# Distributed Maintenance of Resource Reservation Paths in Multihop 802.11 Networks

Emma Carlson[1], Christian Bettstetter[2], Holger Karl[1], Christian Prehofer[2], Adam Wolisz[1]

[1] Technische Universität Berlin, Telecommunication Networks Group (tkn.tu-berlin.de), Berlin, Germany

[2] DoCoMo Euro-Labs, Future Networking Lab (docomolab-euro.com), Munich, Germany

In Proc. (VTC Fall), Los Angeles, California, USA, September 2004

***Abstract*— Supporting applications with high quality-of-service requirements in wireless multihop networks is challenging, mainly due to uncertainties about the packet forwarding times introduced by the data link layer at every node on the path to the destination. For instance, networks based on IEEE 802.11 exhibit such variable forwarding times due to their CSMA (carrier sense multiple access) mechanism. In previous work, we have designed a fully distributed reservation protocol that introduces time slots into this CSMA mechanism. In the present paper, we analyze the performance of our protocol in scenarios where nodes randomly switch on and off. We show by simulation that our protocol achieves a lower average and less variable end-to-end delay.**

***Index Terms*— Ad hoc networks, medium access control, resource reservation, CS/TDMA, self-organization.**

## I. Introduction

THE medium access control (MAC) of IEEE 802.11 shows weak performance when being used in a wireless multihop scenario [1, 2]. This is especially true for real-time applications with demanding quality-of-service (QoS) requirements, in particular, if low and stable delay is needed. Here, 802.11 imposes uncertainties as to when a packet can be transmitted. These uncertainties sum up over multiple hops, resulting in large variations of the end-to-end delay.

This drawback has been our motivation to develop an extension to 802.11 which allocates radio resources for real-time traffic on the path between a source and destination node. Our approach is to reserve time slots on the MAC layer at each hop on the path. The reservation and maintenance of the time slots is performed in a completely distributed way. The basic functionality of our protocol, called *Distributed end-to-end Allocation of time slots for REal-time traffic* (DARE) has been presented in [3]. The paper at hand investigates its performance in more detail. In particular, it addresses the open issue of maintaining and repairing the reservation path in case of topology changes.

The remainder of this paper is organized as follows: Section II recalls the basic operation of DARE and lists related work. Section III discusses design choices for path maintenance and repair. Section IV presents a thorough simulation-based analysis of the DARE performance and compares the results with those of 802.11. Finally, Section V concludes and gives issues for further work.

## II. DARE Basics

The transmission path between a source and destination node is determined by the routing protocol. The task of the DARE protocol is to (a) reserve resources in the nodes along this path and assure that nodes located adjacent to the path do not interfere, and (b) handle the transmission of the real-time data.

The resource reservation is initiated by the source node. It sends out a *request-to-reserve* (RTR) message, which includes the requested duration and periodicity of a time slot (see Fig. 1). The RTR message propagates through the entire path via all intermediate nodes to the destination node. The message indicates to all these nodes how often and for how long they must be available for real-time transmissions. The destination node responds using a *clear-to-reserve* (CTR) message, which travels the same path back indicating to the intermediate nodes that the reservation request is accepted by all nodes. It is important that nodes adjacent to the reservation path also receive the RTR and CTR messages, thus getting informed about the reserved time slots. In essence, we make the CSMA-based approach of 802.11 aware of TDMA-inspired reserved time slots, while retaining CSMA for non-real-time data.

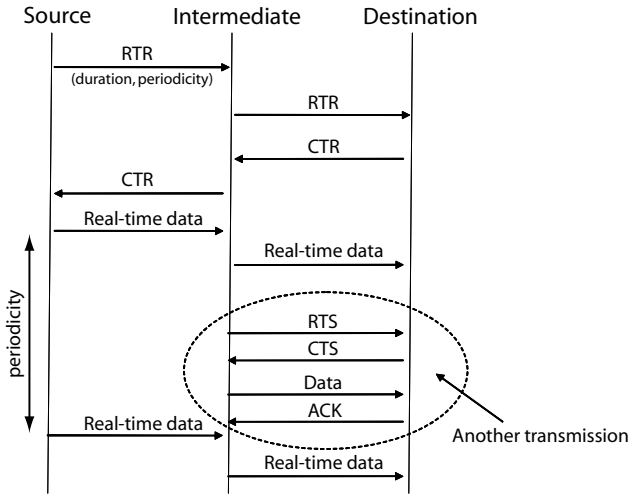



Fig. 1. DARE transmission flow. RTR and CTR reserve the resources at each node between the source and destination. Upon completion of the setup, the reserved real-time transmission can begin. Other transmissions can occur in between the reserved time-slots, using 802.11 DCF.

After completing the reservation setup, the source node transmits real-time data packets during the reserved time slots without performing medium access control, i.e., avoiding the distributed control function (DCF) of IEEE 802.11 [4]. Information regarding the reservation is spread piggy-backed onto control messages of the DARE protocol and in the header of the

real-time data messages. All nodes adjacent to the reservation path abstain from transmitting during the reserved time slots.

DARE does not retransmit lost real-time packets. The rationale behind this is that collisions are rare because the reservation information spreads to possibly interfering nodes "quickly enough." Furthermore, for static environments, DARE has been designed without acknowledgments, as we assume that packet losses due to noise are at an acceptable rate. We have shown in reference [3] that this functionality supports real-time transmissions in a static environment better than 802.11 DCF does. Further details of the protocol design can be found in [3].

We contend that such reservations have to be done at the MAC layer rather than at the network layer, as it is the MAC layer that introduces uncertainties. In fact, the actual reservations in wired networks are also done at a link scheduling level, not at the routing level. Obviously, the network layer's help has to be enlisted to determine the routes as such.

Several MAC reservation protocols apart from the DARE protocol exist for wireless multihop networks [5–11]. The main differences to and the motivation for the DARE protocol are the setup phase and how information is spread to surrounding nonparticipating nodes. The main lack in the related schemes is how they function in an environment where nodes switch on and off or are mobile, i.e., how they can repair a broken reservation path. The papers describing the above mentioned MAC reservation schemes do all assume a static scenario.

## III. Path Maintenance and Repair

If the network experiences topology changes, for example, because nodes switch on and off or move, the transmission path between source and destination may break. Clearly, such breaks must be repaired. If a node can no longer reach its subsequent node in the path, its data link layer will indicate the link break to the network layer. While the transmission route itself is then repaired by the routing protocol, i.e., an alternative route is found, we also need a distributed repair procedure for the reservations. In the following, we discuss (a) how nodes detect a break of the path, (b) how the reservation path can be repaired in a distributed manner, and (c) how outdated reservations are released.

### A. Detection of broken path

The basic requirement to initiate a path repair is that the node preceding the "hole" in the path must somehow notice the link break to the subsequent node in the path. This issue has not been considered in our previous DARE design [3].

One option is to use *explicit acknowledgments* (eACKs) that notify a node whether its message did or did not reach the next hop. Each time a node receives real-time traffic in the corresponding time slot, it returns an eACK to the preceding node. The eACK could be included in the allocated time slot or follow the standard 802.11 MAC procedure. As each hop is acknowledged, the signaling overhead increases (even if cumulative eACKs are used).

Another option is to use *negative acknowledgments* (nACKs), which are sent by nodes that do not receive any data in the allocated time slots. The advantage of this approach is its lower signaling overhead. The major drawback is that the information must reach the node preceding the "hole," which calls for higher transmission power, if possible at all.

A more elegant solution is provided by *implicit acknowledgments* (iACKs). If a node A has sent real-time traffic to its subsequent node B, A can overhear B's transmission to its successor C in the next time slot and can hence be sure that its transmission has been received by B. There is no signaling overhead for this approach. However, this solution may be unsuccessful if power control is used, e.g., if B uses a power so small that A cannot overhear the transmission. Furthermore, as the final destination node does not forward any message, the last hop of the path cannot be implicitly acknowledged.

In conclusion, as long as there is no power control, the best option seems to be a combination of iACKs and eACK: For each intermediate hop, up to the last hop, the transmission is acknowledged by overhearing the subsequent node forwarding the packet. Here, we assume that the channel is bidirectional, hence has the same characteristics in both directions between two nodes. In the last hop, an eACK is used. This eACK on the last hop also takes the function of informing potential interferers located adjacent to the destination node. As with the CTR message during reservation setup, we include information regarding the time slot duration and periodicity.

### B. Repair of broken path

We assume that an on-demand routing protocol, such as *ad hoc on-demand distance vector* (AODV), is employed in the ad hoc network. Once a path break is detected, the routing protocol is capable of repairing the route either locally or from the source node. The node that notices the route break (i.e., the node preceding the "hole") starts a route update procedure, where routing messages are exchanged in the network layer. Upon re-establishment of the route, the reservation has to be repaired as well, either locally or source-initiated.

If the routing protocol has performed a *local repair*, the best option for the DARE protocol seems to be that reservations are repaired locally as well. Some nodes that were part of the old route are still part of the new route, and neither the periodicity nor the time slot length have changed. Each node that is located in the communication range of the node initializing the repair is a potential candidate to be employed as a new relay node in the new reservation path. Each of these nodes has overheard the real-time transmission and is thus avoiding the reserved time slots. Thus, it is very likely that these nodes can allocate resources in the hitherto avoided time slots. If so, there is no need for a new reservation setup.

If the routing protocol has performed a *source-initiated repair*, however, possibly a completely different route has been established. In this case, the only option for the DARE protocol is to release the old reservations and set up a completely new reservation from the source. On the one hand, such a source-initiated repair always yields an optimized path. On the other hand, it might cause problems if nodes along the new route might not be able to fulfill the requested reservation.

During the repair process, we can give priority to the messages of the routing protocol, by transmitting them in the reserved time slots. This should accelerate the route repair pro-

cess. The real-time packets may be buffered in the nodes until the route is repaired.

### C. Reservation release

Nodes that are no longer part of the real-time path should release their allocated resources. This can be achieved by a timer and an optional *release* message. The timer is based on a certain number of unused reserved time slots. The *release* message is sent from the source node upon a source-initiated repair; it explicitly releases allocated time slots at nodes in the old path from the source to the "hole." In case of local repair, the release message is not used, as the time slots up to the "hole" must not be released. A possible extension uses release messages "backward" from the destination toward the other end of the hole.

### D. Summary

Nodes notice that the path is broken via iACKs from every relay node and an eACK from the final destination node. The routing protocol handles the route repair. For a source-initiated route repair on the network layer, DARE performs source-initiated reservation repair on the MAC layer. For a local route repair, it performs a local reservation repair as well. During the repair procedure, we give the messages of the routing protocol priority and buffer the real-time packets. Reservations that are no longer needed will time out after three unused time slots (initial simulations have shown that this a reasonable value); release messages are not used in this paper.

## IV. Simulation-Based Performance Analysis

This section analyzes the functionality of the enhanced DARE protocol and evaluates its performance by simulation. We implemented DARE into the simulation tool ns2 [12], use existing 802.11 functionality, and employ AODV for routing.

### A. Simple deterministic scenario without background traffic

We first perform a study using a simple scenario shown in Figure 2. Initially, there is a real-time path from Node S via A and B to D. Every $\Delta = 0.1$ s, Node S sends a packet of size $s = 512$ bytes. Thus, the required data rate on the path is $b_0 = s/\Delta = 41$ kbit/s. We use the basic channel of IEEE 802.11 with data rate $b = 1$ Mbit/s. Hence, at least a fraction $b_0/b = 4.1\,\%$ of the available bandwith must be reserved for the transmission. To fulfill this criterion, the duration of the reserved time slots must be $\tau > b_0/b \cdot \Delta = 4.1$ ms. We employ $\tau = 5$ ms in our simulations.

Upon outage of Node B, the route is repaired locally. Figure 3 shows the sequence of the packets received by D. Here, a single packet is lost, a single packet is delayed, and the path is reestablished after $t_r = 0.2$ s. Upon outage of Node A, the route is repaired from the source. Here, it takes a little longer to repair the path, resulting in the slight delay of an additional packet. These results assume that the AODV routing messages are given precedence over the actual data packets and are sent in the reserved time slots. This priority explains the delay of real-time packets (which are sent as soon as possible after the routing packets); the lost packet is due to a timeout, not due to a collision.

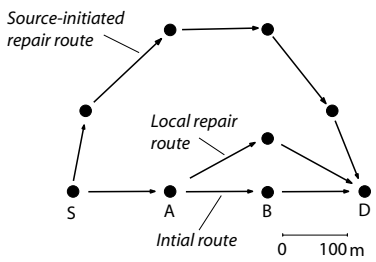



Fig. 2. Simulation scenario with a local or source-initiated repair.

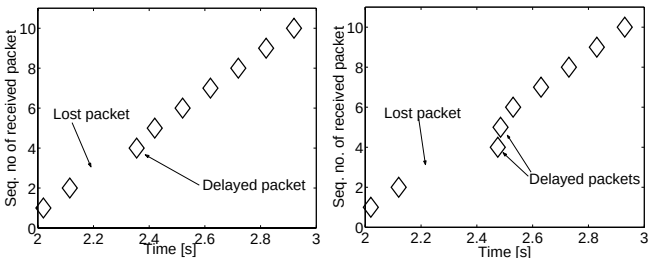



Fig. 3. Sequence of received packets using either local (left) or source-initiated (right) repair.

### B. Random scenario with background traffic

We now consider a scenario in which $n = 100$ nodes are placed randomly with the positions sampled from a uniform distribution on a square area of length $a = 700$ m. As the communication range of IEEE 802.11 is about $r_0 = 200$ m, this setup guarantees that the resulting network is connected with high probability [13], i.e., each node can communicate with each other node either via a direct link or via multihop routing.

One node is randomly chosen to act as the source for the real-time traffic. It transmits real-time packets with the same parameters as in the deterministic scenario. Another node is randomly chosen to act as the destination. All remaining $(n-2)$ nodes act as *non-real-time* (nrt) nodes to generate background traffic. They transmit packets of size $s = 512$ bytes with exponentially distributed inter-arrival times, using the standard 802.11 MAC protocol. All nrt packets are addressed to the same destination as the real-time traffic. This scenario can be interpreted in a way that the destination node is a gateway to the fixed network. The total load of the nrt traffic is at most $500$ kbit/s when all nrt nodes are transmitting simultaneously. We have chosen such a rather low total load because it is well-known that DCF has poor performance for high loads [1] and we have already shown in [3] that DARE outperforms DCF for the real-time traffic in saturated networks.

To study the impact of topology dynamics, we use the following model. Each nrt node switches off after some random time, which is sampled from a negative exponential distribution with a given expected value $E\{T_{\text{on}}\}$. It switches on again after another random time, sampled from the same distribution with the expected value $E\{T_{\text{off}}\}$, and so on. For simplicity, we set $E\{T_{\text{on}}\} = E\{T_{\text{off}}\}$ and call this parameter $\mu$.

The total simulation time of one scenario is $3600$ s. We simulate $50$ random scenarios and then take the average of the performance values. The same experiments are repeated using

conventional DCF.

*1) End-to-end packet delay:* First of all, we analyze the end-to-end delay of real-time packets. The simulation results for the average value of the delay are shown in Figure 4. The use of DARE considerably reduces the average delay compared to DCF. For both protocols, the average delay increases, as $\mu$ decreases. This is due to the fact that small $\mu$ causes frequent topology changes which induce more route update procedures.

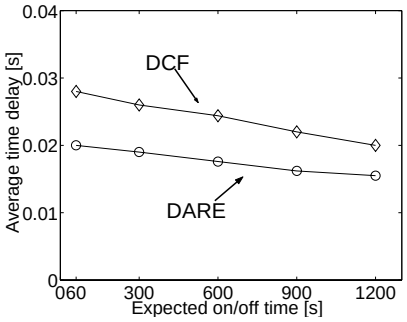
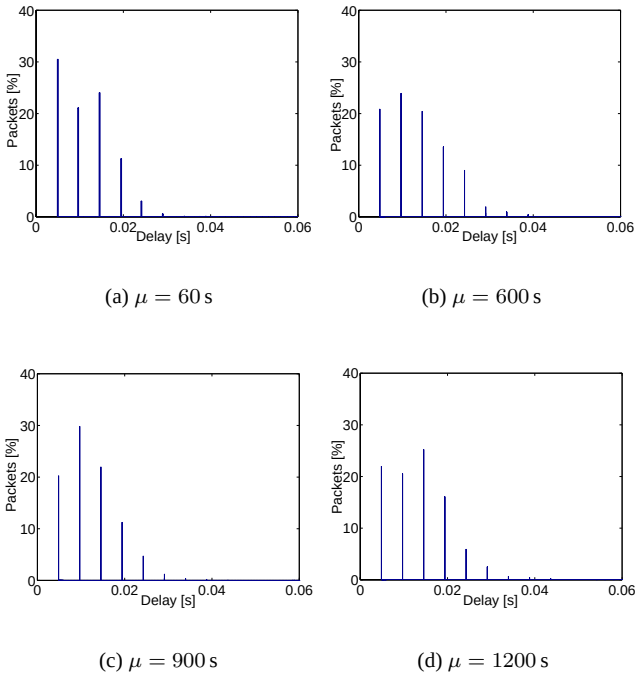



Fig. 4. Average end-to-end delay of real-time packets.

A deeper insight into the delay behavior can be obtained by studying the percentage at which certain end-to-end delay values occur (see histogram in Fig. 5). DARE produces a discrete, sharply separated set of equidistant delay values. The different values correspond to different path lengths between the source and destination node. If the source and destination nodes can communicate within one hop, the communication takes $4.8$ ms; if they are two hops apart, it takes $9.6$ ms, and so on. It seems that longer on/off periods cause higher delays to occur with higher probability.

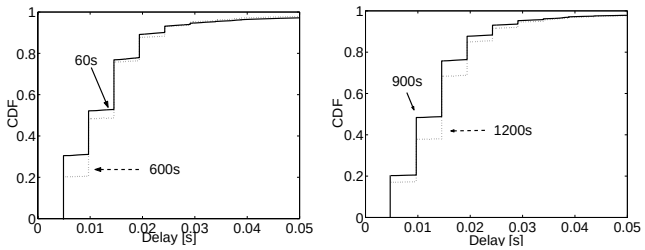



(a) $\mu = 60$ s (b) $\mu = 600$ s

(c) $\mu = 900$ s (d) $\mu = 1200$ s

Fig. 5. Histogram of end-to-end delay for real-time packets using DARE.

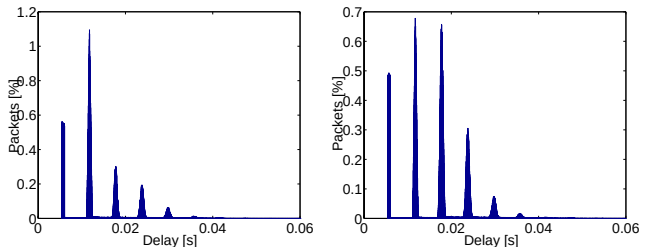
Figure 6 shows the corresponding cumulative distribution function (CDF) of the delay, defined as the percentage of received packets with a delay lower than a certain value. For example, more than $80\,\%$ of the packets experience a delay lower than $0.025$ s (for all $\mu$).




(a) $\mu = 60$ s, $600$ s (b) $\mu = 900$, $1200$ s

Fig. 6. CDF of end-to-end delay of real-time packets using DARE.

The delay histograms using DCF are shown in Figure 7. The histogram is not discrete, but the delay values are distributed around equidistant peaks. Again, the time value at which a peak occurs corresponds to the number of hops between source and destination. A one-hop communication takes about $5.5$ ms. The delay variation increases with the number of hops. For example, the peak for a four hop communication (at about $0.02$ s) is much wider than the peak for a two hop communication (at about $0.01$ s). The corresponding CDF of the delay is depicted in Figure 8, in comparison to that of DARE.



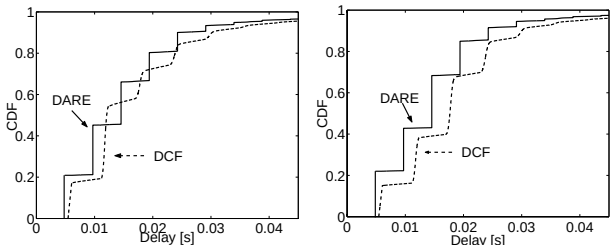

(a) $\mu = 600$ s (b) $\mu = 1200$ s

Fig. 7. Histogram of end-to-end delay of real-time packets using DCF.

In summary, the use of DARE for real-time traffic leads to better delay characteristics than the use of DCF: the end-to-end delay is on average lower and more stable.

*2) Packet loss and throughput:* Let us now study the packet loss rate and throughput of DARE and DCF. As shown in Figure 9 (left side), the average packet loss rate using the DARE protocol is higher than that using DCF. The difference is about $30\,\%$ at small $\mu$ and about $25\,\%$ at larger $\mu$.

The reason for this difference is as follows: Using DARE, the nodes of the reservation path perform no channel sensing but transmit immediately during the time slots. Some of the nrt nodes are located at such a distance to the reservation path that they are too far away to successfully receive the reservation in-




(a) $\mu = 600\,\mathrm{s}$ (b) $\mu = 1200\,\mathrm{s}$

Fig. 8. CDF of end-to-end delay of real-time packets.

formation, but still close enough to interfere the real-time transmission. This is due to the fact that the transmission range between two nodes is in general lower than the interference range between them. The nrt nodes cause losses of real-time packets whenever they transmit just during a reserved time slot. These types of losses do not occur using DCF for real-time transmission, since here each node senses the channel before trying to transmit and backs off in case the channel is busy. In addition, if a collision would occur with DCF (if nodes start a transmission exactly at the same time), the packet is retransmitted, which improves the packet loss rate but increases the end-to-end delay as discussed above. One possibility to reduce the packet loss rate for DARE is to spread the reservation information to an increased distance around each sender.

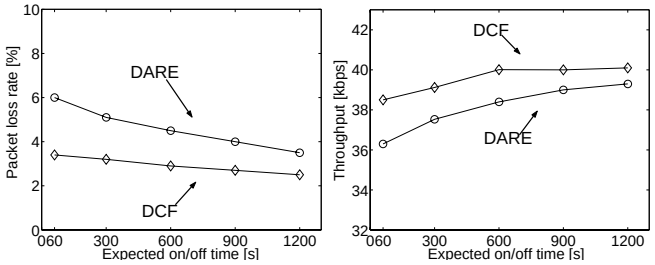



Fig. 9. Average packet loss rate (left) and throughput (right).

In both protocols, the impact of $\mu$ on the packet loss rate is as follows: As $\mu$ increases, i.e., the nodes switch on and off less frequently, less real-time packets get lost. One reason for this is that high $\mu$ causes less topology changes. Another reason is that the nodes perform a route search procedure every time they switch on. As the routing messages have priority over the reserved real-time transmission, the nrt nodes transmit during the allocated time slots, causing interference-based packet losses.

Figure 9 (right side) shows the throughput of both protocols. DARE has a slightly lower throughput than DCF, approximately $5\,\%$ lower for short on/off periods, and $2.5\,\%$ lower for long on/off periods.

## V. Conclusions

This paper presented an improved version of DARE — a distributed MAC protocol for 802.11-based wireless multihop networks that supports QoS for real-time data flows by reserving dedicated time slots. We designed a mechanism to maintain a reservation path, if the topology is changing. To save radio resources, the mechanism exploits implicit acknowledgments on all intermediate hops. We studied the performance with respect to end-to-end delay, packet loss rate, and throughput in scenarios where node continuously switch on and off, thus reservation paths are repeatedly broken.

The application of DARE results in a major benefit compared to conventional 802.11: the average end-to-end delay of real-time packets is lower and packets have a clearly determined delay. DARE is superior in high load situations and in low load/ mobility cases as investigated here. This advantage is bought at the price of slightly lower throughput and a somewhat higher packet loss rate. Both drawbacks are related to the fact that DARE uses no sensing mechanism during real-time data transmission. All in all, DARE provides better support for real-time data flows, where low and predictable delay are important parameters.

In future work, we intend to improve the DARE protocol further to reduce packet losses. Additional work is about handling multiple reservations, analyzing the performance with mobile nodes, using real-time traffic models, and comparing DARE with 802.11e [14].

## References

[1] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop ad hoc networks?," *IEEE Commun. Mag.*, June 2001.
[2] K. Sundaresan, H.-Y. Hsieh, and R. Sivakumar, "IEEE 802.11 over multi-hop wireless networks: problems and new perspectives," *Elsevier Ad Hoc Networks*, Apr. 2004.
[3] E. Carlson, H. Karl, A. Wolisz, and C. Prehofer, "Distributed allocation of time slots for real-time traffic in a wireless multi-hop network," in *Proc. European Wireless*, Feb. 2004.
[4] "IEEE standard 802.11, wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 1999.
[5] G. Hiertz, J. Habetha, P. May, E. Weiß, R. Bagul, and S. Mangold, "A decentralized reservation scheme for IEEE 802.11 ad hoc networks," in *Proc. IEEE Intern. Symp. on Personal, Indoor, and Mobile Radio Commun. (PIMRC)*, Sept. 2003.
[6] S. Jiang, J. Rao, D. He, X. Ling, and C. C. Ko, "A simple distributed PRMA for MANETs," *IEEE Trans. Veh. Technol.*, Mar. 2002.
[7] S.-T. Sheu, T.-F. Sheu, C.-C. Wu, and J.-Y. Luo, "Design and implementation of a reservation-based MAC protocol for voice/data over IEEE 802.11 ad hoc wireless networks," in *Proc. IEEE Intern. Conf. on Commun. (ICC)*, 2001.
[8] J. L. Sobrinho and A. S. Krishnakumar, "Quality of service in ad hoc carrier sense multiple access wireless networs," *IEEE J. Select. Areas Commun.*, 1999.
[9] C. H. R. Lin and M. Gerla, "Asynchronous multimedia multihop wireless networks," in *Proc. IEEE Infocom*, Apr. 1997.
[10] J. L. Sobrinho and A. S. Krishnakumar, "Real-time traffic over the IEEE 802.11 medium access control layer," *Bell Labs Technical Journal*, 1996.
[11] S.-T. Sheu and T.-F. Sheu, "DBASE: A distributed bandwidth allocation/sharing/extension protocol for multimedia over IEEE 802.11 ad hoc wireless LANs," in *Proc. IEEE Infocom*, 2001.
[12] "Network simulator 2." http://www.isi.edu/nsnam/ns/.
[13] C. Bettstetter, "On the connectivity of wireless multihop networks with homogeneous and inhomogeneous range assignment," in *Proc. IEEE Vehicular Techn. Conf. (VTC)*, Sept. 2002.
[14] R. Rollet and C. Mangin, "IEEE 802.11a, 802.11e and HiperLAN/2 goodput performance comparison in real radio conditions," in *Proc. IEEE Globecom*, 2003.