Distributed allocation of time slots for real-time traffic in a wireless multi-hop network[1]

Emma Carlson, Holger Karl, Adam Wolisz
TU Berlin, Einsteinufer 25, 10587 Berlin, Germany
{carlson|karl|wolisz}@tkn.tu-berlin.de
fon: +49.30.31429243, fax: +49.30.31423818

Christian Prehofer
DoCoMo Euro-Labs, prehofer@docomolab-euro.com

Abstract: **This paper examines a new mechanism for distributed resource reservation that offers support for applications with QoS requirements in a wireless multi-hop network. This mechanism is based on IEEE 802.11 DCF and includes end-to-end reservations of time slots and distribution of reservation information to mobile nodes unaware of the reservation, piggy-backed on existing medium access control messages.**

We investigate average packet delay, packet delay distribution, packet loss rates and throughput of our scheme compared to IEEE 802.11 DCF using simulations. Our scheme results in a better support for real-time transmissions than DCF.

1. Introduction

Multi-hopping has been widely advocated as a means to improve the operation of wireless communication systems. But with respect to real-time applications that have stringent Quality-of-Service (QoS) requirements for their transmissions, there are still some unsolved issues (e.g., [3]). These applications require, among other things, a low and stable time delay of packets when transmitted over a multi-hop network as packet delay jitter reduces the user-experienced quality.

A major source of delay variation is the employed wireless transmission system. The distributed Wireless Local Area Network (WLAN) standard IEEE 802.11 is well known, usually used with a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) based, distributed medium access control scheme, the Distributed Coordination Function (DCF) [2]. This standard works well for normal data that has no stringent demands on the time delay, but for real-time applications, the DCF has the disadvantage that it introduces, at every hop of a multi-hopping path, uncertainties as to when a packet can be transmitted. These medium access uncertainties add up to variations in the total delay; also, the absolute delay of a packet can be large due to repeated medium access procedures. Hence, to support real-time applications that are sensitive to both the absolute value and to variations of the packet delay, medium access must be efficient and its uncertainties must be removed. One way of doing this is to allow stations to reserve future resources, i.e. periodic time slots, for their real-time packet transmissions, akin to the case of connection setup in wired networks. This way, real-time transmissions can obtain preference over plain data transmissions and can actually receive guaranteed delay (as far as this is possible in wireless networks).

To perform resource reservation in a centrally controlled network, it suffices to somehow reserve the time slots along the multi-hop path and all possibly conflicting nodes would know about the reservation. Setting up such a reserved multi-hop “virtual connection” in a system with distributed medium access control is more difficult, but can still be done by means of reservation messages. But as nodes move around (or are switched on), they can move into the region of an existing reservation without being aware of it – a case quite different from connections in wired networks. Hence, the reserved “virtual connection” has to be *maintained* as well, requiring mechanisms that allow mobile nodes to learn about existing reservations. In addition, mobility entails breakage of reserved paths, which also requires maintenance. This maintenance has to be light-weight and should extend existing medium access procedures – a need for connection management mechanisms that is absent in wired networks and in this form not yet discussed for wireless networks either.

This paper presents such an extension to IEEE 802.11 that enables a simple end-to-end set-up of periodically reserved time slots over a multi-hopping path and that provides maintenance means to inform oblivious nodes of a reserved transmission with only a small error rate. The protocol is called Distributed end-to-end Allocation of time slots for REal-time traffic (DARE).

This paper is organized as follows: Previous work in the area is described in Section 2, followed by a brief summary of the IEEE 802.11 DCF in Section 3 and a description of our protocol extensions in Section 4. Section 5 describes the evaluation model and in Section 6, the first evaluation results of the DARE mechanism are described. Section 7 concludes this paper and gives suggestions for further work.

2. Related work

Quality-of-service support for wireless networks with distributed control has been discussed by several authors, e.g. [4, 13, 8, 6, 5]; only some of this work is discussed here. In [11, 12], transmission of jamming bursts is used. A station that wants to transmit starts its access proce-

[1]This work has been partially supported by DoCoMo Euro-Labs.

dure by jamming the channel with an energy burst with a length proportional to the time a station has waited for transmission. The one with the longest "Blackburst" transmits first. Whenever a packet is transmitted, the node schedules the transmission for the next packet after a predefined period, which is the same for all nodes and all packets.

In [7], GERLA and LIN describe a periodical packet reservation technique, Multiple Access Collision Avoidance with Piggyback Reservations (MACA/PR), which uses CSMA/CA as MAC layer and a bandwidth reservation technique at the network layer (IP). MACA/PR as a MAC protocol sets up a real-time connection only over a single hop. All nodes have Reservation Tables (RT) to see when and who is transmitting. The first packet in a stream reserves a transmission window for following packets with a certain periodicity. Whenever a new node enters the network, it first listens to the channel to hear ongoing transmissions. This is followed by an exchange of RTs in the system.

References [10, 9] describe MAC protocols that use broadcast messages, so called Reservation Frames (RF), to distribute reservation information. These messages are sent periodically and inform nodes about ongoing reservations. Each station keeps a reservation table with information about ongoing reserved transmissions. A voice station that wants to transmit listens for a free time slot during one cycle and asks for it if found.

All these mechanisms essentially reserve on a hop-per-hop basis, that is, each hop independently from preceding hop(s). If an intermediate node cannot set up a reservation for the next hop, the medium access procedure must be performed for each packet separately at this node, hence no guarantees for a bounded end-to-end time delay exists. Therefore, we suggest to use an end-to-end reservation setup.

The distribution of reservation information is handled by explicit exchange of reservation tables or broadcasted reservation frames. This demands a separate medium access procedure, which can cause overhead. Another option would be to piggyback the information onto existing protocol messages. This would not require a new medium access procedure, hence seems as the best option. Therefore, it is our suggestion to handle the reservation information exchange only with piggybacking on existing messages.

3. IEEE 802.11 DCF

A node that wants to transmit a packet starts the access procedure by sensing the channel. If it is idle for more than a DCF Inter Frame Space (DIFS), the node can transmit its packet. If the channel is busy, the node must defer until the medium has been idle for DIFS and then must perform a back-off procedure [2]. When the back-off timer reaches zero, the transmission of a packet can start. If two back-off timers reach zero at the same time, a collision can occur. There is an option to use a handshake mechanism where Request-To-Send (RTS) and Clear-To-Send (CTS) messages are exchanged between the transmitter and receiver before the transmission of data, allocating the channel for the data packet transmission. This exchange is useful when the data packet is large since a collision and retransmission of such a packet take up more bandwidth than with a collision and retransmission of an RTS. After the data packet is transmitted, the receiver must first wait for a Short Inter Frame Space (SIFS) and can thereafter transmit an acknowledgement (ACK) to the transmitter. The SIFS is also used by the receiver before transmitting the CTS.

Each node keeps a Network Allocation Vector (NAV) that records the duration that a node must defer before starting the medium access procedure. The NAV is updated when a node hears a packet with the duration field of the header [2].

4. DARE protocol

DARE enables a node to set-up a periodic reservation for real-time flows over several hops.[2] The length and periodicity of the reservation can be chosen according to the needs of the application. Further, DARE distributes information about the reservation by piggy-backing the information on other messages.

By virtue of the periodic reservation, real-time packets are directly transmitted without any medium access procedure (RTS/CTS exchanges); also, no acknowledgments or retransmissions are used. Non-real-time packets are transmitted using the standard DCF mechanisms, with RTS/CTS exchange. The intuition is that reservation state can be spread sufficiently quickly such that collision-induced packet losses will be rare enough to be negligible.

4.1 Reservation set-up and maintenance

First, a request-to-reserve (RTR) message is sent, end-to-end. Intermediate nodes read the message and forward it when they can make the requested reservation. The destination node responds with a clear-to-reserve (CTR) message, which travels the same way back.[3] When the source node receives the CTR message, it starts the transmission of real-time packets at the next reserved interval. All nodes have a time-table that informs them when they should be available or silent. This table is updated for nodes part of the reserved real-time transmission as they receive the RTR and for all other nodes as they overhear an RTR, CTR, or a real-time packet, which have additional reservation information in their headers.

There is a possibility for an intermediate node to shift the reserved slot in time if it cannot process the request right away due to another reserved transmission. Information about this time shift can be included in the CTR so the source node can decide whether the total delay is acceptable or not.

The release of the reservation can be based on timers (no packet transmitted for a certain time) or explicit release messages (the application that made the reservation terminates). If a reserved path breaks, this will also be detected by timeouts and the path can be repaired locally

[2]Determining multi-hop routes is left to the routing protocol and not considered here.

[3]We assume a routing protocol that provides symmetric routes.

or the source can be informed to take remedial actions or to abort the call.

4.2 Distribution of Reservation Information

4.2.1 When to distribute

A station not involved in a reservation can learn about the reservation by overhearing a real-time transmission, either a real-time data packet or the set-up messages. Another option is to include information in packets not part of the reservation itself: Suppose a node moves into radio range of nodes that have an active reservation and wants to transmit a packet. When it transmits an RTS to a node involved in a reservation or otherwise aware of a reservation, the receiving node can include information about reservations in the CTS. Further, if a node part of the reservation wants to transmit packets to other nodes not involved in the reservation, it could be useful to include information in the RTS, Data or ACK.

By piggy-backing reservation information onto existing protocol messages that have to be exchanged anyway, DARE succeeds in spreading reservation information to so-far oblivious nodes, which can then abstain from sending in reserved time slots. This quick and efficient, local spreading of reservation information is the ultimate reason why real-time packets can be transmitted without any medium access procedure at all: the residual error rate caused by uninformed nodes is marginal. Evidently, there is a trade-off between additional overhead and required information that has to be characterized; for the first evaluation of the DARE protocol presented in Section 6, only reservation information in the set-up and data packets is used; the improvements due to piggy-backing of reservation onto RTS/CTS messages will be investigated in further work.

4.2.2 What to distribute

In addition to the question *which packets* distribute information to nodes that do not belong to a reserved path, it is also important to judiciously determine *how long* such a message reserves the medium. Consider Figure 1 as an example: There is a reserved path set up from A over B and C to a base station BS. Node D is located such that it can sense a transmission from B, but it cannot correctly receive transmissions from B. As node D transmits, it will generate interference at surrounding nodes up to a certain range, in Figure 1 named interference zone. Here, node B is situated within this interference zone. Further, D can communicate with C, hence D can retrieve information from the real-time packets transmitted by C about the reserved time slot at C. If D now abstains from transmitting in the time slots where C sends or receives, it could still interfere with B's reception of A's packets.

This means that in this scenario, node D must avoid the time slot from the starting time of reception at node B to the end of reception at the BS. In fact, simulations have shown that only communicating to D the transmission information of node C (its send and receive time slots) results in sub-optimal performance. Therefore, we include information about the reserved time slot at the

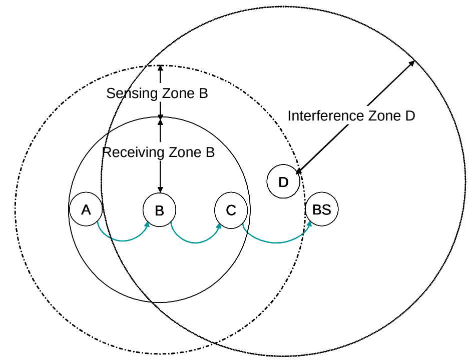

Figure 1: Reception, sensing and interference zones of a transmission. Node B is exposed for interference from the transmission of node D.

preceding node (in this example, the sending and receiving of node B) in the reservation information of packet headers as well. As this example (and the evaluations in Section 6) show, there are cases where this extended information is crucial. There will also be cases where this extension can hamper system capacity as some time slots can be needlessly blocked. We will investigate more sophisticated rules that adapt themselves to the actual distances and transmission power between different nodes in the future.

5. Evaluation model

DARE is investigated in a single-cell scenario with one base station. Multiple wireless stations surround this base station. We only consider uplink traffic to the base station here (no traffic between terminals expect forwarding).

One three-hop real-time transmission path to the base-station, reserved by DARE, exists. The reservation is 5 milliseconds long, repeated every 100 milliseconds. No other traffic is going through the nodes involved in this transmission. The real-time source node transmits Constant Bit Rate (CBR) generated packets during the 5 millisecond reserved time slot, else nothing. The reservation neither breaks nor is it released. Further, we have non-real-time stations transmitting packets with an exponentially distributed inter-arrival time and constant size (a Poisson process) directly to the base station. We consider two models. In model 1, one up to eight non-real-time stationary stations, and in model 2, one station that moves in from a long distance at a speed of 2.5 m/s to the area with the reserved real-time transmission (Figure 2).

For model 1, each non-real-time station has a load of 100 kbps. For model 2, the load of the non-real-time stations is varied, 100 kbps–800 kbps. For both real-time and non-real-time traffic, the packet size is 512 bytes.

The comparison case is IEEE 802.11 DCF with RTS/CTS. AODV is used as a routing protocol (which does lead to real-time packet losses during route setup at the simulation start; these initial transients are ignored in the following results).

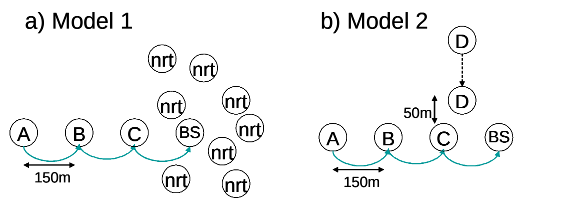

Figure 2: In a) Model 1 with up to eight non-real-time stations (nrt) and in b) Model 2 with one non-real-time station, node D that moves in on reserved transmission A to D with a speed of 2.5 m/s.

6. Performance results

We investigated the DARE scheme using simulations with the network simulator NS-2 [1]. Each parameter combination is simulated 5 times, each time for 1000 seconds simulated time. The channel bandwidth is 1 Mbps, which partially is reserved for real-time traffic. We compare throughput, packet time delay and packet loss rates for DARE and DCF.

The throughput results in Figure 3 and Figure 4 show that DARE offers the same real-time throughput, irrespective of the non-real-time traffic load. DCF achieves for real-time traffic under high loads in model 1 only one tenth of the DARE throughput. For the highest non-real-time load case, the system in model 1 is saturated. For model 2, the value is better. The high load case of model 2 is not saturated as only one node performs medium access according to the DCF.

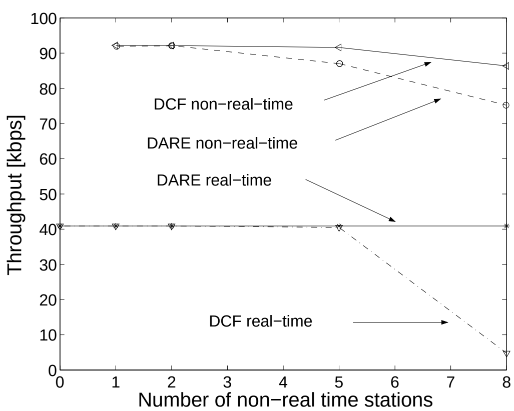

Figure 3: Throughput per node for real-time and non-real-time traffic, model 1.

For non-real-time traffic, DARE provides, at a larger number of terminals, a reduced throughput compared to standard DCF. There is a trade-off for the non-real-time traffic with reservation mechanisms for real-time traffic.

But more important for real-time traffic are the time delay results. Figure 5 shows histograms of packet delays for both DCF and DARE from model 1 (for 5 non-real-time stations). The distribution of these packet delays is considerably narrower (constant 0.0145 seconds) for DARE than DCF, hence less jitter, which better supports real-time transmissions.

The average time delay for real-time packets is shown in Figure 6 and Figure 7. The average real-time packet

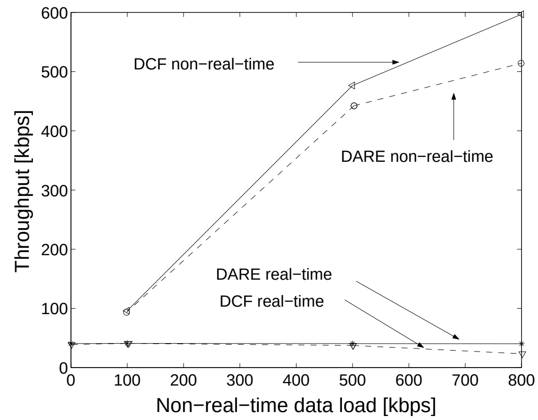

Figure 4: Throughput per node for real-time and non-real-time traffic, model 2.

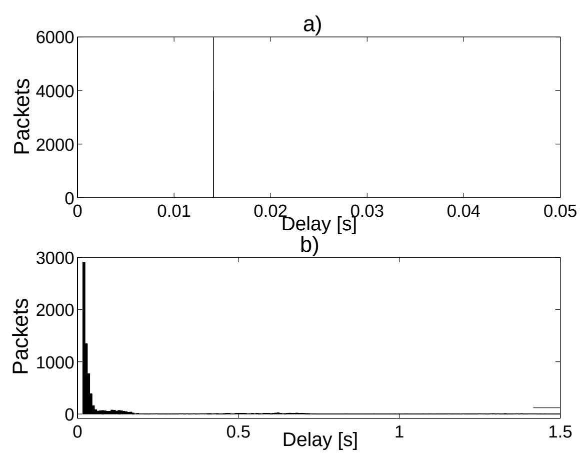

Figure 5: Real-time packet time delay distribution for model 1, 5 non-real-time stations in a) DARE and b) DCF

(note the different scale of the x and y axis).

delay is constant for DARE in both models, 0.0145 seconds (three hops, each approximately 0.0048 seconds transmission time). The average real-time packet delay for DCF is larger. In model 1, for 8 non-real-time stations, the average real-time packet delay is 1000 times larger than with DARE, 1.65 seconds. For model 2 and 800 kbps non-real-time load, the average real-time packet delay for DCF is 10 times larger. The difference between these two models is the number of access procedures, which in model 1 introduces a larger packet delay. This is also described earlier for the throughput results.

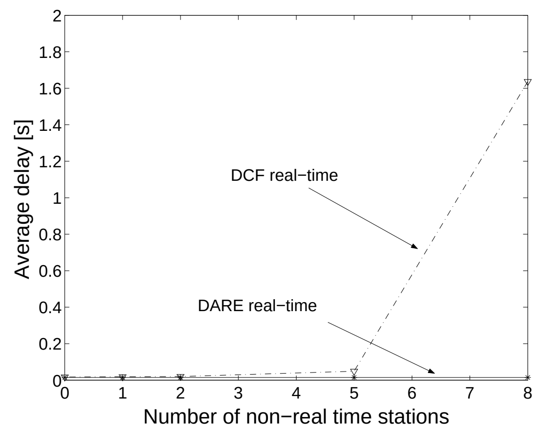

Figure 6: Average real-time packet time delay vs. non-real-time station load in model 1.

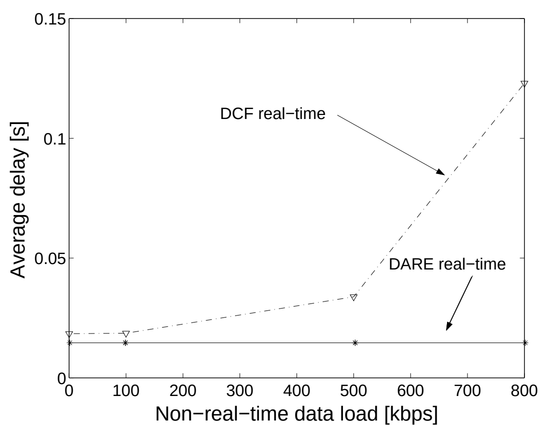

Figure 7: Average real-time packet time delay vs. number of non-real-time stations in model 2

The delay results have to be considered together with the packet loss rates, as one main claim was that oblivious nodes can be informed efficiently about existing reservations. The packet loss rates for model 1 in Figure 8 show that the real-time packet loss rate using the DARE is indeed 0% for all numbers of non-real-time stations. It is higher for DCF, which has more collisions and MAC time-outs that generate packet losses. For the 8 non-real-time station scenario, the real-time packet loss rate for the DCF is approximately 11%.

For model 2, the packet loss rate can be seen in Figure 9. The packet loss rate for DARE is 0% for both non-real-time and real-time transmissions. For the DCF, this rate is for real-time traffic up to 2%. This figure shows the loss rates after the non-real-time station has overheard a real-time packet and set its timer for avoiding the reservation. Before this happens, some real-time packet losses occur, which is up to 1.5% for the maximum non-real-time load.

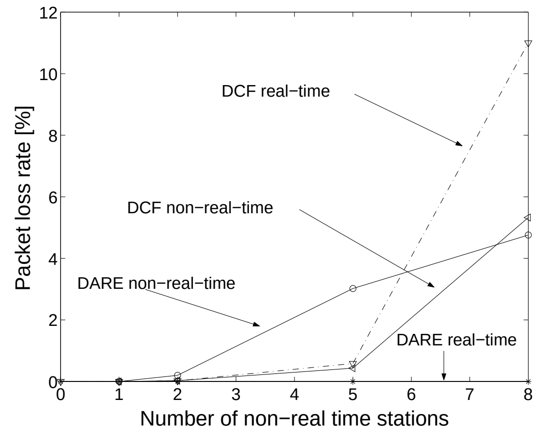

Figure 8: MAC packet loss rate vs. number of non-real-time nodes in model 1.

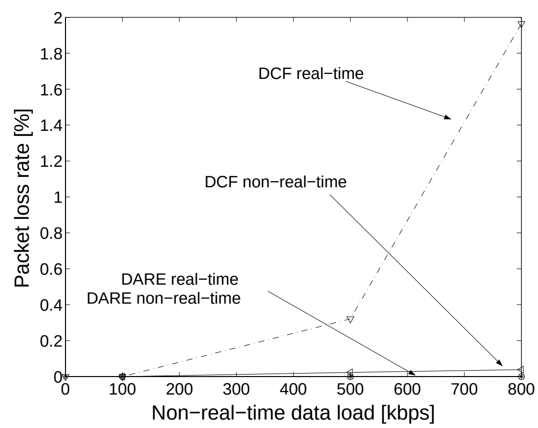

Figure 9: MAC packet loss rate vs. non-real-time load for model 2.

The above described packet loss rates justify the previous described throughput results.

7. Conclusions

In this paper we presented a new multi-hop medium access procedure for applications with real-time packet transmissions. The mechanism includes an end-to-end periodic time reservation for real-time flows and a piggyback mechanism for distributing information about the reservation to nodes not aware of the reservation. DARE offers better real-time transmission support than DCF and can efficiently distribute reservation information. We have looked at throughput, average time delay, packet time delay distribution and packet losses.

For further investigations, DARE will be investigated in larger scenarios with more reserved paths and mobility among non-real-time and real-time stations. Further, we will investigate more sophisticated rules for the non-real-time stations to determine how long they must abstain from transmission, which depends on distance and transmit power.

Further conceptual work consists of including information about the reservation in other messages, mainly RTS and CTS of non-real-time transmissions. We will also investigate the usage of negative acknowledgements when setting up the reserved path. We will include the possibility for intermediate nodes to shift reservation in time if they cannot immediately reserve a time slot. This information will be carried back to the source in the CTR message. Also, we want to investigate the possibility of duplex path set-up.

REFERENCES

[1] Network simulator 2. http://www.isi.edu/nsnam/ns/.

[2] IEEE Standard 802.11. "Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specifications". 1999.

[3] M. Gerharz, C. de Waal, M. Frank, and P. James. "A Practical View on Quality-of-Service Support in Wireless Ad Hoc Networks". In *Proc. of the 3rd IEEE Workshop on Applications and Services in Wireless Networks (ASWN)*, July 2003.

[4] S. Jiang, J. Rao, D. He, X. Ling, and C. C. Ko. "A Simple Distributed PRMA for MANETs". In *IEEE Transactions on Vehicular Technology, Vol. 51, no 2*, March 2002.

[5] S. Koutroubinas, T. Antonakopoulos, and V. Makios. "A new Efficient Access Protocol for Integrating Multimedia Services in the Home Environment". In *Proc. IEEE Transactions on Consumer Electronics*, pages 481–487, Aug 1999.

[6] S. Lal and E. Sousa. "Distributed Resource Allocation for DS-CDMA Based Multi-media Wireless LANs". In *Proc. IEEE Military Communications Conference (MILCOM)*, pages 583–588, Oct 1998.

[7] C.H.R. Lin and M. Gerla. "Asynchronous Multimedia Multihop Wireless Networks". In *Proc. Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM)*, April 1997.

[8] M.K. Marina, G.D. Kondylis, and U.C. Kozat. "RBRP: a Robust Broadcast Reservation Protocol for Mobile Ad Hoc Networks". In *Proc. IEEE Intl. Conf. on Communications (ICC)*, pages 878–885, June 2001.

[9] 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. Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM*, pages 1558–1567, 2001.

[10] S.-T. Sheu, T.-F. Sheu, C.-C. Wu, and J.-Y. Luo. "Desing and Implementation of a Reservation-based MAC Protocol for Voice/Data over IEEE 802.11 Ad Hoc Wireless Networks". In *Proc. IEEE Intl. Conf. on Communications (ICC)*, pages 1935–1939, 2001.

[11] J. L. Sobrinho and A. S. Krishnakumar. "Quality of Service in Ad Hoc Carrier Sense Multiple Access Wireless Networs". *IEEE J. on Selected Areas in Communications*, 17(8):1353–1368, 1999.

[12] J.L. Sobrinho and A.S. Krishnakumar. "Real-time Traffic over the IEEE 802.11 Medium Access Control Layer". In *Bell Labs Technical Journal*, pages 172–187, 1996.

[13] Z. Ying, A.L. Ananda, and L. Jacob. "A QoS Enabled MAC Protocol for Multi-Hop Ad Hoc Wireless Networks". In *Proc. IEEE Intl. Conf. on Performance, Computing, and Communications*, pages 149–156, 2003.