Contention-Aware Admission Control for Ad Hoc Networks

Yaling Yang, *Student Member*, *IEEE*, and Robin Kravets, *Member*, *IEEE*

Abstract—An admission control algorithm must coordinate between flows to provide guarantees about how the medium is shared. In wired networks, nodes can monitor the medium to see how much bandwidth is being used. However, in ad hoc networks, communication from one node may consume the bandwidth of neighboring nodes. Therefore, the bandwidth consumption of flows and the available resources to a node are not local concepts, but related to the neighboring nodes in carrier-sensing range. Current solutions do not address how to perform admission control in such an environment so that the admitted flows in the network do not exceed network capacity. In this paper, we present a scalable and efficient admission control framework—**C**ontention-aware **A**dmission **C**ontrol **P**rotocol (CACP)—to support QoS in ad hoc networks. We present several options for the design of CACP and compare the performance of these options using both mathematical analysis and simulation results. We also demonstrate the effectiveness of CACP compared to existing approaches through extensive simulations.

Index Terms—Admission control, ad hoc network, multihop, QoS routing, Quality of Service, contention-aware, simulations.

1 Introduction

The expanded availability of small wireless computers has enabled the design and deployment of mobile ad hoc networks. Many suggested applications for these networks include multimedia data that require quality of service (QoS) support for effective communication. While many solutions for QoS support have been proposed in wired networks, the differences between wired and wireless communication demand the design of new solutions for ad hoc networks.

Since the nodes in an ad hoc network must cooperate with each other to provide routing, they must also cooperate with each other to provide QoS support. Such cooperation includes policing at the endpoints of the flows and admission control along the routes to prevent new flows from consuming too many resources and disrupting the guarantees made to existing flows. Assuming the support of well-policed flows, the goal of our research is to provide an effective, scalable admission control protocol for wireless ad hoc networks so that end-to-end connections with QoS requirements can be maintained.

The goal of any QoS support is to provide applications with guarantees in terms of bandwidth, delay, or jitter. To provide such guarantees in a networked environment, the MAC layer is responsible for resource allocation at individual nodes, while the network layer must consider resources along the entire route of communication. While many approaches have dealt with such QoS support in wired networks or multichannel wireless networks, such as TDMA or CDMA, the physical characteristics of single-channel wireless networks, such as IEEE 802.11 networks [1], do not lend themselves well to such guarantees. The main problem stems from the shared nature of the wireless medium in single-channel networks. Essentially, nodes that cannot communicate with each other directly may still contend directly with each other for the same resources. This extended contention area, called the *c-neighborhood* in this paper, affects resource allocation at individual nodes in two ways. First, allocation decisions at an individual node require bandwidth information of nodes outside of its communication range and along the entire route. Second, contention for resources may involve multiple nodes along a route. Our research provides admission control based on knowledge of these characteristics of wireless communication.

In this paper, we present CACP (Contention-aware Admission Control Protocol), which provides admission control for flows in a single-channel ad hoc network based on knowledge of both local resources at a node and the effect of admitting the new flow on neighboring nodes. We focus on ad hoc networks based on single-channel MAC layers like IEEE 802.11 because these single-channel protocols are widely available and typically support ad hoc communication. Additionally, such protocols are simple and robust and do not rely on tight time synchronization or code/slot assignment algorithms that are hard to implement in ad hoc networks. Throughout this paper, IEEE 802.11 is used for the description and analysis of CACP, although CACP can be combined with other single channel MAC protocols such as IEEE 802.11e [2] and SEEDEX [3].

Currently, CACP focuses on QoS support in terms of bandwidth allocation. However, simulations show that by controlling bandwidth allocation, delay and jitter can also be controlled. These characteristics of CACP come from our novel approach to admission control, which is not limited to nodes in communication range but extended to all nodes in contention range (i.e., *c-neighbors* throughout this paper). We demonstrate through mathematical analysis and simulations that CACP is efficient in terms of overhead, while maintaining effective use of available communication resources.

• *The authors are with the Department of Computer Science, University of Illinois at Urbana-Champaign, 201 N. Goodwin, Urbana, IL 61801. E-mail: {yyang8, rhk}@cs.uiuc.edu.*

Manuscript received 17 May 2004; revised 23 Oct. 2004; accepted 25 Oct. 2004; published online 27 May 2005.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0166-0504.

The rest of the paper is organized as follows: Section 2 presents the characteristics of wireless communication relevant to admission control in ad hoc networks. Section 3 discusses the challenges and solutions of providing admission control in ad hoc networks. Section 4 describes the CACP protocol design in detail. Section 5 analyzes the overhead of admission control in c-neighborhoods. Section 6 compares simulation results of CACP to SWAN [4] and DSR [5]. Finally, Section 7 concludes our work and discusses future extensions.

2 Characteristics of Wireless Communication

Allocation of communication resources is always necessary for communication over a shared medium. In a wired network, communication resources are well-defined. A node knows that whatever resources it uses will be directly removed from the pool of resources available to other nodes sharing the medium. Additionally, all nodes see the same amount of physical resources available to the pool. In a wireless network, this is no longer true. Essentially, the characteristics of the shared wireless medium do not provide a unified view of the medium to all nodes due to the physical differences between wired and wireless communication as described in detail in the following section. The problem is increased when considered in the context of an ad hoc network with dynamically moving mobile nodes. In this section, we discuss these challenges and describe some of the current research in the area.

2.1 Wireless Channels

The first difference between wired and wireless networks lies in the openness of wireless communication channels. Whether point-to-point or shared, wired links can be isolated to ensure that only authorized devices can use the communication channel. Wireless links are inherently shared and no such isolation is possible. Any node with a wireless transmitter can simply send data and contend for the wireless channel. There is also no isolation from other sources that may be using the channel with a completely different infrastructure (e.g., IEEE 802.11 and Bluetooth) or simply causing noise (e.g., a microwave oven). While we certainly make no claim to provide a solution for such interference, our approach dynamically monitors the communication medium to base resource allocation and admission control decisions on knowledge of such interference and the current state of the wireless channel.

The second difference also stems from the physical characteristics of wireless communication. The structures of point-to-point and shared medium wired networks share the characteristic that nodes can hear each other's transmission and see the same channel state. However, for a wireless ad hoc network, the channel no longer maps to a physical medium, but instead maps to a physical space. Therefore, each node has a different view of the state of the communication channel.

This unique structure of wireless channels introduces the two challenges listed below:

- *Prediction of Available Bandwidth*. In shared medium wireless networks, when a node starts to transmit a flow, it consumes bandwidth at its c-neighbors. Because each node has a different view of the network, the node cannot decide on its own whether its c-neighbors have enough free bandwidth for the new flow. In addition, obtaining c-neighbor information is not trivial since a node may consume the bandwidth of a c-neighbor but not be able to directly communicate with that c-neighbor if the c-neighbor is located outside transmission range and inside carrier-sensing range.
- *Prediction of Flow Bandwidth Consumption*. Because multiple nodes on a route may contend for bandwidth at a single location and not know about each other, a node on the route of a flow cannot tell how much bandwidth the flow will consume at its c-neighbors.

To address these two challenges, new mechanisms, which may require additional hardware support or introduce extra message overhead, must be used. In Section 3, we discuss possible solutions to these challenges in detail.

2.2 Mobility

While it may be possible to find a feasible route for a flow, strict QoS, as in wired networks, cannot be guaranteed in an ad hoc network when mobility is present because mobility can break routes frequently. In addition, there is no guarantee that resources will remain available since available bandwidth may decrease when communicating nodes move into range of each other. Therefore, we support the idea introduced in [6] that states that QoS requirements in ad hoc networks should be relaxed to allow a better-than-best-effort service. The QoS commitment that CACP provides is that no node intentionally breaks the QoS commitment to the flows in the network by admitting too many flows. However, when the commitment is broken due to mobility, a notification message is sent to the source indicating changes in the route. The source can either search for a new route or reduce its QoS requirement to accommodate a broken or degraded route. Since the wireless medium is not isolated, the management of communication resources depends on knowledge of the communication in the network and the network topology. However, the limited bandwidth of wireless ad hoc networks requires the limitation of any message overhead from information collection. In addition, due to mobility, information gathered about the network only has a limited lifetime. Therefore, it is best to collect information as close as possible to the time and location that it is needed. Hence, CACP coordinates admission control in an on-demand fashion and so ties message overhead to the presence of requesting flows.

2.3 Existing Approaches

As discussed, the differences and challenges in wireless mobile ad hoc networks demand a very different perspective on network QoS management. In current research, TDMA-based approaches are proposed to provide QoS in wireless ad hoc networks [7], [8], [9], [10], [11]. However, such approaches require effective synchronization between all nodes in the network. Applying highly synchronized solutions in an ad hoc network becomes expensive and

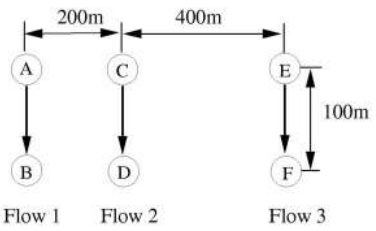

Fig. 1. Simulation topology.

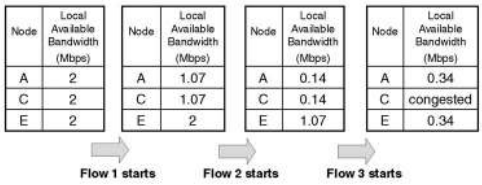

Fig. 2. Changes of local available bandwidth.

synchronization can fail when the nodes are mobile. The slot allocation algorithm in TDMA schemes is also vulnerable to mobility in the network since slot allocations must be reconfigured whenever there are changes in available bandwidth or changes to routes in the network.

As a result, single channel MAC layer scheduling approaches have been suggested to provide resource allocation in ad hoc networks [3], [12], [13]. In these approaches, only one channel is shared between all nodes. QoS is realized by coordinating the transmission schedules of packets between nodes. Although these approaches are based on localized decisions and, hence, are more flexible in the presence of mobility, they all focus on the packet level and only deal with fair resource allocation at individual nodes. To support QoS guarantees for end-to-end flows, these approaches need to be combined with admission control, which supports end-to-end resource allocation along the route of a flow.

Recently, some solutions related to admission control in ad hoc networks have been proposed [4], [6], [14], [15], [16], [17], [18], [19]. However, work such as INSIGNIA [14], MMWN [16], and connectionless routing architecture [17] focus on high level issues and do not provide solutions for bandwidth allocation in the presence of contention between c-neighbors. Others, like SWAN [4] and VMAC [6], [15], [18], [19], do not give enough attention to the fact that, when making admission control decisions, a node must not only consider local resources but also consider the resources of its c-neighbors since it may consume their resources through contention. Our work fills this hole by allowing CACP to consider both local resources and resources at c-neighbors when making admission control decisions. The challenge to CACP lies in the fact that information about contention cannot as easily be obtained as in a wired network. Since each node can only observe the amount of its local resources, new mechanisms must be used to collect c-neighbor resource information while imposing minimal additional overhead and contention on the network. Our research addresses these challenges with the goal of providing an effective, scalable admission control protocol for wireless ad hoc networks so that end-to-end connections with QoS requirements can be established.

3 Contention-Aware Admission Control

The aim of admission control is to determine whether the available resources can meet the requirements of a new flow while maintaining bandwidth levels for existing flows. As discussed in Section 2.1, due to the shared nature of the wireless channel, the challenges of achieving this goal in ad hoc networks include predicting the available bandwidth and predicting the bandwidth consumption of a flow. In this section, we discuss these challenges and their solutions.

3.1 Prediction of Available Bandwidth

The first challenge to CACP is evaluating the available bandwidth in the network so that the bandwidth requirements of all the flows do not exceed the resources in the network. Since each node sees a different channel state, the available bandwidth in the network is not simply a local concept. To capture this complexity, we define two terms: *c-neighborhood available bandwidth* and *local available bandwidth*. C-neighborhood available bandwidth is the maximum amount of bandwidth that a node can use for transmitting without depriving the reserved bandwidth of any existing flows in its carrier-sensing range (c-neighborhood). Local available bandwidth is the amount of unconsumed bandwidth as observed by a given node. Since a node can consume the bandwidth of nodes that are in its c-neighborhood, the c-neighborhood available bandwidth for a given node is equal to the smallest local available bandwidth of all of its c-neighbors. Therefore, to successfully admit a flow, a node must have enough local and c-neighborhood available bandwidth.

To demonstrate this relationship, we show a simple simulation using NS2 [20]. In this scenario, there are six mobile hosts positioned as in Fig. 1. The MAC layer protocol is IEEE 802.11 with 250m radio transmission range and 550m carrier-sensing range. The bandwidth of the wireless channel is 2Mbps. Node C and Node E are each other's c-neighbors. Node A is Node C's neighbor and is out of Node E's carrier-sensing range (c-neighborhood). Three 133 packets per second CBR flows with packet size of 512 Bytes are established between Nodes A and B, Nodes C and D, and Nodes E and F. Due to the overhead of the MAC layer RTS-CTS-DATA-ACK handshake and collisions, each flow requires about 930Kbps channel bandwidth. At time 1 second, Node A initiates Flow 1 to Node B. At 40 seconds, Node C initiates Flow 2 to Node D. Finally, at 80 seconds, Node E initiates Flow 3 to Node F. Fig. 2 shows the changes in local available bandwidth at each source node as the three flows start successively. Fig. 3 and Fig. 4 show the throughput and delay of each flow over time.

As shown in Fig. 2, after Flow 2 starts, Node E has 1.07 Mbps bandwidth that is not consumed by the contention from Flow 2. Therefore, to Node E, there is 1.07 Mbps local available bandwidth, which is enough to admit Flow 3. Therefore, when Flow 3 starts, it can get its desired throughput and delay (See Fig. 3 and Fig. 4). In previous admission control approaches [6], [4], [15], [18], [19], since no consideration is given to c-neighborhood available bandwidth, Node E would admit Flow 3. However, since Node E is Node C's c-neighbor, Flow 3

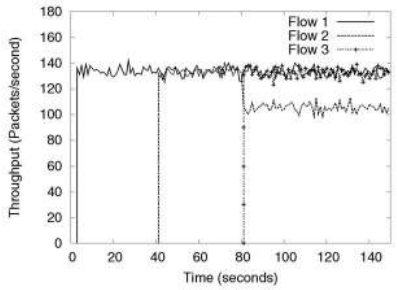

Fig. 3. Throughput of Flow 1, 2, and 3.

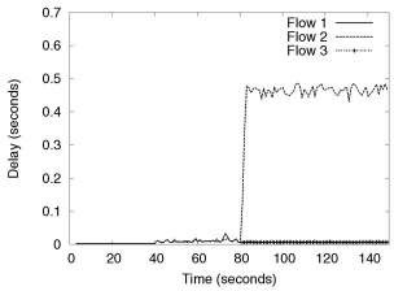

Fig. 4. Delay of Flow 1, 2, and 3.

consumes Node C's local available bandwidth. When Node E starts Flow 3, the contention from Flow 3 actually decreases the throughput of Flow 2 by 20 percent and increases the delay of Flow 2 dramatically (See Fig. 3 and Fig. 4). The reason for the decrease of QoS to Flow 2 is that Node C only has 0.14 Mbps local available bandwidth before Flow 3 starts, which is much less than Flow 3's bandwidth consumption. In other words, Node E does not have enough c-neighborhood available bandwidth for Flow 3. Therefore, enough local available bandwidth is not sufficient to accept a flow. There must also be enough c-neighborhood available bandwidth.

3.1.1 Calculation of Local Available Bandwidth

Since local available bandwidth is defined as the unconsumed bandwidth at a given node, each node can determine its own local available bandwidth by passively monitoring network activities. In this paper, we use the fraction of idle channel time during the past history as an indication of local available bandwidth at a node. It may seem that this approach is too optimistic since it does not consider that some of the channel time cannot be used due to idle time caused by the backoff algorithm in IEEE 802.11 and the collisions in the network. However, in [21], it is demonstrated that, for IEEE 802.11, the amount of unusable idle time and collision time is negligible compared to the packet transmission time. Therefore, using the fraction of idle channel time can be a simple approximation for local available bandwidth. An alternative approach used in [18], [19] predicts local available resources from the reciprocal of the current transmission delay, which essentially reflects the local achievable bandwidth. However, local achievable bandwidth is the maximum amount of bandwidth that a flow can achieve by competing with existing flows and potentially reducing the throughput of existing flows, which is not appropriate for the purpose of admission control since admission control should maintain the throughput of existing flows. Therefore, CACP only uses the idle channel time to estimate local available bandwidth.

In general, the channel at a node can be perceived as either idle or busy. The channel is idle if the node is **NOT** in the following three states. First, the node is transmitting or receiving a packet. Second, the node receives an RTS or CTS message from another node, which reserves the channel for a period of time specified in the message. Third, the node senses a busy carrier with signal strength larger than a certain threshold, called the *Carrier-sensing Threshold*, but the node cannot interpret the contents of the message. By monitoring the amount of idle channel time, T_{idle}, during every period of time, T_p, the local available bandwidth, B_{local}, for a node can be estimated using a weighted moving average as follows:

$$B_{local} = \alpha B_{local} + (1-\alpha)\frac{T_{idle}}{T_p} B_{channel}, \quad (1)$$

where $B_{channel}$ is the channel capacity and weight $\alpha \in [0,1]$.

In the context of a MAC layer that supports priority-based scheduling, such as IEEE 802.11e [2], the estimation of local available bandwidth may be more complex since it may be desirable to allow high priority flows to deprive the bandwidth of admitted low priority flows. In such a network, the local available bandwidth to a flow may depend on its priority. A method of calculating the local available bandwidth for each priority is presented in our work in [22] and can easily be combined with CACP.

3.1.2 Prediction of c-Neighborhood Available Bandwidth

Since each node has a different view of the network, knowledge of a node's own local available bandwidth cannot provide information about its c-neighborhood available bandwidth since it does not know the amount of local available bandwidth at other nodes. To obtain bandwidth information at c-neighboring nodes, two kinds of approaches can be used: active approaches and passive approaches. In active approaches, c-neighbors actively exchange bandwidth information between each other. Since c-neighbors may not be able to directly communicate with each other, such exchanges may impose relatively high message overhead. In passive approaches, a node passively monitors the channel to estimate its c-neighbors' local available bandwidth. While the message overhead of passive approaches is low, estimations of the local available bandwidth at c-neighbors may not be accurate. In CACP, we propose two active approaches and one passive approach that can be used to obtain bandwidth information at c-neighbors.

In the first active approach, a node broadcasts queries that have a limited hop count to attempt to contact all c-neighbors. The variant of CACP that adopts this approach is referred to as *CACP-Multihop*. This approach may not work in some topologies since a small hop count may not reach all c-neighbors and a large hop count may reach too many nodes. For example, in Fig. 5, if Node A sends queries limited to 2 hops, Nodes G and E cannot be reached although they are c-neighbors of Node A. By sending queries limited to 3 hops, Node H is falsely included although it is outside Node A's carrier-sensing range.

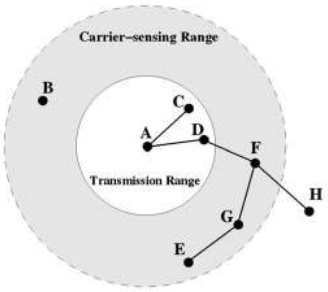

Fig. 5. Multihop approach.

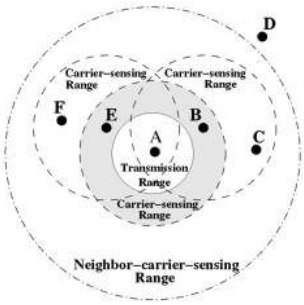

Fig. 6. Neighbor carrier sensing approach.

Furthermore, although Node B is Node A's c-neighbor, Node B cannot be reached by the queries no matter how many hops the queries take. The cost of this approach is based on the number of hops used to reach the c-neighbors. The first message is broadcast by the querying node. At each successive hop, all nodes that receive the query rebroadcast the message. Since all of these transmissions are broadcast at the MAC layer, the amount of contention introduced by this approach may cause bad hot spots around the querying node and its n-hop neighbors. Since, in IEEE 802.11, the carrier-sensing range is about twice the transmission range, in all our simulations based on IEEE 802.11, CACP-Multihop uses two hops as its broadcast range.

The second active approach takes advantage of the power control capabilities of today's wireless technologies. With the help of additional hardware, a sender is able to use a larger transmission power level for its queries than the transmission power level used for normal data transmission. Using this approach, the queries from the sender can reach all of its c-neighbors. The version of CACP that uses this approach is denoted as *CACP-Power*. In this approach, the query messages with increased power may contend/interfere with more nodes than a message at the normal power level. To compensate, only one message is needed to reach all c-neighbors. It is important to note that this technique is only necessary to transfer bandwidth information, which is a much rarer event than the transmission of data messages. Data messages are sent at the normal transmission power level, since increased transmission power for normal data communications can reduce the capacity of the network as shown in [23].

The third approach that we propose is a passive approach in which no query messages are sent to c-neighbors. During normal IEEE 802.11 operations, a node passively monitors the medium using a threshold called the *Neighbor-carrier-sensing Threshold*, which is set much lower than the *Carrier-sensing Threshold*. The sensing range using this threshold, called the *neighbor-carrier-sensing range*, covers the carrier sensing ranges of all of the sensing node's c-neighbors as shown in Fig. 6. When the signal strength of the carrier sensed by a node is smaller than the Neighbor-carrier-sensing Threshold, there is no communication activity in its c-neighborhood and all c-neighbors of the node experience idle channels. By measuring the amount of time that the channel is in this *idle neighbor state*, $T_{idle}^{neighbor}$, for every period of time, T_p, c-neighborhood available bandwidth, $B_{neighbor}$, can be approximated by the following moving average:

$$B_{neighbor} \approx \alpha B_{neighbor} + (1-\alpha)\frac{T_{idle}^{neighbor}}{T_p} B_{channel}. \quad (2)$$

The variant of CACP that uses this approach is referred to as *CACP-CS*. Although this approach has the lowest message overhead compared to CACP-Multihop and CACP-Power, the c-neighborhood available bandwidth estimation in CACP-CS is conservative as the example in Fig. 6 shows. In Fig. 6, Nodes E and B are in Node A's carrier-sensing range. Node F is in Node E's carrier-sensing range while Node C is in Node B's carrier-sensing range. Both Nodes F and C are outside Node A's carrier-sensing range and inside Node A's neighbor-carrier-sensing range. Assume the channel capacity is 2Mbps and Nodes F and C are transmitting 1Mbps, respectively. Since the local available bandwidth at Nodes E and B are both 1Mbps, the c-neighborhood available bandwidth at Node A should be 1Mbps. However, when either Node C or F is transmitting, the channel at Node A is not in the idle neighbor state. Therefore, as long as Nodes F and C's transmissions do not completely overlap, Node A's c-neighborhood bandwidth estimation calculated using the Neighbor-carrier-sensing Threshold will be smaller than 1Mbps. This is because, by simply monitoring the medium, Node A does not know that Node C is outside of Node E's carrier-sensing range and does not consume Node E's bandwidth. Therefore, Node A can only conservatively assume that any transmission activity in its neighbor-carrier-sensing range consumes bandwidth at all of its c-neighbors.

Evaluations of all three versions of CACP in terms of their accuracy and message overhead are presented in Section 5.

3.2 Bandwidth Consumption

The second challenge for CACP is to quantify the bandwidth that a new flow will consume so that it can be decided whether the available bandwidth can satisfy the requirements of the flow. First, the application's data rate must be translated into the corresponding channel bandwidth requirement. In this translation, the protocol overhead in the MAC layer and networking layer must be considered. For example, if IEEE 802.11 is used at the MAC layer, for each application data packet, the MAC layer performs an RTS-CTS-DATA-ACK handshake. Therefore, the transmission time of each data packet, T_{data}, can be expressed as:

$$T_{data} = T_{difs} + T_{rts} + T_{cts} + \frac{L+H}{B_{channel}} + T_{ack} + 3T_{sifs}, \quad (3)$$

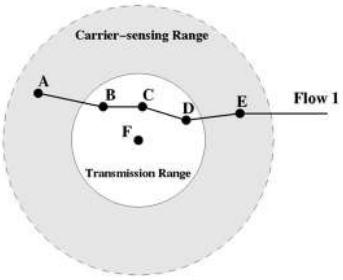

Fig. 7. Bandwidth consumption of a multihop flow.

where L is the size of the data packet and H is the IP and MAC packet header length. T_{rts}, T_{cts}, and T_{ack} represent the time for transmitting RTS, CTS, and ACK packets, respectively. T_{sifs} and T_{difs} denote the interframe space SIFS and DIFS that are defined in the IEEE 802.11 standard. If, at every second, the application generates R packets with average packet size L, the corresponding channel bandwidth requirement, W, of the flow can be expressed as:

$$W = R \times T_{data} \times B_{channel}. \quad (4)$$

Second, because multiple nodes on the route of a flow may contend for bandwidth at a single location, each of these nodes consumes an amount of bandwidth that equals W at this location. The number of these nodes is called the *contention count* of the route and is denoted as N_{ct}. Therefore, the bandwidth consumption of the flow at this location, B_c, can be expressed as:

$$B_c = N_{ct} \times W. \quad (5)$$

For example, in Fig. 7, Flow 1 goes through route $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$. Since Nodes A, B, C, D, and E are in Node F's carrier-sensing range, they all contend for bandwidth at Node F and the contention count at Node F is 5. If the bandwidth requirement of Flow 1 is 2kbps, at least 10kbps bandwidth is consumed by Flow 1 at Node F.

In general, assume Node P is a c-neighbor of n transmitting nodes on the route of a flow that requires Wbps channel bandwidth. If the flow goes through Node P, the contention count of the flow at Node P is $(n+1)$ and the bandwidth consumption is $(n+1)W$bps; if the flow does not go through Node P, the contention count is n and the bandwidth consumption is nWbps. To calculate the contention count, Node P must collect the identities of its c-neighbors, called the *c-neighbor set* (S_{c-nb}). Given the c-neighbor set and the route of a flow, the contention count, N_{ct}, at Node P is determined by the number of common nodes between the c-neighbor set (S_{c-nb}) and the route, expressed as:

$$N_{ct}(P) = \begin{cases} |(\text{Route} - \text{Dest.}) \cap S_{c-nb}|, & \text{if } P \notin \text{Route} \\ |(\text{Route} - \text{Dest.}) \cap S_{c-nb}| + 1, & \text{if } P \in \text{Route}, \end{cases} \quad (6)$$

where the destination is subtracted from the route since the destination only passively receives traffic and, hence, does not contend for the channel. Both active and passive methods may be used to collect the c-neighbor set. For the active method, every node periodically broadcasts a hello message to its one-hop neighbors. The hello message carries the initiator's identity as well as its k-hop neighbors' identities and hop counts learned through listening to other nodes' broadcasts. A node receiving a hello message caches the identities of the initiator and the k-hop nodes in its c-neighbor set. Eventually, every node learns the identities of its $(k+1)$-hop neighbors. Given a route of a flow, a node can immediately tell how many nodes on the route are in its k-hop neighborhood. This method may not work in some topologies since a small k may not include all c-neighbors while a large k may involve too many non-c-neighbors. Furthermore, this method imposes relatively high message overhead in the network due to the periodic hello messages. In the passive method, instead of actively exchanging hello messages, the c-neighbors can be learned through passively monitoring the routes and initiators information carried in control and data messages. This method explores the broadcast nature of wireless channel and imposes no extra message overhead on the network. It can be achieved because CACP uses source routing for the reasons discussed in Section 4.1. For example, in Fig. 7, if Node F hears a message that Node D sends to Node E with source route $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$, Node F learns that Node D is its one-hop neighbor since it can hear Node D's transmission. Node F also knows that Node E and Node C are at most two hops away, Node B is at most three hops away, and Node A is at most four hops away from the source route information. If Node F hears a message from Node B with the same route, Node F is able to update its distance to Node B as one hop and to Node A as two hops. In this way, Node F learns accurate information about the distance to its c-neighbors gradually by monitoring traffic. By caching the identities of nodes within k hops in its c-neighbor set, Node F can determine the contention count of a flow. A likely concern of this method is that the c-neighbor set may not be complete at the time that the bandwidth consumption of a flow must be calculated to perform admission control. In Section 4, a step by step example shows that the c-neighbor set built through this passive method is complete enough at the time that an accurate estimation of the contention count of a flow is needed. In CACP, this passive method is used for collecting c-neighbor sets due to its low message overhead. Since the carrier-sensing range in IEEE 802.11 is around twice the transmission range, in our simulations, the c-neighbor set caches 2-hop neighbors.

4 Basic Protocol Design

CACP (Contention-aware Admission Control Protocol) works with single channel MAC layer protocols such as IEEE 802.11 [1], IEEE 802.11e [2], and SEEDEX [3], etc., and performs both bandwidth-aware routing and admission control. The routing of CACP maintains low message overhead in the presence of mobility and enables a natural integration with the local approach of bandwidth consumption estimation. CACP consists of four parts: route discovery, admission control, building c-neighbor sets, and mobility management.

4.1 Route Discovery

The aim of route discovery is to find a route between the sender and the receiver that has enough resources for the flow. CACP uses on-demand route discovery with source

routing, similar to DSR [5]. We use a source routing-based approach because it allows CACP to specify directly which route the flow will use so that the packets for the flow are ensured to only go through the specified route that has been admitted by the admission control and has enough bandwidth for the flow. It also provides easy traffic splitting at the source node so that two flows with the same destination can follow different routes to avoid creating hot spots in the network. Other routing protocols that do not use source routing, such as DSDV [24], AODV [25], and TORA [26], do not pin a flow to its route and, hence, may potentially route the packets of the flow to some other route where there is not enough resources for the flow.

To reduce the message overhead of route discovery, CACP performs *partial admission control* (see details in Section 4.2) during the process of route discovery to preliminarily eliminate routes without enough bandwidth. When a source node has data to send, it broadcasts a route request to its neighbors. The route request contains the bandwidth requirement of the connection calculated using (4), the address of the initiator of the request, the address of the destination, and a record of the sequence of hops taken by the route request as it is propagated through the ad hoc network. We call this sequence of hops the *partial route (PRoute)*, which is used to determine the lower bound of the contention count of the full route and can also be used to eliminate circular routes. Each node that receives a route request performs partial admission control to determine if the network has enough bandwidth for the flow along the partial route. If the partial admission control fails or if the partial route has loops, the route request is dropped. Otherwise, the node adds its own address to the partial route and rebroadcasts the route request.

When the intended destination node receives a route request, the partial route in the route request becomes a *full route*. The destination then reverses the full route and sends a route reply back to the source along that route. If multiple route requests carrying different routes arrive at the destination, the destination only sends the route reply along one route based on some selection rules, such as shortest route or first route request. The other routes are cached for a short period of time as backup in case the first route reply does not reach the source due to link breakage or admission failure. At each node that the route reply traverses, *full admission control* (see details in Section 4.2) is performed. If admission control succeeds at a node, a soft reservation of bandwidth is set up in the node and a route reply is forwarded to the next hop. Otherwise, an admission failure message is sent to the destination. In this case, the soft reservations of bandwidth along the route are explicitly torn down when nodes along the route receive the admission failure message. When the destination node receives the admission failure message, it selects another cached route and sends a route reply along it. When a route reply successfully arrives at the source, enough end-to-end bandwidth has been reserved for the flow and communication can start. Since the soft reservation of bandwidth along the route is refreshed by the arrival of data packets, if no packet arrives due to link breakage after a node forwards a route reply, the soft reservation at the node times out.

4.2 Distributed Admission Control Algorithm

Route discovery finds multiple possible routes to reach a destination. Admission control must be used to determine which of these routes can admit the new flow. At each node on the route, admission decisions must be based on the expected bandwidth consumption of the flow as well as the available bandwidth at the node and its c-neighbors. Admission control in CACP is performed in two phases of route discovery. First, partial admission control is performed during the route request phase when a node receives a route request. Second, full admission control is performed during the route reply phase when a node receives a route reply. CACP separates admission control in the two phases because, in the route request phase, the full route to the destination is unknown. Since the contention count of the full route cannot be calculated in this phase, the expected bandwidth consumption of the flow calculated using the partial route carried in a route request may be smaller than the actual bandwidth consumption of the final route. Therefore, admission control in this phase is not accurate due to an underestimation of the bandwidth consumption of flow and, hence, is called *partial admission control*. Since partial admission control may be over-optimistic in admitting flows, it is used as a first pass to cheaply weed out routes and reduce the message overhead by avoiding flooding route requests in hot spots. The effectiveness of partial admission control in reducing the message overhead is demonstrated by simulations in Section 6. During the route reply phase, since the full route to the destination is known, the admission control is accurate and, hence, is called *full admission control*.

4.2.1 Partial Admission Control

In CACP, when a node receives a route request, partial admission control is performed by comparing available bandwidth with the possibly underestimated bandwidth consumption that is calculated using the partial route (see (5) and (6)). However, the types of available bandwidth used in partial admission control in the three versions of CACP (CACP-Multihop, CACP-Power, and CACP-CS) are different. In CACP-Multihop and CACP-Power, since estimating c-neighbor available bandwidth involves querying c-neighbors, which is an expensive operation, it is not desirable to perform this operation on nodes that are not along a viable route to the destination. To reduce overhead, as route requests are flooded through the whole network during the route request phase, only local available bandwidth is estimated according to (1) and compared with the bandwidth consumption of the flow. If the local available bandwidth is smaller than the bandwidth consumption of the flow, admission control fails. Otherwise, admission control succeeds and the route request can be forwarded to the next hop. In CACP-CS, since estimating c-neighborhood available bandwidth does not impose extra message overhead on the network, the bandwidth consumption of the flow is compared to both local and c-neighbor available bandwidth estimated using (2).

4.2.2 Full Admission Control

In the route reply phase, when a node receives a route reply, it performs full admission control. First, the bandwidth consumption of the flow at the node's location is calculated and compared to the node's local available bandwidth. Since the route reply carries the full route, the estimation of bandwidth consumption in this phase is

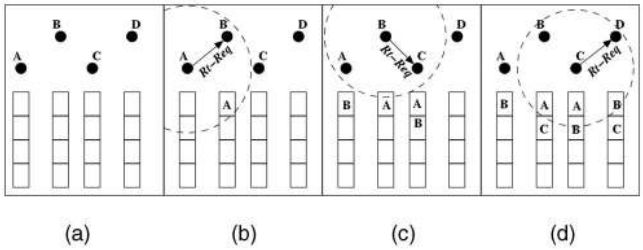

Fig. 8. Route request (CACP-Multihop, CACP-Power, and CACP-CS).

accurate. If the local available bandwidth is larger than the bandwidth consumption of the flow, the node proceeds to compare its c-neighborhood available bandwidth with the bandwidth consumption of the flow.

As discussed in Section 3.1.2, the three variants of CACP take different approaches to achieve this. In the active approach used in CACP-Power and CACP-Multihop, the node that receives the route reply broadcasts (via multihop or enhanced power techniques) an admission request message, which carries the full route of the flow, to its c-neighborhood. Each node that receives the admission request message calculates the bandwidth consumption of the flow at its location according to (4), (5), and (6) and compares it with its local available bandwidth. If the bandwidth consumption of the flow is larger than the node's local available bandwidth, the node sends an admission rejection message back to the initiator and the c-neighborhood admission control fails. If the initiator of the admission request message does not receive any admission rejection message after a certain period, it times out and assumes that the full admission control succeeds. The length of this timeout period is determined by the channel propagation delay, the transmission time of the admission rejection message, and the computation time. Since the admission request message is a broadcast message, some c-neighbors may not receive the admission request message due to collisions. However, since a node is usually the c-neighbors of multiple nodes along a route, the probability that the node does not receive any admission request message is slim. For example, in Fig. 7, Node F is c-neighbors of Nodes A, B, C, D, and E. During the admission control of Flow 1, Nodes A, B, C, D, and E all broadcast admission request messages. As long as Node F receives one of these admission request messages, the admission control is correctly performed.

In CACP-CS, a passive approach is used to obtain c-neighborhood available bandwidth, where no admission request/rejection messages are needed. The node that receives the route reply directly estimates its c-neighborhood available bandwidth using (2) and compares it with the bandwidth consumption of the flow to make admission decisions. As shown in Section 3.1.2, the message overhead and the accuracy of the full admission control schemes in the three versions of CACP are different and the comparisons of their performance are presented in Section 6.

4.3 Building the c-Neighbor Set

As described in Section 4.2, the correctness of admission control depends on the completeness and accuracy of the *c-neighbor set*. In CACP, information about c-neighbors is acquired by monitoring control and data messages and cached in the c-neighbor set. If there is no communication activity in a node's neighbor area for a long time, the c-neighbor set entries at this node may all time out. An important concern is that, if a new flow needs to go through this node, the node would not be able to give accurate estimation of the bandwidth consumption of the flow due to its empty c-neighbor set. However, the following analysis shows that this case rarely happens. During the route request phase, since the route requests are flooded through the whole network and carry the partial routes that they have traversed, a node can collect its c-neighbors information by caching the last two hops of the partial routes into its c-neighbor set. In addition, since in CACP-Multihop and CACP-power the admission request messages are sent to reach c-neighbors, the c-neighbor set can be further completed by caching the senders and forwarding nodes of these messages in the c-neighbor set. Finally, when a node receives a route reply, it also can add the last two nodes that forwarded the route reply into its c-neighbor set. Therefore, at the time that the node needs to perform full admission control, its c-neighbor set has already been filled up with high probability.

4.4 An Example

To better illustrate the three variants of CACP, we present an example of the process of route discovery and admission control for each variant in Figs. 8, 9, 10, 11, 12, 13, and 14. The tables by the side of the nodes show the c-neighbor set cached at each node. The carrier-sensing range in the example is assumed to be twice the transmission range and hence the c-neighbor sets cache 2-hop neighbors. Initially, no node has received any messages and the c-neighbor sets are empty (See Fig. 8a).

As Node A initiates a connection to Node D, Node A first translates the rate requirement of the connection into its bandwidth requirement, W, according to (4). Then, Node A

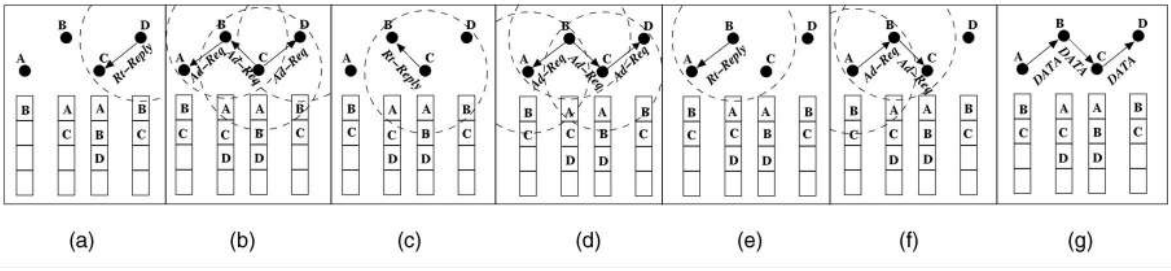

Fig. 9. Admission success of CACP-Multihop.

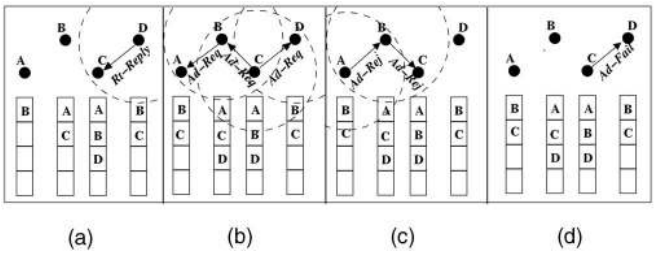

Fig. 10. Admission failure of CACP-Multihop.

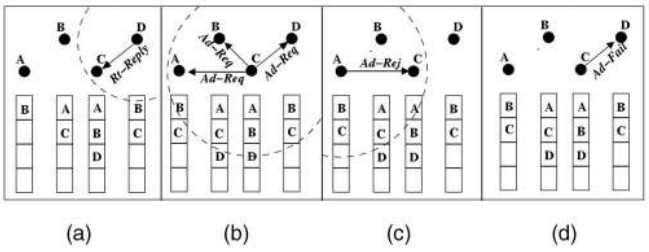

Fig. 12. Admission failure of CACP-Power.

invokes partial admission control. If the flow passes the partial admission control, Node A broadcasts a route request with partial route {A} and bandwidth requirement W (See Fig. 8b). When Node B receives the route request, it adds Node A to its c-neighbor set, updates the partial route to {A, B} and performs partial admission control. Based on (5) and (6), the bandwidth consumption of the flow at Node B can be estimated as:

$$\begin{aligned} B_c &= N_{ct}(B) \times W \\ &= [|(\text{PRoute} - \text{Dest.}) \cap S_{c-nb}| + 1] \times W \\ &= [|\{A, B\} \cap \{A\}| + 1] \times W \\ &= 2W. \end{aligned}$$

As mentioned in Section 4.2, B_c is underestimated since only the partial path is known. However, it does help to weed out all routes that start with $A \rightarrow B$ if it can be decided that B does not have enough bandwidth. In CACP-Multihop and CACP-Power, Node B compares B_c with its local available bandwidth in (1). If it has enough local available bandwidth, partial admission control succeeds and Node B rebroadcasts the route request with partial route {A, B} (See Fig. 8c). In CACP-CS, due to the lightweight estimation of the c-neighborhood available bandwidth, Node B also compares B_c with its c-neighborhood available bandwidth from (2) during the partial admission control.

When Node A receives the route request from Node B, it adds Node B to its c-neighbor set. Since Node A is already in the partial route, it drops the route request silently to avoid creating a circular route. When Node C receives the route request from Node B, it adds both Nodes A and B to its c-neighbor set and performs partial admission control similar to Node B. If the flow passes the partial admission control, Node C broadcasts the route request with partial route {A, B, C} (See Fig. 8d). When Node B receives the route request, it caches Node C in its c-neighbor set and drops the route request. When destination D receives the route request, it adds Nodes C and B to its c-neighbor set and performs partial admission control. If the partial admission control succeeds, Node D reverses the route and sends a route reply back to Node C (See Fig. 9a, Fig. 11a, and Fig. 13a).

In CACP-Multihop and CACP-Power, when Node C receives the route reply, it performs full admission control by first comparing its local available bandwidth with the bandwidth consumption of the flow, calculated as:

$$\begin{aligned} B_c &= N_{ct}(B) \times W \\ &= [|(\text{Route} - \text{Dest.}) \cap S_{c-nb}| + 1] \times W \\ &= [|(\{A, B, C, D\} - \{D\}) \cap \{A, B, D\}| + 1] \times W \\ &= 3W. \end{aligned}$$

If the flow passes the first step, Node C broadcasts an admission request message through multihop or enhanced power to all c-neighbors (See Fig. 9b and Fig. 11b). Node A hears the message, adds Node C to its c-neighbor set and calculates the bandwidth consumption of the flow as $3W$ because:

$$\begin{aligned} N_{ct}(A) &= |(\text{Route} - \text{Dest.}) \cap S_{c-nb}| + 1 \\ &= |(\{A, B, C, D\} - \{D\}) \cap \{B, C\}| + 1 \\ &= 3. \end{aligned}$$

If Node A's local available bandwidth cannot accommodate the flow, it sends an admission rejection message back to Node C and Node C informs Node D about the failure of admission. (See Figs. 10a, 10b, 10c, and 10d and Figs. 12a, 12b, 12c, and 12d). If all of Node C's c-neighbors have enough local available bandwidth, no node sends an admission rejection message. After Node C times out, it sets up a soft reservation of the bandwidth and forwards the route reply to Node B (See Fig. 9c and Fig. 11c). If the flow passes the full admission control at Node B, Node B forwards the route reply to Node A. After the flow passes the full admission control at Node A, the route $A \rightarrow B \rightarrow C \rightarrow D$ is known to have enough bandwidth for the flow and

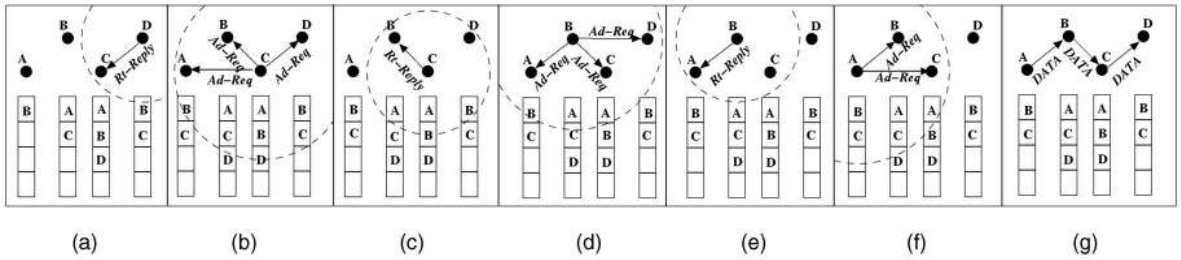

Fig. 11. Admission success of CACP-Power.

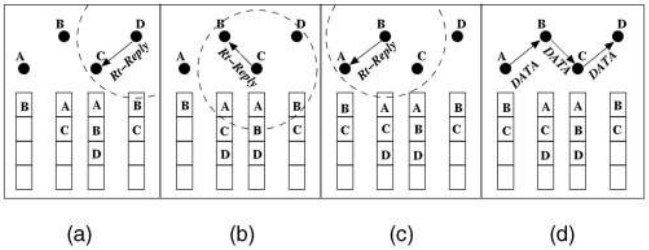

Fig. 13. Admission success of CACP-CS.

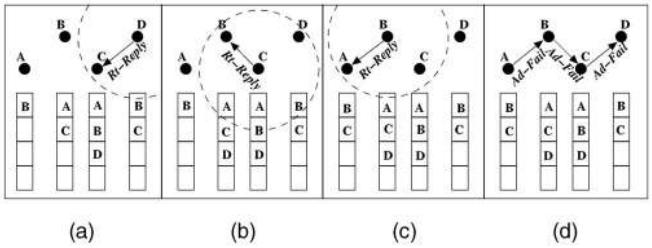

Fig. 14. Admission failure of CACP-CS.

the data messages can start (See Figs. 9a, 9b, 9c, 9d, 9e, 9f, 9g, and 9h and Figs. 11a, 11b, 11c, 11d, 11e, 11f, 11g, and 11h).

In CACP-CS, no admission request message is sent as seen in Fig. 13. When a node receives a route reply, it performs full admission control by comparing the bandwidth consumption of the flow with the directly estimated c-neighbor available bandwidth calculated using (2) as well as its local available bandwidth. If there is enough c-neighborhood available bandwidth for the flow, the route reply is forwarded to the next hop until it reaches Node A, as seen in Fig. 13. If at any node on the route, for example, Node A, the c-neighborhood available bandwidth is smaller than the bandwidth consumption of the flow, an admission rejection message is sent back to the destination, informing it about the admission control failure as seen in Fig. 14.

4.5 Mobility Management

As mentioned in Section 2.2, strict QoS cannot be guaranteed in ad hoc networks since the nodes of an ad hoc network are inherently subjected to mobility that is beyond any protocol's ability to predict or control. Therefore, it is likely that QoS violations can be quite frequent in ad hoc networks. To deal with such dynamics, each node monitors the actual sending rates of its flows. If a node notices that one of its flows does not get the reserved bandwidth due to increased congestion levels, or if the next hop of the flow moves out of the range of the node, a notification message is sent to the source of the flow indicating changes in the route. The source can either search for a new route or reduce its QoS requirement to accommodate the broken or degraded route. Since the reestablishment of a QoS commitment may take a long time and cost extra message overhead, it is desirable to reduce the frequency of QoS violations. A possible method is to reserve some resources for unexpected events, such as when a new node moves into carrier-sensing range of a flow and contends for bandwidth with the flow. If some extra bandwidth is reserved to compensate for this, the QoS of the flow may still hold. However, a tradeoff must be made since the reserved bandwidth may never be used, reducing the efficiency of bandwidth usage in the network. We are currently investigating these effects of mobility to balance high throughput with effective admission control.

5 Message Overhead of c-Neighborhood Available Bandwidth Estimation

In this section, we compare the overhead for estimating c-neighbor available bandwidth. Since CACP-CS does not introduce any additional message overhead for estimating c-neighbor available bandwidth, we only compare the overhead of CACP-Multihop and CACP-Power.

Each time a control message is sent at an enhanced power level or in multihop mode, it causes more interference to the network than had it been sent as a normal message. We use the total number of times that nodes receive admission request and admission rejection messages as the measurement for the overhead of the two approaches. We first provide an analytical evaluation and then verify it through simulations.

5.1 Analytical Study

Suppose the transmission range of a node is R and the carrier-sensing range is $2R$. The total number of nodes in the network is a constant M. $\rho(x,y)$ is the density function, which represents the number of nodes in a unit area centered at location (x,y).

Let Δ_{xy} represent a very small square region centered at location (x,y) with area Δ. The number of nodes in Δ_{xy} can be expressed as $\rho(x,y)\Delta$. Assume that each node in the network has the same probability q to generate an admission request per unit time. Then, in Δ_{xy}, the expected number of admission requests per unit time is $q\rho(x,y)\Delta$. Assume CACP-multihop is used with 2 hops, which is most conservative in terms of overhead but may not reach all nodes in carrier-sensing range. For each admission request, $\pi R^2\rho(x,y)$ nodes hear the message and each rebroadcasts the message to its own $\pi R^2\rho(x,y)$ neighbors. Therefore, an admission request message is received $\pi R^2\rho(x,y) + (\pi R^2\rho(x,y))^2$ times in CACP-Multihop. If CACP-Power is used, $4\pi R^2\rho(x,y)$ nodes hear the admission request message.

Based on the above analysis, if we divide the whole network area into square regions with area Δ, the expected ratio of the overhead of the two approaches, Φ, can be expressed as

$$\begin{aligned}\Phi &= \frac{\text{Message Overhead of CACP-Multihop}}{\text{Message Overhead of CACP-Power}} \\ &= \lim_{\Delta\to 0}\frac{\sum_{i=1}^{n}\left\{\pi R^2\rho(x_i,y_i) + \left[\pi R^2\rho(x_i,y_i)\right]^2\right\}q\rho(x_i,y_i)\Delta}{\sum_{i=1}^{n}4\pi R^2\rho(x_i,y_i)q\rho(x_i,y_i)\Delta} \\ &= \frac{1}{4} + \frac{\pi R^2}{4}\lim_{\Delta\to 0}\frac{\sum_{i=1}^{n}\rho^3(x_i,y_i)}{\sum_{i=1}^{n}\rho^2(x_i,y_i)},\end{aligned} \tag{7}$$

where (x_i,y_i) is the location of the i_{th} Δ region and $n = \frac{\text{Area of the network}}{\Delta}$.

According to general means inequality, for n positive numbers $X_1, X_2, \cdots X_n$,

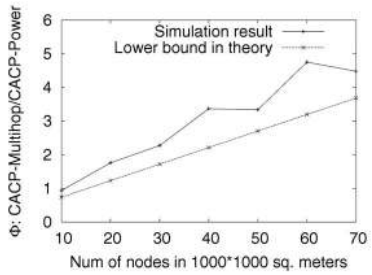

Fig. 15. Overhead ratio of CACP-Multihop versus CACP-Power.

$$\frac{\sum_{i-1}^{n} X_i^3}{\sum_{i-1}^{n} X_i^2} \geq \left(\frac{1}{n}\sum_{i-1}^{n} X_i^2\right)^{\frac{1}{2}} \geq \frac{1}{n}\sum_{i-1}^{n} X_i, \tag{8}$$

where the equality holds when $X_1 = X_2 = \cdots = X_n$. Based on (7) and (8),

$$\begin{aligned}\Phi &\geq \frac{1}{4} + \frac{\pi R^2}{4} \lim_{\Delta \to 0}\left[\frac{1}{n}\sum_{i=1}^{n}\rho(x_i, y_i)\right] \\ &= \frac{1}{4} + \frac{\pi R^2}{4} \lim_{\Delta \to 0} \frac{\sum_{i=1}^{n}\rho(x_i, y_i)\Delta}{n\Delta} \\ &= \frac{1}{4} + \frac{\pi R^2}{4} \frac{M}{\text{Area of the network}},\end{aligned} \tag{9}$$

where the equality holds if $\rho(x, y)$ is a constant for all (x, y). Hence, the theoretical lower bound of the overhead ratio of CACP-Multihop 3 CACP-Power is $\left(\frac{1}{4} + \frac{\pi R^2 M}{4(\text{Area of the network})}\right)$, which can be achieved when the density of nodes in the network is constant. Therefore, the higher the density of the network, the lower the overhead of CACP-Power compared to CACP-Multihop. If the number of neighbors in a node's transmission range, $\pi R^2 \rho(x, y)$, exceeds 3, CACP-Multihop has a higher overhead than CACP-Power.

5.1.1 Simulation and Results

Simulations are used to verify the results of the above analysis under different node densities. In the simulations, 10 to 70 mobile hosts are randomly distributed in $1{,}000 \times 1{,}000m^2$. Five scenarios are examined for each density. Radio transmission range is 250m and carrier-sensing range is 550m. The bandwidth of the channel is 2 Mbps. Ten pairs of nodes are randomly chosen to establish connections with a 512B 10 packets/s CBR traffic source. The movement pattern of nodes is the random way-point model. The speed of nodes is 5m/s and the pause time is 20s. The simulations run for 200s.

Fig. 15 shows the overhead ratio of CACP-Multihop versus CACP-Power from the simulations and the theoretical analysis. As expected, the overhead ratio from the simulations is higher than the lower bound calculated. This is because, in the simulations, the nodes are randomly distributed in the network and may cluster in some parts of the network. As (9) shows, when the nodes are not absolutely evenly distributed, the overhead ratio is higher than the theoretical lower bound. The simulations support that CACP-Power has a lower overhead than CACP-Multihop when the average density of nodes is more than 15.3 nodes/$10^6 m^2$. Since 15.3 nodes/$10^6 m^2$ is usually too low a density to maintain connectivity of an ad hoc network as shown in [27], it can be concluded that for a well-connected random ad hoc network, CACP-Power has a lower message overhead than CACP-Multihop.

TABLE 1
Configurations of CBR Sources

Flow No.	Rate (Pkts/s)	Packet Size (Bytes)	Starting Time (s)
1	13.5	112	10
2	42.65	381	20
3	35.55	311	30
4	16.99	481	60
5	37.69	519	80
6	18.69	855	90
7	44.04	317	100
8	46.20	786	110
9	14.92	402	140

6 Evaluation

In this section, we evaluate CACP by simulations in ns2 [20]. The simulations use IEEE 802.11 [1] as a MAC layer since it has almost no support for QoS. Therefore, the simulation results demonstrate the effectiveness of CACP rather than that of QoS scheduling algorithms. Since IEEE 802.11 exhibits temporary unfairness due to the binary exponential backoff after a collision, CACP is only able to provide guarantees in terms of average performance of the flows over short periods of time (1 second) instead of instantaneous throughput/delay of the flows. By integrating QoS scheduling algorithms [3], [12], [13] with CACP, we expect that better performance and QoS guarantees in finer granularity can be achieved. The accuracy of bandwidth management and the overhead of the three versions of CACP (CACP-Multihop, CACP-Power and CACP-CS) are compared with DSR [5] and SWAN [4].

6.1 Illustration of Effectiveness

To illustrate the effectiveness of CACP, we first present a simple simulation in a $1{,}000m \times 1{,}000m$ static network with 20 randomly positioned nodes. Nine connections of CBR sources are attempted to be established in the network with their destinations and sources randomly chosen. The packet size, rate, and starting time of each connection is shown in Table 1.

Fig. 16 and Fig. 21 show the throughput and delay of the nine flows when DSR is used. Since no admission control is performed in DSR, the network becomes congested as new flows are added to the network, resulting in decreased throughput and dramatically increased delay of the flows.

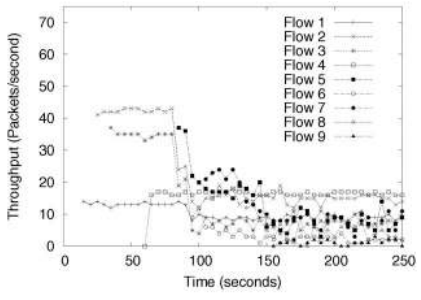

Fig. 16. Throughput of DSR.

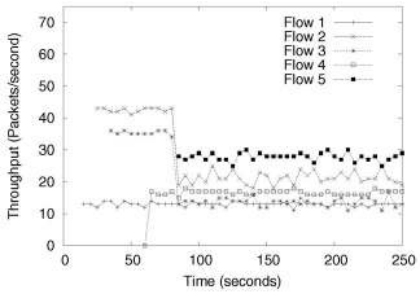

Fig. 17. Throughput of SWAN.

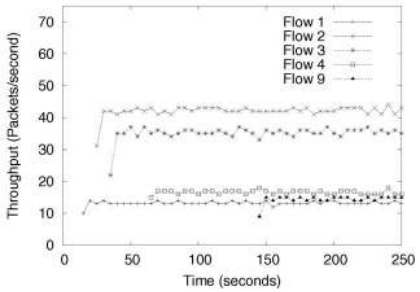

Fig. 18. Throughput of CACP-Multihop.

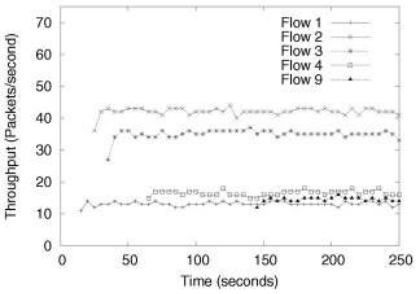

Fig. 19. Throughput of CACP-Power.

Fig. 17 and Fig. 22 depict the throughput and delay of the flows when SWAN is used. It shows that only Flows 1 to 5 are admitted by SWAN and the throughput of the flows are more stable and the delay of the flows is much lower than in DSR. However, the throughput of the flows still shows significant degradation and the delay still increases dramatically as the number of flows increases. The reason is that SWAN does not consider contention between flows located in each other's c-neighborhood. Therefore, SWAN may falsely admit flows that may affect the QoS of existing neighboring flows as shown in the throughput and delay of

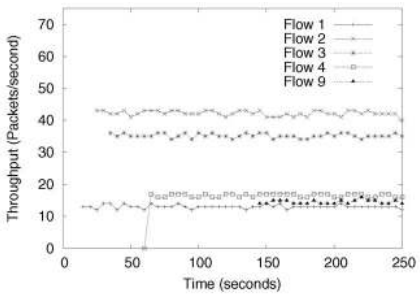

Fig. 20. Throughput of CACP-CS.

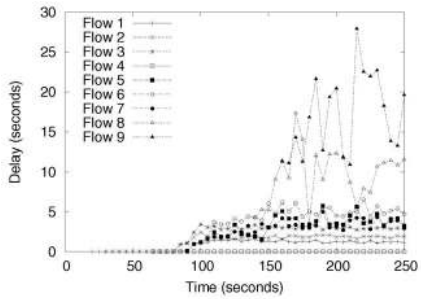

Fig. 21. Delay of DSR.

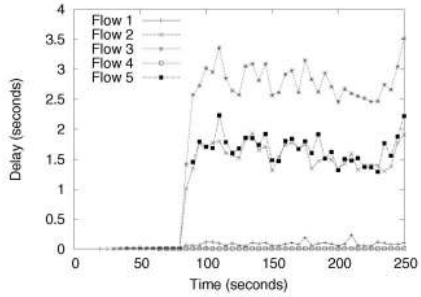

Fig. 22. Delay of SWAN.

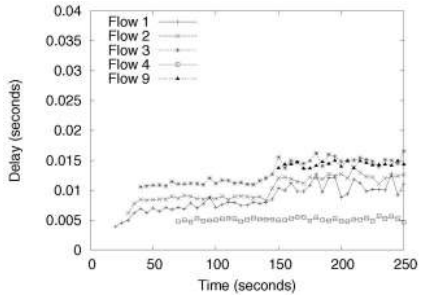

Fig. 23. Delay of CACP-Multihop.

Flows 2 and 3 in Fig. 17 and Fig. 22. In Figs. 18, 19, 20, 23, 24, and 25, the throughput and delay of the flows in CACP-Multihop, CACP-Power and CACP-CS are shown. As can be seen, all three versions of CACP maintain the throughput of the admitted flows. The worst delay of the flows in all three versions of CACP is below 35ms, which is 100 times smaller than the worst delay of SWAN, 3.5s, and 823 times smaller than the worst delay of DSR, 28s. (Note that the scales of Fig. 22 and Fig. 21 are different from each other and are much larger than the scale used in Figs. 23, 24, and 25.

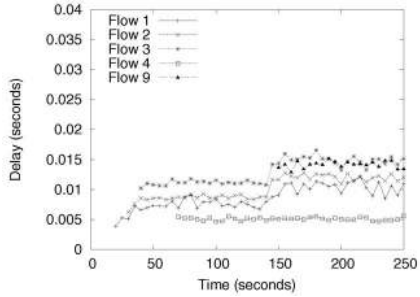

Fig. 24. Delay of CACP-Power.

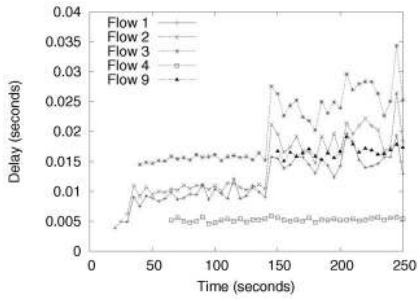

Fig. 25. Delay of CACP-CS.

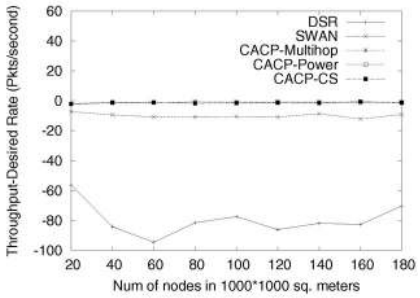

Fig. 26. Rate of QoS violation.

6.2 Accuracy of Bandwidth Management

In this section, we examine the accuracy of CACP's admission control quantitatively. The accuracy of admission control can be evaluated using two metrics. The first metric is the number of *false admissions*. A false admission means that admission control admits a flow whose bandwidth consumption is beyond the capacity of the network. A falsely admitted flow either degrades the QoS of admitted flows or is not able to achieve its own desired QoS. Therefore, the rate of QoS violations of admitted flows can indicate the number of false admissions, which is defined as the summation of the actual throughput of admitted flows subtracting the summation of the traffic generation rate of their CBR sources. Ideally, admission control should keep the QoS violation at zero and a negative value of the QoS violation indicates false admission. The second metric to evaluate the accuracy of admission control is *bandwidth utilization*. If an admission control algorithm is over-conservative so that it rejects flows whose bandwidth consumptions are not beyond the capacity of the network, this admission control algorithm hurts the bandwidth utilization of the network by reducing the amount of traffic that the network can carry. Hence, the total throughput of all admitted flows in the network indicates the bandwidth utilization. In our simulations, the QoS violations and total throughput of admitted flows in CACP-Multihop, CACP-Power, and CACP-CS are compared with the admission control algorithm used in SWAN [4]. The performance of DSR is also examined to demonstrate the necessity of admission control.

In the simulations, 450 $1{,}000m \times 1{,}000m$ networks are generated randomly. The numbers of nodes in the networks range from 20 to 180. Each simulation runs 200 seconds. Twenty randomly chosen pairs of nodes try to establish a connection with each other with a CBR traffic source. The packet rates of the CBR sources are uniformly distributed in $[10, 50]$ packets per second with the packet size uniformly distributed in $[100, 1000]$ Bytes. The random way point model is used for the mobility of nodes with maximum speed 5 meters/seconds and pause time 10 seconds.

Fig. 26 shows the QoS violation rate of CACP-Multihop, CACP-Power, CACP-CS, SWAN, and DSR. The QoS violation rates of all three variants of CACP are very close to 0 and overlap each other regardless of the densities of the network. The QoS violation rates of SWAN and DSR are much larger than the three versions of CACP, indicating that they cause more false admissions. The total throughput of admitted flows is shown in Fig. 27. It can be seen that the throughput of all three variants of CACP is almost always larger than the throughput of SWAN and DSR. The only exception is that, when the density of nodes is 60 nodes per $10^6 m^2$, SWAN's throughput is close to CACP-CS's throughput, although it is still smaller than the throughput of CACP-Multihop and CACP-Power. This demonstrates that the bandwidth utilization of CACP is high since the capacity of the network is not reduced. In addition, because CACP has significantly fewer false admissions, all versions of CACP reduce the amount of collisions between flows, increasing the capacity of the network. It is also interesting to note that the throughput of CACP-CS is lower than the throughput of CACP-Multihop and CACP-Power when the network density is low, which is due to the conservative c-neighbor bandwidth estimation method (See Section 3.1.2) used in CACP-CS. As the density of the network increases, the throughput of CACP-CS becomes larger than the throughput of CACP-Multihop and CACP-Power. This is because, as the network becomes denser, the message overhead for performing active c-neighbor bandwidth estimation used in CACP-Multihop and CACP-Power increases, which consumes more network capacity than the capacity wasted by CACP-CS's conservative but low overhead bandwidth estimation method. Therefore, CACP-CS is more efficient in terms of bandwidth utilization in dense networks than CACP-Multihop and CACP-Power.

6.3 Admission Control Message Overhead

The effects of admission control on control message overhead is two-fold. On one hand, active admission control methods, like CACP-Multihop and CACP-Power, increase the amount of control messages since they require c-neighbors to exchange control messages for admission control (e.g., admission request and admission rejection messages). On the other hand, effective admission control reduces the number of control messages used for route discovery for two reasons. First, the admission control performed during

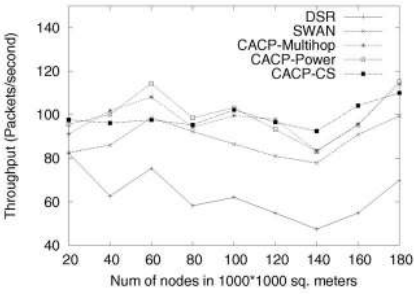

Fig. 27. Network total throughput.

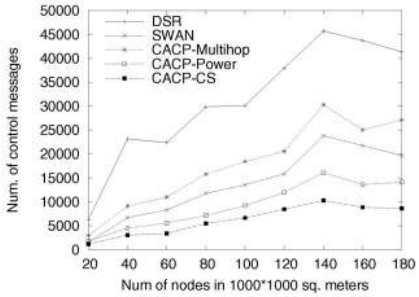

Fig. 28. Control message overhead.

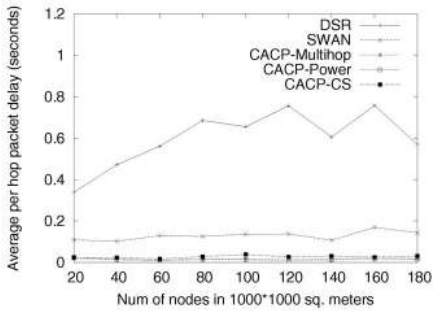

Fig. 29. Average per hop packet delay.

the route request phase can preliminarily eliminate routes that do not have enough bandwidth and, hence, reduce the number of route requests in hot spots. Second, by preventing the network from being overloaded, effective admission control can reduce the number of link breaks due to collisions between neighboring nodes, which in turn reduces the number of control messages caused by reestablishing routes.

To evaluate the effect of CACP on control message overhead, the number of all control messages in the simulations in Section 6.2, including route request, route reply, admission request, and admission rejection messages, is recorded and compared with DSR and SWAN. Fig. 28 depicts the total number of control messages in the simulations. The control message overhead of DSR is the largest, demonstrating that the overall effect of admission control can reduce control message overhead. The control message overhead of CACP-Multihop is larger than SWAN due to its extra message overhead of admission request and admission rejection messages. The control message overhead of CACP-Power is smaller than CACP-Multihop as our discussion in Section 5 has demonstrated. Because CACP-Power reduces control message overhead for route discovery, which compensates for its extra message overhead of contacting c-neighbors, it has less message overhead than SWAN. CACP-CS has the lowest control message overhead. This is not surprising since CACP-CS does not introduce any extra control messages besides the messages used for route discovery. The simulations demonstrate that, even though CACP introduces more types of control messages for performing admission control, its bandwidth-aware routing reduces the total control message overhead. Given the benefit of accurate admission control, CACP's message overhead is acceptable and does not reduce the capacity of the network as shown in Fig. 27.

Even though CACP is mainly designed to manage bandwidth, its bandwidth-aware routing scheme essentially balances the load in the network since routes through hot spots do not get admitted. Therefore, even though CACP achieves higher throughput than SWAN and DSR, it controls the packet delay in the network by avoiding creating congested areas in the network. Fig. 29 presents the average per-hop packet delay in the simulations in Section 6.2. It shows that all three versions of CACP achieve much lower packet delay than SWAN and DSR, indicating CACP's excellent ability to balance network load.

7 Conclusion and Extensions

We have presented three methods to achieve contention-aware admission control on a single channel medium. Our main contribution is the inclusion of information from nodes inside carrier-sensing range and outside transmission range during the admission control process. The success of our protocol is shown through simulations, where CACP effectively manages requests for bandwidth beyond the capabilities of the network, imposing acceptable or even reducing the control message overhead on the network.

Since CACP is only the admission control part of a QoS protocol stack, it can be combined with many existing QoS protocols, such as QoS-aware MAC protocols or end host policing protocols. Different QoS-aware MAC protocols and admission policies may affect the definition of local available bandwidth so that extensions of (1) and (2) may be needed to estimate local/c-neighborhood available bandwidth as shown in our previous work in [22]. However, CACP's process of using local and c-neighborhood available bandwidth to perform admission control should not be affected.

Acknowledgments

This work was partially supported by a Vodafone-US Foundation Graduate Fellowship and US National Science Foundation Grant ITR 0081308.

References

[1] IEEE Computer Society, "802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.
[2] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service," *Proc. European Wireless,* 2002.
[3] R. Rozovsky and P.R. Kumar, "SEEDEX: A MAC Protocol for Ad Hoc Network," *Proc. ACM Symp. Mobile Ad Hoc Networking & Computing,* 2001.
[4] G.-S. Ahn, A. Campbell, A. Veres, and L.-H. Sun, "SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks," *Proc. Infocom,* 2002.
[5] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing,* vol. 353, 1996.
[6] D. Maltz, "Resource Management in Multi-Hop Ad Hoc Networks," Technical Report CMU CS 00-150, School of Computer Science, Carnegie Mellon Univ., July 2000.
[7] Y.-C. Hsu and T.-C. Tsai, "Bandwidth Routing in Multihop Packet Radio Environment," *Proc. Third Int'l Mobile Computing Workshop,* 1997.
[8] T.-W. Chen, J.T. Tsai, and M. Gerla, "QoS Routing Performance in Multihop Multimedia Wireless Networks," *Proc. IEEE Int'l Conf. Universal Personal Comm. (ICUPC),* 1997.

[9] C.R. Lin and C.-C. Liu, "An On-Demand QoS Routing Protocol for Mobile Ad Hoc Networks," *IEEE Global Telecomm. Conf.*, 2000.
[10] C. Zhu and M.S. Corson, "QoS Routing for Mobile Ad Hoc Networks," Technical Report CSHCN TR 2001-18, Inst. for System Research, Univ. of Maryland, 2001.
[11] C.R. Lin and J.-S. Liu, "QoS Routing in Ad Hoc Wireless Networks," *IEEE J. Selected Areas in Comm.*, vol. 17, no. 8, pp. 1426-1438, Nov./Dec. 1999.
[12] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed Multi-Hop Scheduling and Medium Access with Delay and Throughput Constraints," *Proc. Seventh Ann. Int'l Conf. Mobile Computing and Networking*, 2001.
[13] H. Luo, S. Lu, V. Bharghavan, J. Cheng, and G. Zhong, "A Packet Scheduling Approach to QoS Support in Multihop Wireless Networks," *ACM J. Mobile Networks and Applications*, special issue on QoS in heterogeneous wireless networks, 2002.
[14] S.-B. Lee, G.-S. Ahn, X. Zhang, and A. Campbell, "INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks," *J. Parallel and Distributed Computing*, special issue on wireless and mobile computing and communications, vol. 60, pp. 374-406, 2000.
[15] M.G. Barry, A.T. Campbell, and A. Veres, "Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks," *Proc. Infocom*, 2001.
[16] R. Ramanathan and M. Steenstrup, "Hierarchically-Organized, Multihop Mobile Wireless Networks for Quality-of-Service Support," *Mobile Networks and Applications*, vol. 3, no. 1, pp. 101-119, 1998.
[17] S. Murthy and J.J. Garcia-Luna-Aceves, "A Routing Architecture for Mobile Integrated Services Networks," *Mobile Networks and Applications*, vol. 3, no. 4, pp. 391-407, 1999.
[18] S.H. Shah, K. Chen, and K. Nahrstedt, "Dynamic Bandwidth Management for Single-Hop Ad Hoc Wireless Networks," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm.*, 2003.
[19] M. Kazantzidis, M. Gerla, and S.-J. Lee, "Permissible Throughput Network Feedback for Adaptive Multimedia in AODV MANETs," *IEEE Int'l Conf. Comm.*, 2001.
[20] K. Fall and K. Varadhan, "NS Notes and Documentation," *The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC*, 1997.
[21] G. Bianchi, "Performance Analysis of the IEEE 802. 11 Distributed Coordination Function," *IEEE J. Selected Areas in Comm.*, vol. 18, no. 3, 2000.
[22] Y. Yang, J. Wang, and R. Kravets, "Achievable Bandwidth Prediction in Multihop Wireless Networks," Technical Report UIUCDCS-R-2003-2367, Dec. 2003.
[23] P. Gupta and P.R. Kumar, "Capacity of Wireless Networks," *IEEE Trans. Information Theory*, no. 2, pp. 388-404, 2000.
[24] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *ACM SIGCOMM '94 Conf. Comm. Architectures, Protocols and Applications*, 1994.
[25] C. Perkins, "Ad-Hoc On-Demand Distance Vector Routing," *Proc. Military Comm. Conf.*, 1997.
[26] V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proc. INFOCOM*, pp. 1405-1413, 1997.
[27] O. Dousse, P. Thiran, and M. Hasler, "Connectivity in Ad-Hoc and Hybrid Networks," *Proc. INFOCOM*, pp. 1079-1088, June 2002.

Yaling Yang received the BS degree in telecommunications at the University of Electronic Science and Technology of China in 1999. She is currently a PhD candidate in computer science at the University of Illinois at Urbana-Champaign. Her research interests include resource management and QoS in wireless networks and network routing and congestion control. For more detailed information, please visit http://www.cen.uiuc.edu/~yyang8. She is a student member of the IEEE.

Robin Kravets received the PhD degree from the College of Computing, Georgia Institute of Technology, in 1999. She is currently an assistant professor with the Computer Science Department at the University of Illinois, Urbana-Champaign. She is the head of the Mobius group at UIUC, which researches communication issues in mobile and ad hoc networking, including power management, connectivity management, transport protocols, admission control, location management, routing, and security. Her research has been funded by various sources, including the US National Science Foundation and HP Labs. She actively participates in the mobile networking and computing community, both through organizing conferences and being on technical program committees. She is currently a member of the editorial board for *IEEE Transactions on Mobile Computing* and *Elsevier Ad Hoc Networks Journal* and was an associate editor of *MC2R: Mobile Computing and Communications Review*, a publication of ACM SIGMOBILE. She is also a member of the steering committee for WMCSA, the IEEE Workshop on Mobile Computing Systems & Applications. For a list of publications and more detailed information, please visit: http://www-sal.cs.uiuc.edu/~rhk. She is a member of the IEEE and the IEEE Computer Society.