoc networks that offers a unified set of effective single-hop transmission services. AGENT features a hybrid design that combines a TDMA allocation protocol and a contention protocol that employs a collision-avoidance dialogue. The allocation component provides access delay bounds, while the contention component increases spatial bandwidth reuse. We provided an approximate analytical framework in which we examined the performance of AGENT with respect to point-to-point and multi-point traffic. We also analyzed the reliability of AGENT, and showed that the probability of multi-point failure is very low. Finally, we evaluated the performance of AGENT in a simulated ad hoc environment. Our results illustrate the effective operation of a hybrid design, and show significant improvement over similar MAC designs.

Although the current hybrid design of the AGENT protocol performs well, its application is somewhat limited. The use of TDMA means that the delay bound is directly proportional to the network size. For larger networks consisting of thousands of nodes, the current AGENT protocol may no longer be a feasible alternative. Moreover, the network size is typically unknown and time varying. Our future research efforts will focus on overcoming these limitations through the use of other, more scalable, protocol combinations.

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful comments that greatly improved the quality of this paper.

References

[1] S. Basagni, I. Chlamtac and V.R. Syrotiuk, Geographic messaging in wireless ad hoc networks, in: *Proc. IEEE VTC'99*, Vol. 3 (May 1999) pp. 1957–1961.

[2] D. Bertsekas and R. Gallager, *Data Networks* (Prentice Hall, 1992).

[3] C. Chiang, M. Gerla and L. Zhang, Adaptive shared tree multicast in mobile wireless networks, in: *Proc. IEEE GLOBECOM'98*, Vol. 3 (November 1998) pp. 1817–1822.

[4] I. Chlamtac and A. Faragó, Making transmission schedules immune to topology changes in multi-hop packet radio networks, IEEE/ACM Transactions on Networking 2(1) (February 1994) 23–29.

[5] I. Chlamtac, A. Faragó, A.D. Myers, V.R. Syrotiuk and G. Záruba, ADAPT: A dynamically self-adjusting media access control protocol for ad hoc networks, in: *Proc. IEEE GLOBECOM'99*, Vol. 1a (December 1999) pp. 11–15.

[6] I. Chlamtac, A.D. Myers, V.R. Syrotiuk and G. Záruba, An adaptive media access control (MAC) protocol for reliable broadcast in wireless networks, in: *Proc. IEEE ICC'00* (June 2000).

[7] C. Diot, W. Dubbous and J. Crowcroft, Multipoint communication: A survey of protocols, functions and mechanisms, IEEE Journal on Selected Areas in Communications 15(3) (April 1997) 277–290.

[8] J.J. Garcia-Luna-Aceves and E. Madruga, A multicast routing protocol for ad-hoc networks, in: *Proc. IEEE INFOCOM'99*, Vol. 2 (March 1999) pp. 784–792.

[9] D. Jeong and W. Jeon, Performance of an exponential backoff scheme for the slotted-ALOHA protocol in local wireless environment, IEEE Transactions on Vehicular Technology 44(3) (August 1995) 470–479.

[10] P. Karn, MACA – A new channel access protocol for packet radio, in: *Proc. ARRL/CRRL Amateur Radio 9th Computer Networking Conference* (September 1990).

[11] B. McDonald and T. Znati, A path availability model for wireless ad-hoc networks, in: *Proc. IEEE WCNC'99*, Vol. 1 (September 1999) pp. 35–40.

[12] B.A. Sharp, E.A. Grindrond and D.A. Camm, Hybrid TDMA/CSMA protocol for self managing packet radio networks, in: *Proc. IEEE ICUPC'95* (November 1995) pp. 929–933.

[13] Z. Tang and J.J. Garcia-Luna-Aceves, A protocol for topology-dependent transmission scheduling in wireless networks, in: *Proc. IEEE WCNC'99*, Vol. 3 (September 1999) pp. 1333–1337.

[14] F. Tobagi and L. Kleinrock, Packet switching in radio channels. I. Carrier sense multiple access models and their throughput delay characteristics, IEEE Transactions on Communications COM-23(12) (December 1975) 1400–1416.

[15] C. Zhu and S. Corson, A five-phase reservation protocol (FPRP) for mobile ad hoc networks, in: *Proc. INFOCOM'98*, Vol. 1 (March 1998) pp. 322–331.

[16] *DARPA/ITO Global Mobile Information Systems Principle Investigator Meeting*, Richardson, TX (February 1999).

[17] Wireless Medium Access Control and Physical Layer WG, IEEE Draft Standard P802.11, Wireless LAN, IEEE Standards Department, D3 (January 1996).

Andrew D. Myers received the B.S. (with honors) and M.S. degrees in computer science from the Erik Jonsson School of Engineering and Computer Science at the University of Texas at Dallas (UTD) in 1997 and 1999, respectively. In 1997, he joined the Center for Advanced Telecommunications Systems and Services where he developed and tested signaling protocols for the DARPA sponsored GloMo Project. He is currently pursuing a Ph.D. degree in computer science at UTD, and is a senior Research Assistant at the Center. Current interests include communication protocols, routing algorithms, and applications for ad hoc and sensor networks.
E-mail: amyers@utdallas.edu

Gergely V. Záruba received the M.S. degree in computer engineering in 1997 from the Technical University of Budapest, Department of Telecommunications and Telematics, with excellent classification. Since 1997 he is pursuing research towards a Ph.D. degree. From 1995 to 1998 he was a member of Hungary's leading telecom research laboratory, the High Speed Networks Laboratory at the Technical University of Budapest. In 1998 he joined the Center for Advanced Telecommunications Systems and Services (CATSS) at the University of Texas at Dallas, where he is currently pursuing research on communication protocols for wireless networks as a Research Assistant. His research interests include MAC protocol issues in ad hoc networks, admission control in wireless networks, and Bluetooth scatternet formation problems.
E-mail: zaruba@utdallas.edu

Violet R. Syrotiuk received the Ph.D. degree in computer science from the University of Waterloo (Canada) in 1992. She joined the University of Texas at Dallas in 1994 and is currently an Assistant Professor of Computer Science. In 1997, she joined the Center for Advanced Telecommunications Systems and Services where she was co-PI and Technical Manager of a DARPA sponsored GloMo Project. She served on the organizing committee of the First Annual MobiHoc Workshop held in August 2000. Dr. Syrotiuk's current research interests include medium access and network layer protocols for wireless mobile multi-hop networks, network simulation, and distributed algorithms and systems.
E-mail: syrotiuk@utdallas.edu