

# Sticky CSMA/CA: Implicit synchronization and real-time QoS in mesh networks

Sumit Singh <sup>a,\*</sup>, Prashanth Aravinda Kumar Acharya <sup>b</sup>, Upamanyu Madhow <sup>a</sup>,  
Elizabeth M. Belding-Royer <sup>b</sup>

<sup>a</sup> Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106, United States

<sup>b</sup> Department of Computer Science, University of California, Santa Barbara, CA 93106, United States

Received 31 March 2006; received in revised form 24 October 2006; accepted 9 December 2006

Available online 14 January 2007

## Abstract

We propose a novel approach to QoS for real-time traffic over wireless mesh networks, in which application layer characteristics are exploited or shaped in the design of medium access control. Specifically, we consider the problem of efficiently supporting a mix of Voice over IP (VoIP) and delay-insensitive traffic, assuming a narrowband physical layer with CSMA/CA capabilities. The VoIP call carrying capacity of wireless mesh networks based on classical CSMA/CA (e.g., the IEEE 802.11 standard) is low compared to the raw available bandwidth, due to lack of bandwidth and delay guarantees. Time Division Multiplexing (TDM) could potentially provide such guarantees, but it requires fine-grained network-wide synchronization and scheduling, which are difficult to implement. In this paper, we introduce Sticky CSMA/CA, a new medium access mechanism that provides TDM-like performance to real-time flows without requiring explicit synchronization. We exploit the natural periodicity of VoIP flows to obtain implicit synchronization and multiplexing gains. Nodes monitor the medium using the standard CSMA/CA mechanism, except that they remember the recent history of activity in the medium. A newly arriving VoIP flow uses this information to grab the medium at the first available opportunity, and then *sticks* to a periodic schedule, providing delay and bandwidth guarantees. Delay-insensitive traffic fills the gaps left by the real-time flows using novel contention mechanisms to ensure efficient use of the leftover bandwidth. Large gains over IEEE 802.11 networks are demonstrated in terms of increased voice call carrying capacity (more than 100% in some cases). We briefly discuss extensions of these ideas to a broader class of real-time applications, in which artificially imposing periodicity (or some other form of regularity) at the application layer can lead to significant enhancements of QoS due to improved medium access.

© 2007 Elsevier B.V. All rights reserved.

**Keywords:** Mesh networks; Quality of service; Medium access control; Voice over IP; Time division multiplexing

## 1. Introduction

Wireless mesh networks offer an attractive solution for providing voice, multimedia, and best effort data services in areas where deployment of a wired infrastructure is not viable or is economically

\* Corresponding author. Tel.: +1 805 893 7520.

E-mail addresses: [sumit@ece.ucsb.edu](mailto:sumit@ece.ucsb.edu) (S. Singh), [acharya@cs.ucsb.edu](mailto:acharya@cs.ucsb.edu) (P.A.K. Acharya), [madhow@ece.ucsb.edu](mailto:madhow@ece.ucsb.edu) (U. Madhow), [ebelding@cs.ucsb.edu](mailto:ebelding@cs.ucsb.edu) (E.M. Belding-Royer).

unattractive. Such networks can extend the reach of the wired backbone through a backhaul mesh of wireless routers, not only providing communication services to remote areas but also offering a flexible and economical means of scaling network capacity to serve increasing demand in enterprises and campus communities. Mesh networks can also provide a self-contained communication infrastructure that can be deployed in a plug-and-play fashion for search, rescue and disaster recovery operations. However, a key drawback of current wireless mesh technology based on IEEE 802.11 standards is that it is not possible to provide bandwidth and delay guarantees to real-time applications such as voice and video [1]. The primary reason is the variability in delay and loss performance that results from the requirement that each packet must contend afresh for the medium. These drawbacks of IEEE 802.11 are only partly alleviated by its QoS extension IEEE 802.11e. Given that medium access control is the bottleneck for providing QoS, an alternative approach is to employ Time Division Multiplexing (TDM) to obtain a collision-free transmission schedule. However, this requires fine-grained network-wide synchronization and scheduling, which are difficult to implement. In this paper, we introduce Sticky CSMA/CA,<sup>1</sup> a new medium access control mechanism that provides TDM-like performance to real-time flows without requiring explicit synchronization.

Our starting point is the observation that the traffic generated by real-time flows exhibits a significant regularity. Can this regularity be exploited for more effective medium access control? After all, contention between nodes occurs because a node cannot predict the actions of its neighbors. If the transmission schedule for a node follows a regular pattern, its neighbors should be able to learn this pattern using the CSMA mechanism, and thus avoid contention. While these ideas are potentially applicable to a broad class of real-time flows, in this paper, we focus specifically on illustrating how they enable efficient support of Voice over IP (VoIP) over wireless mesh networks. The success of VoIP over the wireline Internet leads us to expect that it will be a key application driving the wider accep-

tance and commercial success of wireless mesh networks. A broad range of wireless VoIP solutions are commercially available and are being deployed in enterprises and campuses [3,4]. However, these products typically optimize VoIP performance over a single wireless hop from a tetherless node to an Access Point connected to the wired network. Our goal is to support VoIP in wireless mesh networks, thus significantly increasing the flexibility of deployment. The challenge is to provide an adequate level of QoS, in terms of assured bandwidth, and bounded delay and delay jitter.

Sticky CSMA/CA is based on the assumption that all the real-time flows in the network are either naturally periodic (e.g., VoIP) or have been shaped by the higher layers as periodic (constant bit rate) streams with the same period. A node that has to transmit packets from a real-time flow monitors the medium using the standard CSMA mechanism. When it detects an opportunity to transmit, it attempts packet transmission, and on being successful, *sticks* to a periodic schedule. Neighbors detect such periodic schedules using the CSMA mechanism and avoid interfering with them. This medium access approach leads to a TDM-like sharing of the medium among the real-time flows, with the periodicity of the application and transmission schedule being used to obtain an implicit form of synchronization. Delay-insensitive traffic has a lower priority than real-time traffic and utilizes the bandwidth left over from the real-time flows.

From the point of view of implementation, Sticky CSMA/CA requires that the nodes maintain *carrier sense tables* that record the periodic flows whose existence it can infer using the CSMA mechanism. Upon arrival of a new real-time flow, a node looks at its carrier sense table to determine a periodic schedule over which the medium is free, and contends for and grabs the medium at the first such opportunity (i.e., it starts sending in a time interval that does not interfere with the existing real-time flows in its carrier sense table). If the flow setup is successful, the node locks onto a periodic transmission schedule. All the other nodes in the network that overhear the control message exchange update their carrier sense tables immediately, and the nodes that sense the new periodic transmission infer that a new flow has been set up. Delay-insensitive traffic has lower priority than real-time traffic and fills in the gaps remaining after the real-time flows have been accommodated. We develop new contention resolution mechanisms for delay-insensitive traffic

<sup>1</sup> The term *sticky routing* has been used for routing protocols in telephone networks [2] that grab and hold resources while feasible. Sticky CSMA/CA also grabs and holds resources, but for channel access: the context and protocol details are completely different.

that enhance the efficiency with which they utilize the left-over bandwidth.

Our performance evaluations focus on a mix of VoIP and data traffic, for which large gains in VoIP call carrying capacity are demonstrated relative to both IEEE 802.11b and IEEE 802.11e. We show through analysis and extensive simulations that Sticky CSMA/CA achieves more efficient medium utilization than IEEE 802.11e and can be used to provide the required QoS in terms of bandwidth, delay and delay jitter to real-time flows. The underlying reason is that this scheme obviates the need for medium contention for every frame that a node transmits, exploiting the predictable and periodic nature of the high priority traffic. This approach significantly reduces the probability of collision and the overhead due to the backoff required for collision avoidance for every frame in IEEE 802.11. We also show that, along with efficient support of significantly more real-time flows, the performance in terms of throughput achieved for the low priority delay-insensitive traffic is comparable to that of IEEE 802.11. These encouraging results indicate that even for applications that do not exhibit periodicity and constant data rates naturally (e.g., video, or even high-speed *data pipes* in a wireless backhaul), it may be useful to artificially impose periodicity and predictability to improve the efficiency of medium access. Detailed investigation of cross-layer architectures that shape variable rate real-time traffic to take advantage of Sticky CSMA/CA is beyond the scope of this paper. However, we do provide some discussion as to how this might be done for applications such as video in Section 6.

The rest of the paper is organized as follows. Section 2 describes related work. The design of Sticky CSMA/CA is described in Section 3. In Section 4, we present an approximate analysis to obtain insight into the performance of IEEE 802.11 and Sticky CSMA/CA. Section 5 presents the simulation results obtained in mesh networks with line and grid topologies. We conclude with a discussion of various design and application aspects of Sticky CSMA/CA in Section 6. Note that we use the terms VoIP and voice interchangeably throughout this paper.

## 2. Related work

In the past few years, a considerable amount of work has addressed the area of MAC layer design

for providing the required QoS to voice, multimedia and data applications over wireless networks. We discuss some of the proposed approaches next.

The IEEE 802.11 standard MAC sublayer provides two different access mechanisms: the Distributed Coordination Function (DCF), and the Point Coordination Function (PCF) [5]. PCF is a centralized approach to perform bandwidth reservations, which splits time into a contention-free period (CFP) and a contention period (CP). A polling scheme controlled by a Point Coordinator that resides in the Access Point arbitrates channel access during the CFP. PCF is not suited for multihop wireless networks due to the requirement of centralized control and the Point Coordinator at the Access Point. The Distributed Coordination Function (DCF) is based on CSMA/CA for frame transmissions and a random backoff mechanism to avoid packet collisions. We describe this scheme in more detail in Section 3. DCF's drawback is that it does not provide service differentiation. Due to lack of both medium access priorities and QoS support for different applications, DCF is not suited for supporting real-time applications.

The IEEE 802.11e standard extends the IEEE 802.11 framework by including mechanisms for service differentiation [6]. Four access categories (AC), mapped to eight traffic priorities, are defined for IEEE 802.11e. These ACs correspond to different values of the following parameters: minimum contention window size ( $CW_{min}$ ), maximum contention window size ( $CW_{max}$ ), and the Arbitration Inter-frame Space (AIFS) (time-interval between idle transition of the medium and the start of channel access or backoff). A lower value of these parameters causes shorter backoff intervals or wait periods before medium access, thereby prioritizing traffic. IEEE 802.11e has different queues for different ACs. It also provides an optional token-like mechanism called TXOP (Transmission Opportunity) to support packet bursts, where a node that has obtained access to the medium can send multiple frames for an AC, up to a maximum time-limit, without further contention. This contention-free bursting mechanism aids in transmission of bursty traffic and improves medium utilization. One of the drawbacks of IEEE 802.11e is the lack of assured QoS for individual flows. IEEE 802.11e can only provide service differentiation among different ACs, but QoS degrades in the presence of multiple contenders in the same AC because of medium access contention with equal priority for every

packet belonging to the same AC. The underlying reason for lack of support of a deterministic QoS in IEEE 802.11e and IEEE 802.11 is that every packet requires contention for the medium in these schemes, and no attempt is made to exploit the (natural or possibly imposed) predictability of packet transmissions to minimize contention losses.

Blackburst [7] is a scheme based on modified IEEE 802.11 CSMA/CA designed to provide QoS to real-time applications. In this scheme, a node with real-time data contends for medium access by sending an energy burst (called a blackburst) for time proportional to the wait time of the node. The node with the longest burst wins the contention and transmits a frame. Blackburst does not address the hidden terminal problem and requires that the nodes have the capability to jam the medium. Another distributed MAC scheme derived from IEEE 802.11 DCF is Distributed Fair Scheduling (DFS) [8], which is based on Self-Clocked Fair Queuing (SCFQ) [9]. DFS achieves service differentiation by assigning different backoff intervals to packets that belong to different flows based on the weights assigned to the flows. This feature ensures higher throughput to flows with higher weights, and fairer overall allocation of bandwidth as compared to IEEE 802.11 DCF. DFS does not consider the delay bounds of real-time packets, nor does it address mechanisms for determining appropriate weights for different flows, both of which greatly influence the performance.

Multiple Access Collision Avoidance with Piggyback Reservations (MACA/PR) [10] is a scheme that supports bandwidth reservations for real-time traffic. In this scheme, each node maintains the real-time scheduling information of the neighboring nodes in its reservation tables (RT) by overhearing the data packets or ACKs. The global reserved slot information is obtained by periodic RT exchange among neighbors. The RT exchange takes care of the hidden terminal problem. The drawbacks of MACA/PR are the piggybacking-based reservation approach and the overhead of periodic RT exchange.

A few recent proposals attempt to take advantage of TDMA schedules for reducing interference and for fair resource allocation over CSMA based networks [11,12]. The Overlay MAC Layer (OML) [11] is an access control and scheduling scheme that partitions time into equal size slots and allocates these slots among loosely synchronized contending nodes. Though OML implements temporal fairness and reduces interference by improving the predict-

ability of medium access among nodes, the coarse time-slot based resource allocation does not address the QoS requirements of different applications. Another interesting MAC framework proposed in the context of sensor networks is ZMAC [12]. ZMAC exploits TDMA schedules among locally synchronized neighbors to improve medium utilization and reduce packet collisions among two hop neighbors in a CSMA network operating under high contention. Under low contention, all the nodes contend for a slot but the slot-owner nodes have a higher priority over their neighbors by virtue of a shorter contention window. This scheme incurs initial schedule allocation overhead in the network deployment phase and, because it was designed for a sensor network, it does not consider service differentiation among different applications for QoS support.

To the best of our knowledge, the proposed Sticky CSMA/CA scheme is the first approach that exploits the characteristics of real-time applications to devise medium access control strategies that reduce delay and delay jitter. Exploiting the natural periodicity of VoIP is shown (in the succeeding sections) to lead to large capacity gains relative to IEEE 802.11. This motivates future work (see the discussion in Section 6) in taking this approach one step further, by imposing artificial regularity at the application layer in order to obtain the enhanced QoS obtained using Sticky CSMA/CA medium access.

### 3. The sticky CSMA/CA protocol

In this section, we describe the Sticky CSMA/CA protocol. We briefly outline conventional CSMA/CA and introduce history considerations into CSMA/CA. We then describe the carrier sensing mechanism and discuss the operation of Sticky CSMA/CA for real-time and delay-insensitive traffic.

#### 3.1. Conventional CSMA/CA

The IEEE 802.11 DCF uses CSMA/CA for data packet transmission, along with a random backoff mechanism to avoid collisions. A node that has a frame to transmit senses the transmission activity in the medium. If the medium is free for a distributed interframe space (DIFS) interval, the frame is transmitted. If the medium is sensed as busy, the node waits until the medium is free for a fixed time interval

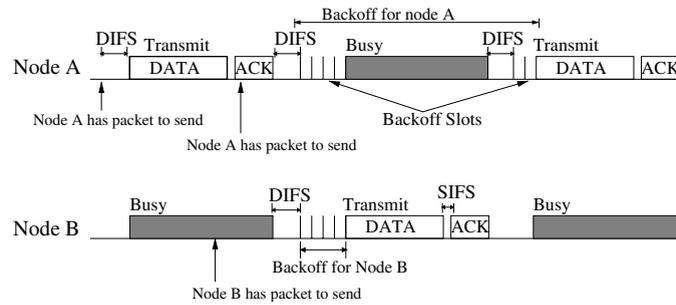


Fig. 1. CSMA/CA: Backoff mechanism.

(DIFS if the last frame was received without any transmission errors, or extended interframe space (EIFS) otherwise). After the medium is sensed idle for a DIFS/EIFS interval, the node generates a backoff counter  $b_i$ , which is a pseudo-random integer with a uniform distribution in the interval  $[0, CW_i]$ , where  $CW_i = \min(CW_{\max}, 2^{i-1}(CW_{\min} + 1) - 1)$  for the  $i$ th transmission attempt for a frame. The node then decrements the backoff counter  $b_i$  by one for every *slot width* interval if the medium remains idle. If the medium becomes busy before  $b_i$  reaches zero, the backoff procedure is frozen and is continued after the medium is sensed idle again for a DIFS interval. When the backoff counter reaches zero, the node transmits the frame. If the transmission fails and the ACK is not received from the receiver, then the node moves to the next backoff stage ( $i + 1$ ) and repeats the procedure. Fig. 1 describes the backoff procedure for two nodes, A and B, that can hear each other.

### 3.2. Introducing history into CSMA/CA

CSMA/CA with history incorporates the history of the recent transmission activity in the medium into the conventional CSMA/CA mechanism for packet transmissions. The primary difference from conventional CSMA/CA is that the medium activity history information is used to infer the time windows that are occupied by periodic real-time flows. Time windows with periodic transmissions are considered *reserved* by the existing real-time flows, and hence there is no contention for these time windows. We assume that all the real-time flows are periodic with a common period. The most relevant example of such flows is voice applications, which generate periodic data based on the codec used (e.g., ITU-T G.711 codecs periodically generate fixed sized packets at a data rate of 64 Kbps).

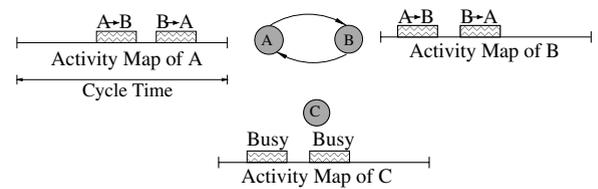


Fig. 2. Activity map: nodes A and B have a bidirectional real-time flow between them. Node C is a neighbor of nodes A and B. The activity maps of nodes A, B and C contain reservations for the real-time flow between nodes A and B. Note that the cycles need not start at the same time.

We define a *cycle* as the time period between two consecutive packet transmissions of any ongoing real-time session. Therefore, the transmission activity over the last few cycles can be used to infer the real-time flow reservations of the medium. All the nodes in the network maintain a *medium activity map* containing the medium busy/free state information over a fixed number of past cycles. A node invokes conventional CSMA/CA to find a transmit opportunity, taking into account its medium activity map to ensure that it does not interfere with any existing real-time flow. A backoff mechanism similar to IEEE 802.11e, but with smaller contention window parameters, is followed to avoid collisions. As an example, consider a three node network with a bidirectional real-time flow between nodes A and B. Fig. 2 shows the activity maps of the medium at nodes A, B and C, assuming that node C can hear nodes A and B.

### 3.3. Sticky CSMA/CA

We view continuous time in terms of time-slots. As mentioned before, we assume that the cycle time is the same for all the nodes in the network. Note that since we assume an asynchronous framework, the cycles can start at different times at different

nodes. A node with a new voice session practices a greedy algorithm to grab a free time window. A free time window is defined as a set of contiguous time-slots during which the medium is free. If the attempt to establish a flow is successful (we describe the mechanism of establishing a call later in this section), the node sticks to the time window. This means that the node sends packets periodically during the same time window in every cycle, unless the session performance, measured in terms of packet loss, degrades beyond a certain threshold. All new voice connection setup attempts and other packet transmissions honor the time windows occupied by the existing voice connections and do not contend for those slots. We describe the carrier sensing and contention mechanisms of Sticky CSMA/CA, and discuss various implementation considerations next.

### 3.3.1. Carrier sensing mechanism

All nodes in the network maintain a set of *Carrier Sense History Tables (CSH Tables)*. A single CSH Table is a log of medium transmissions sensed by a node over a cycle. The CSH table contains one entry for each time-slot of the cycle. Each entry in the CSH Table is marked busy or free, based on the transmissions sensed in the medium during the corresponding time-slot. A node marks a slot as busy if it senses a transmission in its neighborhood during that time-slot, or if it transmits or receives in that time-slot. The CSH Table is updated at every medium transition among the four possible physical layer states: Idle, Sensing, Receiving, and Transmitting. It is important to note that the MAC layer does not need to be explicitly informed about the state of each time slot. MAC layer can deduce the slot states from the medium state transition information passed from the PHY layer without any additional overhead. Medium state transition information is required even in conventional CSMA MAC schemes. Fig. 3 shows an example of CSH Table maintenance. The number of CSH Tables required in order to maintain the history of medium activity over the past  $N$  cycles is  $N$ .

In addition to the CSH Tables, the nodes maintain another table called the *Carrier Sense Table (CS Table)*. The CS Table is an activity map of the medium that represents the predicted busy and free time-slots over the current cycle. The CS Table has a format similar to a single CSH Table. The CS Table is derived at the beginning of every cycle using the CSH Tables over the previous  $N$  cycles. A simple approach to obtain a CS Table is to do a major-

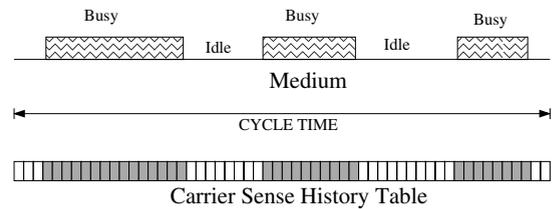


Fig. 3. Nodes use CSMA to monitor the medium and mark their current carrier sense history table according to the sensed medium busy and idle times.

ity decision over each slot in a cycle, e.g., if a slot is found busy in 75% of the CSH Tables over the last  $N$  cycles, it is marked as busy in the CS Table. Fig. 4 shows an example of CSH and CS Tables.

The CS Table is maintained only on the basis of periodicity of the sensed transmissions, and the information about the flows for which the node itself is the sender or receiver. This approach significantly reduces the control overhead as compared to the other techniques involving message exchanges, piggybacking control information over data packets, or high power transmissions. There is no extra message exchange required in this technique.

At high medium utilization, a few large sized data packets (e.g., FTP packets of size 1500 bytes) transmitted in the carrier sensing range (CSR) of a node over contiguous cycles might cause the node to falsely interpret these transmissions as voice connections. This can cause the node to mark the corresponding slots in its CS Table as busy, resulting in underutilization of bandwidth. We observe that a high percentage of data packets in the network are significantly larger than voice packets. We use this property to distinguish such data packets from voice packets. Whenever a node senses a transmission for a time interval significantly longer than that possible for a voice packet (e.g., more than double the time required for a voice packet), it concludes that it has sensed a low-priority data packet transmission and does not mark its CSH Table. Also, a node does

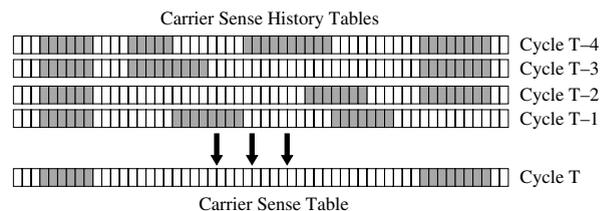


Fig. 4. At the beginning of each cycle, nodes derive their carrier sense table as a summary of their carrier sense history tables. The carrier sense table helps to identify periodic real-time flows.

not mark its CSH Table whenever it receives or overhears low-priority data packet transmissions. In addition, simple sanity checks of the CS Table data can also help in removing irrelevant history information from the CS Table. The underlying reason for not accounting for low priority data transmissions in the CSH Table is to ensure that a node considers only relevant high priority periodic data transmission information while preparing its CS Table, thus foregoing transmission opportunities only for the periodic high priority traffic in the network. These mechanisms improve medium utilization and aid in prioritizing voice flows in the entire CSR of a node.

### 3.3.2. Contention mechanism for real-time flows

To set up a new voice connection, a node checks its CS Table to find a set of contiguous free time-slots to accommodate the new flow. If a free time window is found, the node waits until the beginning of the free time window. It then employs the IEEE 802.11 CSMA/CA mechanism to transmit a Real-time RTS (R-RTS) message to the intended receiver. The R-RTS message contains the required packet transmission duration information for the new flow and a flow-identifier. The flow-identifier, along with the source node address, uniquely identifies a flow in the network. On receiving an R-RTS message, the receiver checks its CS Table (activity map) to determine whether it can support the flow from the time instant when the R-RTS transmission started (the propagation delay is negligible as compared to the transmission duration, and thus is not considered). If the corresponding time-slots are free, the node sends a Real-time CTS (R-CTS) message to the sender. The R-CTS message contains the flow-identifier and the duration information

obtained from the R-RTS message. The sender and the receiver mark their CS Table and overwrite all the CSH Tables to indicate the presence of the new flow. The neighboring nodes of the sender and the receiver overhear the R-RTS and R-CTS messages and use the duration information to update their CSH Tables in a similar manner. The nodes in the carrier sense range (CSR) of the sender and the receiver sense this periodic transmission and update their CS Tables after a few cycles. Fig. 5 shows the flow setup procedure for a node.

To ensure that real-time flows win over low priority datagrams while contending for the medium during flow setup, we use an approach similar to IEEE 802.11e. IEEE 802.11e uses the contention window parameters ( $CW_{min}$ ,  $CW_{max}$ ) and the AIFS to prioritize different Traffic Classes (TC). Smaller  $CW_{min}$  and  $CW_{max}$  values and a shorter AIFS period ensure that a packet that belongs to a high priority TC accesses the medium ahead of a packet belonging to a low priority TC, thereby prioritizing different TCs.

After a successful flow setup, the sender locks on to the slots; i.e., it transmits voice packets during the same time-slots in every cycle. The receiver does not send ACKs for the voice packets. ACKs result in a significant overhead for real-time flows with short packets such as voice because the size of the ACK packets is comparable to the size of the data packets [13]. Also, the physical layer preamble and header (PLCP preamble and header), which are transmitted at the lowest data rate, consume most of the transmission time required for these short packets.

Packet loss information summaries are sent by the receiving node to the sender through a *feedback mechanism*. The feedback mechanism is required for the sender to evaluate the condition of a real-time

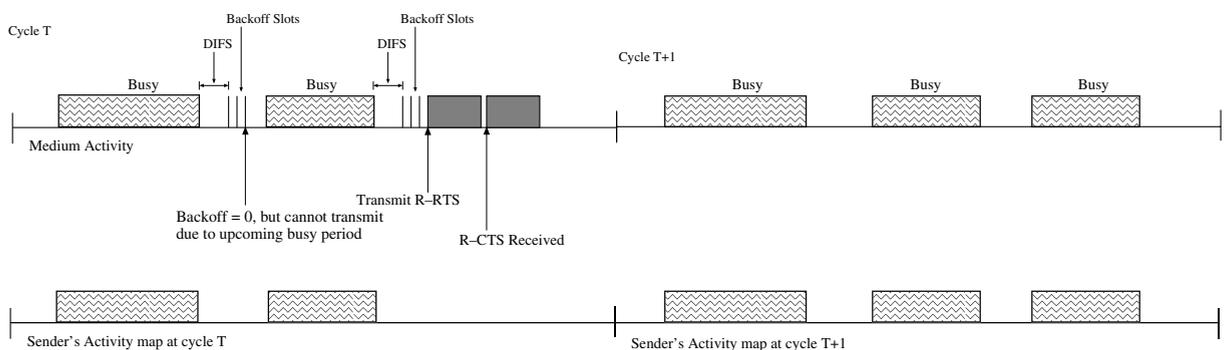


Fig. 5. Establishing a new flow: the sender uses its activity map to select a suitable time window when it can establish a new real-time flow. The node transmits an R-RTS message at the chosen slot and receives an R-CTS message in response, thus setting up the flow.

flow in terms of packet loss in the medium. This is necessary because the ACK mechanism of IEEE 802.11 is not used for real-time packets. Feedback messages are used to inform the sender of the cumulative packet losses seen by the receiver over the last  $M$  cycles. A sender node requests a feedback message by setting the Feedback Requested bit in the MAC frame header of the voice packet. The sender requests a feedback message after every  $M/2$  cycles without feedback and continues sending the request with every packet of the flow until it receives a feedback message. If the maximum number of cycles  $M$  without the feedback message is reached, the sender pauses the flow and reattempts flow setup at a different time-slot through an R-RTS/R-CTS dialog. We call this mechanism *slot reconfiguration*. The sender also utilizes the cumulative packet loss information sent in the feedback messages to decide whether slot reconfiguration is required.

Sticky CSMA/CA addresses the hidden terminal problem in the following ways. The control message exchange for slot reservation (R-RTS and R-CTS) between the sender and the receiver incorporates the receiver's view of the network before choosing slots for data transfer. A receiver does not respond to the sender's R-RTS message if it is aware of periodic transmissions over a portion of the time window requested, or if it senses the medium to be busy during these slots. In case a sender-receiver pair observes persistent packet losses over the reserved slots because of interference (e.g., from a hidden terminal), the node pair attempts to shift to another free time window that has a higher probability of being interference free. Thus, nodes avoid becoming trapped in lossy transmission patterns by attempting slot reconfiguration after the observed packet loss exceeds a certain threshold. Our performance evaluation shows that slot reconfiguration is an effective technique to avoid lossy transmission patterns.

Another effective solution to counter packet losses due to interference is *out-of-stream* packet retransmission. When the receiver of a periodic flow does not receive an expected packet, it sends a retransmit request to the sender. On receiving a *retransmit request*, the sender retransmits the corresponding packet as a *high priority* datagram. The requested packet should be sent out-of-stream because inclusion of this packet in the periodic stream will delay the subsequent packets in the stream. Our performance evaluation indicates that out-of-stream retransmissions effectively counter

losses of high priority voice packets. Another approach for minimizing packet losses due to interference is to provide redundancy in the transmitted payload by including multiple voice payloads in one frame, so that the loss of a few frames does not affect the performance [13].

In general, the hidden terminal problem can be partially addressed by using sensitive antennas (i.e., with low carrier sense threshold) at the nodes so that any node that wishes to transmit can sense whether it will interfere with its neighbors' ongoing packet receptions, and thereby avoid such interference. This solution has an associated tradeoff of reduced spatial reuse because of the aggravated exposed terminal problem. Nevertheless, this approach has been proposed in literature for reducing packet collisions in systems where the required antenna sensitivity is not a design constraint [14]. We have observed this tradeoff for Sticky CSMA/CA in our performance evaluation.

### 3.3.3. Contention mechanism for delay-insensitive flows

Delay-insensitive flows are assigned a lower priority than real-time flows. MAC layer operation for delay-insensitive data flows is based on CSMA/CA with history considerations. The data packet transmissions honor the reservations made by real-time flows. Nodes do not attempt to transmit data packets over slots in which high priority periodic transmission is expected. As mentioned before, data packets have higher backoff parameters and a higher AIFS than that of real-time flows to ensure that real-time flow setup requests receive a higher priority. However, the  $CW_{\min}$  and  $CW_{\max}$  parameters are set to lower values as compared to IEEE 802.11. This is because the time gaps left by the real-time flows might not be sufficiently large to incorporate the probable long delays due to the exponential backoff mechanism used in IEEE 802.11. Moreover, with reduced contention in a voice/data network, large contention windows are not required for collision avoidance. Our performance evaluation shows that even with smaller contention window parameters, the throughput achieved by Sticky CSMA/CA for delay-insensitive best effort traffic is comparable to that for IEEE 802.11.

With a large number of voice calls in the network, the time gaps left might not be sufficient to transmit data packets, which can lead to starvation of data flows. In systems where starvation of data

flows is not acceptable, this problem can be alleviated by setting a threshold on the number of voice calls admitted. Admission control schemes to enforce such thresholds are higher layer functionalities that are beyond the scope of this paper. We therefore do not impose any limits on the number of voice calls in our performance evaluation.

### 3.3.4. Implementation considerations

We discuss some important considerations for the implementation of Sticky CSMA/CA in this section.

*Time-slots:* Time-slots are the time units in terms of which we define the entries (marked busy or free) in the CS Table and the CSH Tables. We define time-slots of a fine granularity. We mark a time-slot as busy even when the medium is busy for only a fraction of the time-slot duration. Thus, a time-slot should be small enough so that the portion of time for which the medium is not utilized is insignificant. Also, a small slot width increases the memory requirements for the tables. Based on these considerations, we chose the time-slot duration for Sticky CSMA/CA as 20  $\mu$ s, which is the default slot duration (used for backoff) of IEEE 802.11b. In our simulations, we assume the cycle time as 20 ms (the packetization interval for voice). To allow for deviations of clocks of different nodes, extra slots (called *Leeway slots*) at the beginning and the end of the transmission are marked as busy in the CSH Table.

*CS Table Maintenance:* The CS and the CSH Tables are implemented as bitmaps, with each bit corresponding to a time-slot. The CSH Table for the current cycle is updated only on medium transitions in order to avoid the overhead of updating the table at every time-slot. Given a fixed cycle time, the size of the CS Table and the CSH Tables depends on the width of the time-slots. Also, the total size of the CSH Tables depends on the number of previous cycles ( $N$ ) needed to derive the CS Table. For example, with the above mentioned slot width and cycle time, the memory requirement for a system with CSH Tables spanning six cycles is 875 bytes.

*Reattempt Timer:* It is necessary that the CS Table represent an accurate prediction of the transmissions in the current cycle. A node that senses the medium as free might not be allowed to contend for the channel because its CS Table indicates an impending transmission. An example scenario where this can happen is when a real-time flow ends in the carrier sensing range (CSR) of a node; the node will have the corresponding slots in its CS

Table marked as busy in the current (and possibly the next) cycle, depending on the majority decision rule used to decide the busy/free status of each slot. In order to improve medium utilization in such cases, we use a *Reattempt timer*. A node starts a reattempt timer when it is not allowed to set up a new flow or transmit a data packet because the medium is expected to be busy as per its CS Table. The timer is set to expire at one of the intermediate slots of the expected busy period. When the reattempt timer expires, the node determines whether the medium is actually busy. If the medium is found to be free, the node initiates the Sticky CSMA/CA packet transmission procedure again. This mechanism increases medium utilization even in the less probable case of an inconsistent CS Table.

*Dummy Reservation Packets:* When there is a packet loss at an upstream node during an ongoing multihop voice flow, the downstream nodes of the flow do not have packets to send to their next hop neighbors. In this situation, a downstream node sends dummy packets at its scheduled transmission slot to the next hop node. This transmission helps in avoiding contention for the slot by the CSR neighbors, in case of loss of a few packets (less than the loss threshold  $M$ , at which we delete the flow). Dummy reservation packets can also be used to maintain slot reservations during short silence periods (up to a maximum time limit) when VoIP packets are not generated because of the silence suppression methods being used in the codec.

*Probabilistic CSH Table Updates:* A node that receives a data packet from its neighbor cannot simultaneously sense the real-time transmissions of another non-interfering node in its CSR. The repeated occurrence of this event (due to high data traffic) can cause the nodes to lose the periodic flow information in their CS Tables. This in turn allows them to transmit in the slots reserved by their CSR neighbors and cause interference. For this reason, we increase the probability of a node's retention of the real-time flow information in its CS Table by using *probabilistic updates* to the CSH Table. Whenever a node loses the periodic transmission information of its CSR neighbors from its CS Table due to receiving a long data packet, it copies the information from its CS Table to the current CSH Table with a high probability, assuming that the CS Table has more reliable information for those slots. Probabilistic CSH Table updates reduce the chances of losing the slot reservation information and result in improved performance for voice flows

in the CSR of a node. We incorporate probabilistic updates in our simulations with a high probability value ( $p = 0.95$ ).

#### 4. Performance analysis for a clique

We now provide an approximate analytical comparison of the number of VoIP calls supported by Sticky CSMA/CA against those supported by IEEE 802.11b DCF [15] and IEEE 802.11e EDCF [6]. For this purpose, we consider single hop flows over a clique, which is a network where all nodes can hear each other. We assume that there are no hidden terminals and the radio link is error free. Despite these simplifications, the clique-based analysis can be used to predict performance for multihop flows over more complex networks, which can be roughly viewed as multiple cliques *stitched* together. Because of the shared nature of the wireless medium, the bandwidth required by a multihop flow can be estimated as the number of hops traveled times the bandwidth required over one link. Such computations can then be used to identify bottleneck links, which limit the maximum throughput in a multihop wireless network [16]. Applying the clique-based estimates to the nodes surrounding the bottleneck link then provides capacity estimates that match closely with the simulated performance.

##### 4.1. IEEE 802.11b DCF performance

The performance of IEEE 802.11 DCF has been analyzed in detail in [17,18] for single hop scenarios. Refs. [19,20] incorporate the considerations for multihop flows. Our objective is to obtain the total voice call carrying capacity for an IEEE 802.11b network. We estimate the voice carrying capacity of IEEE 802.11b using the saturation throughput analysis in [17], where saturation throughput is defined as the maximum stable throughput of the network. This analysis assumes ideal channel conditions (no

hidden terminals, error free radio link, and backlogged packet queues at every node) for a network of  $n$  nodes. Through Eqs. (1)–(3), we briefly outline the derivation found in [17] for convenience (refer to [17] for details).

Continuous time is divided into variable length time intervals, defined as *epochs*, at the beginning of which the backoff counter of each node decreases by one. These epochs can be the IEEE 802.11b *slot width* (hence forth called  $\delta$ ) of 20  $\mu$ s if the medium is free, or can also include a medium busy time interval, since the backoff counter is frozen when the medium is sensed busy. In the following analysis of IEEE 802.11b DCF, by the term epoch, we will refer to this definition of variable length time intervals. Fig. 6 shows the epochs as the time intervals between two consecutive backoff counter decrements. A key assumption in this model is that at each transmission attempt, irrespective of the backoff state of the node, the packet transmitted collides with a constant and independent probability  $p$ , called the *conditional collision probability*. Under these assumptions, the stochastic processes representing the backoff stage  $i \in (0, m)$ ; and the backoff counter  $k \in (0, W_i - 1)$ , where  $W_i = 2^i W_{\min}$ ; can together be modeled as a bidimensional discrete-time Markov chain. Using this Markov chain model, the probability that a node transmits in a chosen epoch is obtained as

$$p_t = \frac{2(1 - 2p)}{(1 - 2p)(W_{\min} + 1) + pW_{\min}(1 - (2p)^m)}, \quad (1)$$

where  $W_{\min}$  is the minimum contention window, and  $m$  is the maximum allowed backoff state (i.e.,  $W_{\max} = 2^m W_{\min}$ ). In order to evaluate the conditional collision probability  $p$ , due to the independence assumption,  $p$  can be defined as

$$p = 1 - (1 - p_t)^{n-1}. \quad (2)$$

This means that the probability of collision for a transmitted frame is the probability that at least

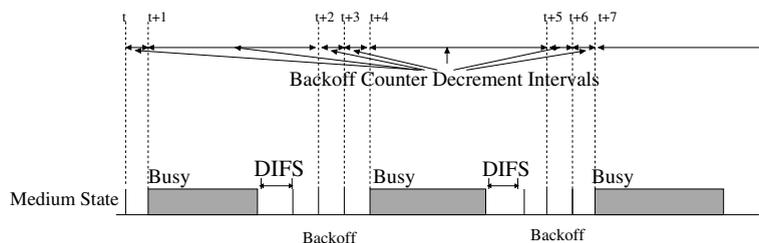


Fig. 6. Variable time-intervals between two consecutive backoff counter decrements at an IEEE 802.11 node.

one of the other  $n - 1$  nodes also transmits in the same epoch. Eqs. (1) and (2) can be solved for different values of  $n$ ,  $m$ , and  $W_{\min}$  to obtain  $p$  and  $p_t$  for these cases.

Define the probability that there is a transmission in the considered epoch as  $P_{tx}$ , and the probability that a transmission that occurs in an epoch is successful as  $P_s$ . The normalized saturation throughput  $S$  is defined as the fraction of time the channel is used to successfully transmit payload bits, and can be evaluated as the ratio of the average time spent in the transmission of useful payload information over an average epoch duration. Define  $T_{\text{payload}}$  as the time taken to transmit a payload of a fixed size,  $T_{tx}$  as the time taken for successful transmission of a complete packet including headers and other overheads, and  $T_{\text{collision}}$  as the epoch duration when the medium is sensed busy by each node during a collision. Therefore,

$$S = \frac{P_{tx}P_sT_{\text{payload}}}{(1 - P_{tx})\delta + P_{tx}P_sT_{tx} + P_{tx}(1 - P_s)T_{\text{collision}}}. \quad (3)$$

The normalized saturation throughput  $S$  can be evaluated for different sets of  $n$ ,  $m$ , and  $W_{\min}$ . The capacity of a bottleneck link in a multihop network can be approximated through  $S$  evaluated for the two node case using (3), where both nodes always have packets to transmit. The bottleneck link capacity constrains the maximum number of multihop flows in the network where all the nodes can hear each other. For example, if the bottleneck link supports a maximum of  $k$  voice calls, the network can support a maximum of  $k/2$  two hop voice calls, or  $k/3$  three hop voice calls, etc. Note that in reality, this approximation might be imprecise because all the transmitters might not transmit with the same power; the nodes might not be spaced equally apart; and the channel conditions are not the same at all the nodes. Nevertheless, it helps to obtain an estimate of the number of calls that can be supported.

#### 4.2. IEEE 802.11b VoIP call carrying capacity

Table 1 shows the overhead associated with each voice frame sent, due to different protocol layer headers and the MAC protocol overhead. We assume that an ITU-T G.711 codec packetizing audio data every 20 ms is used, which results in payloads of size 160 bytes (data rate = 64 Kbps). We assume that IEEE 802.11b operates at the highest data rate of 11Mbps using the basic access mechanism (no RTS-CTS). We also assume that IEEE

Table 1  
Overhead in terms of time, with average data rate  $R$

Overhead	Bytes	Time ( $\mu$ s)
RTP	12	$12 * 8/R$
UDP	8	$8 * 8/R$
IP	20	$20 * 8/R$
IEEE 802.11 MAC	28	$28 * 8/R$
PHY preamble		72/1
PHY header		48/2
SIFS		20
DIFS		50
ACK	14	$14 * 8/R + \text{PHY overhead}$
Feedback	20	$20 * 8/R + \text{PHY overhead}$
Propagation ( $\Delta$ )		1

802.11b uses the *short* physical layer preamble (PLCP preamble), which reduces the synchronization time from 192  $\mu$ s (when the long physical layer preamble is used) to 96  $\mu$ s. The values mentioned above correspond to the state-of-the-art IEEE 802.11b VoIP equipment commercially available [4]. Define  $T_{\text{frame}}$  as the time required to transmit a frame (payload and headers), and  $\Delta$  as the propagation time. From Table 1, we have

$$T_{\text{payload}} = \frac{160 \cdot 8}{11} \mu\text{s},$$

$$T_{\text{frame}} = T_{\text{PHY}} + T_{\text{MAC}} + T_{\text{RTP+UDP+IP}} + T_{\text{payload}} \\ = 262 \mu\text{s},$$

$$T_{tx} = T_{\text{frame}} + \text{SIFS} + \text{ACK} + \text{DIFS} + 2 \cdot \Delta = 430 \mu\text{s},$$

$$T_{\text{collision}} = T_{\text{frame}} + \text{DIFS} + \Delta = 313 \mu\text{s}.$$

Solving Eqs. (1) and (2) numerically for the two node case  $n = 2$ ,  $W_{\min} = 32$ , and  $m = 5$ , (as  $(CW_{\min}, CW_{\max})$  for IEEE 802.11b are (31, 1023)) [15] and plugging in the required probability and transmission time values in (3), we get the normalized saturation throughput  $S = 0.1924$ . This gives the saturation throughput =  $S * 11 \text{ Mbps} = 2.1164 \text{ Mbps}$ . Each voice call is equivalent to two unidirectional constant bit rate (CBR) flows of data rate 64 Kbps. Therefore, the required data rate for a single VoIP flow = 128 Kbps. Thus, we obtain the maximum number of calls supported as 16. This number matches with the capacity we obtain from simulations, as shown in Fig. 7. The simulation parameters are summarized in Section 5. Note that if the long preamble is used, the maximum number of calls supported by IEEE 802.11b network decreases to 12.

The IEEE 802.11e EDCF network for a voice-only scenario is similar to IEEE 802.11b DCF, with different contention window parameters (i.e., (7, 31)

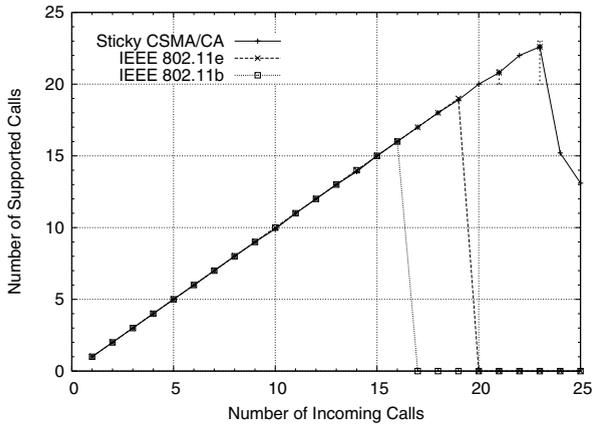


Fig. 7. The total number of voice calls supported as a function of the incoming calls for a two node network, with nodes 150 m apart.

instead of (31,1023) for voice packets). Note that prioritization of different access categories does not play a role in this scenario because we are only considering voice calls. The total number of voice calls obtained in this case is 19, which matches with the capacity obtained via simulations (see Fig. 7). It is important to note that optional IEEE 802.11e protocol features (such as transmission opportunity (TXOP) that facilitates contention free bursting) when enabled, result in a different overall network behavior. The effect of TXOP has been accounted for in the simulation based performance evaluation in Section 5.

#### 4.3. Sticky CSMA/CA VoIP call carrying capacity

Sticky CSMA/CA requires the nodes with real-time flows to stick to a periodic transmission schedule. The other nodes in the carrier sensing range (CSR) neighborhood that hear the periodic transmissions do not attempt to contend for the medium during the slots in which they expect a periodic transmission. By sticking to a periodic transmission schedule and using contention only to *reserve* slots for a real-time flow, Sticky CSMA/CA reduces the number of contending nodes in the medium drastically (in contrast with IEEE 802.11, where the number of contenders at any instant equals the number of nodes that have packets to transmit). This leads to substantial gains in terms of reducing the overhead involved with backoff for every packet transmitted in the medium. Based on this fact, we reason that in a network with only voice flows, the

total number of contenders at any time instant in a Sticky CSMA/CA network equals the number of new arriving calls (and the calls being reconfigured to a different time window), which is significantly less than the total number of nodes with packets to transmit at any instant. With this observation, we argue that it is reasonable to assume that, on average, the first stage of backoff yields a winner with a high probability, even with a very small  $CW_{min}$ . Thus, we use contention window parameters smaller than IEEE 802.11b for real-time flow setup requests in a Sticky CSMA/CA network:  $CW_{min} = 3$  and  $CW_{max} = 7$ . These contention window parameters are used in a manner similar to IEEE 802.11e for service differentiation. We do not need to account for the slots wasted due to collisions in the calculation of the maximum number of VoIP connections because the time-slots wasted in the current cycle due to a collision (and the consequent DIFS interval and backoff) can effectively be used by a new VoIP call in the coming cycles.

Sticky CSMA/CA uses feedback messages sent periodically or on-demand to the sender in place of ACK messages for voice packets. In our simulations, the sender demands a feedback message every sixth cycle. Thus, the average additional medium access time required per cycle to transmit feedback messages for a real-time flow is one-sixth of the time required for transmitting a single feedback message. Note that the slots reserved for a voice packet also include leeway slots ( $L = 1$ ) on both sides of the reserved slots.

We express all the time requirements in terms of time-slots (with slot size  $\delta = 20 \mu s$ ). Because a cycle spans 20 ms (the rate at which the 160 bytes voice packets are generated),  $T_{cycle}^s = 1000$ . The minimum average gap between the slots reserved for a new voice call placed next to an already existing reservation in the cycle is  $DIFS + E[x] \cdot \delta$ , where  $E[x]$  is defined as the expected value of a random variable  $x$  that models the portion of the time gap (expressed in terms of  $\delta$ ) between an established voice call and a new call placed next to it that is contributed by the backoff duration associated with every new call. The last reserved slot of a new call need not have any time gap from the first slot of an already existing call reservation placed next to it. This means that two adjacent calls might or might not have any time gap between them, depending on the order of their arrival. We assign equal probabilities to the two possible call arrival orders. Thus, we define the average minimum time gap associated with each call as

$T_{\text{empty}}^s = 0.5\left(\frac{\text{DIFS}}{\delta} + E[x]\right)$ , where  $E[x] = \frac{\text{CW}_{\text{min}}}{2} = 1.5$  slots. Define the total slots required for transmitting a frame, including the overheads as  $T_{\text{tx}}^s$ ; total slots required for feedback messages (of size 20 bytes) as  $T_{\text{feedback}}^s$ ; and the total number of slots required as  $\text{Slots}_{\text{reqd}}$ . Then,

$$T_{\text{empty}}^s = \frac{(E[x] \cdot \delta + \text{DIFS})}{2\delta} = 2,$$

$$T_{\text{tx}}^s = \frac{(T_{\text{frame}} + \Delta)}{\delta} + T_{\text{empty}}^s = 16,$$

$$T_{\text{feedback}}^s = \frac{T_{\text{PHY}} + \frac{20 \cdot 8}{11} + \Delta + \text{DIFS}}{\delta} + E[x] = 10,$$

$$\text{Slots}_{\text{reqd}} = \left(T_{\text{tx}}^s + \frac{T_{\text{feedback}}^s}{6}\right) + 2 \cdot L = 20.$$

Therefore, the number of slots required per voice connection comprising two unidirectional flows is  $T_{\text{voip}}^s = 2 * \text{Slots}_{\text{reqd}} = 40$ . Thus, the maximum number of calls that can be supported is estimated as  $T_{\text{cycle}}^s / T_{\text{voip}}^s = 25$ . This analysis shows that Sticky CSMA/CA has more than 50% additional call carrying capacity as compared to IEEE 802.11b, and more than 25% additional call carrying capacity as compared to IEEE 802.11e. The maximum number of calls supported for the two node scenario obtained by simulations is 23 (see Fig. 7). The error bars at 21 or more incoming calls indicate that the number of supported calls varies depending on call arrival times in the different simulation runs. The loss at high number of incoming calls can be attributed to the packing inefficiency while placing calls in a cycle. We do not employ any mechanism to tightly pack the calls in the cycle where a greater number of calls could be supported.

Thus, Sticky CSMA/CA achieves almost 45% more VoIP call capacity as compared to IEEE 802.11b, and 20% more VoIP call capacity as compared to IEEE 802.11e over the two node scenario. The underlying reasons are the gains due to periodic scheduling and reduced contention as discussed earlier.

While evaluating the voice call carrying capacity, we have considered only voice traffic in the network. When data is introduced, Sticky CSMA/CA performs much better than IEEE 802.11b/e in terms of the number of supported calls because of the prioritization of voice traffic induced by grabbing a periodic transmission schedule. Moreover, the lower contention window and AIFS parameters help in achieving a higher priority for voice flow setup requests, which aids the overall voice throughput.

In Section 5 we demonstrate that even with a higher priority to voice calls and no packing optimization, the data throughput achieved by Sticky CSMA/CA is comparable to IEEE 802.11b/e.

We now evaluate the performance of Sticky CSMA/CA and compare against both IEEE 802.11b and IEEE 802.11e via simulations.

## 5. Performance evaluation

In this section, we evaluate the performance of Sticky CSMA/CA in terms of VoIP calls supported along with the low priority data traffic, over different network scenarios. Through simulations, we demonstrate that Sticky CSMA/CA achieves a much higher VoIP call carrying capacity than IEEE 802.11b/e while providing the required QoS to the VoIP flows.

### 5.1. Simulation setup

We use the QualNet Network Simulator [21] to evaluate the performance of Sticky CSMA/CA. We consider three different network topologies: a single line of five nodes, spaced 150 m apart; a regular grid of 25 nodes, spaced 150 m apart; and an irregular grid of 25 nodes with an average distance of 150 m between neighboring nodes. Figs. 8 and 9 illustrate the line and the regular grid topologies along with the gateway nodes that connect the mesh network to the wired backbone. We discuss the irregular grid topology later in this section. These topologies model different degrees of medium contention and interference among nodes in mesh networks and aid in understanding the performance of Sticky CSMA/CA and IEEE 802.11b/e over real mesh network deployments. Because all the (outgoing/incoming) voice calls and data traffic are routed to their destinations through the wired backbone, the gateway nodes act as terminal points for the voice and data traffic in a mesh network. This arrangement of nodes models the current single radio mesh network deployments, where the closest gateway for each mesh node is usually no more than



Fig. 8. Line topology: the nodes are 150 m apart. The edge nodes 1 and 5 act as gateway nodes, with all the calls being routed through them.

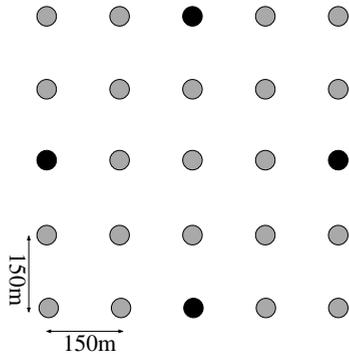


Fig. 9. Grid topology: 25 nodes placed 150 m apart in a square grid. The nodes colored black act as the gateway nodes for the network.

two hops away [22]. Because Sticky CSMA/CA is designed for mesh networks, all the nodes in the network are assumed to be stationary.

We use the two-ray propagation model with constant shadowing as our physical layer model. To evaluate Sticky CSMA/CA using a simple and tractable physical layer model, we consider a flat terrain and neglect the time-varying effects of multipath fading. Table 2(a) lists the simulation parameters used in our simulations. The physical layer parameters used for Sticky CSMA/CA result in a transmission range (TR) of 200 m, and a carrier sensing range (CSR) of approximately 450 m. For nodes spaced 150 m apart, the interference range (IR) is approximately 420 m. We use the short physical layer preamble provided by the IEEE 802.11b standard. IEEE 802.11b uses the basic access mechanism (RTS-CTS turned off). Table 2(c) lists the parameters used for service differentiation in IEEE 802.11e. The simulation parameters used for IEEE 802.11b/e simulations are known default parameters that are recommended by the respective standards

Table 2  
Simulation parameters

(a) Physical layer and MAC layer

MAC layer parameters	Values	PHY layer parameters	Values
Slot time	20 $\mu$ s	Propagation model	Two ray model
SIFS	10 $\mu$ s	Antenna	Omnidirectional
DIFS	50 $\mu$ s	PHY noise factor	10
IEEE 802.11b $CW_{\min}$	31	Transmission power	11.0 dBm
IEEE 802.11b $CW_{\max}$	1023	Data rate	11.0 Mbps
		PLCP preamble	Short preamble (72 bits)
		Synchronization time	96 $\mu$ s
		Propagation delay	1 $\mu$ s
		Constant shadowing mean	4 dB

(b) Sticky CSMA/CA

Sticky CSMA/CA parameters	Values
AIFS (Voice)	50 $\mu$ s
AIFS (Data)	70 $\mu$ s
$(CW_{\min}, CW_{\max})$ voice	(3, 7)
$(CW_{\min}, CW_{\max})$ high priority data	(15, 31)
$(CW_{\min}, CW_{\max})$ background data	(31, 63)
R-RTS	26 bytes
R-CTS	20 bytes
MAC data header	30 bytes
Cycle time	20 ms
Missed feedback threshold	10

(c) IEEE 802.11e

IEEE 802.11e parameters	Values
AIFS (Voice)	50 $\mu$ s
AIFS (Background data)	150 $\mu$ s
$(CW_{\min}, CW_{\max})$ voice	(7, 15)
$(CW_{\min}, CW_{\max})$ background data	(31, 1023)
TXOP limit for voice	3008 $\mu$ s

or are commonly used by the state-of-the-art equipment available in the market [15,6,4].

To simulate the bi-directional VoIP calls, we use two simultaneous constant bit rate (CBR) flows between the source and the destination nodes, one in each direction. The network layer is assumed to support priority queuing of voice packets over the background data traffic. We consider an ITU-T G.711 codec packetizing 20 ms of audio data in 160 bytes of payload (data rate 64 Kbps). We model call arrival as a Poisson process. Table 2(b) lists parameter values specific to Sticky CSMA/CA used in the simulations. In order to prioritize real-time traffic over data packets, we set the contention window parameters ( $CW_{min}$ ,  $CW_{max}$ ) for voice setup requests as (3, 7), for high priority data as (15, 31), and for background data as (31, 63). The Arbitration Interframe Space (AIFS) is 50  $\mu$ s for voice and high priority data, and 70  $\mu$ s for background data traffic. These parameters ensure a higher priority to the voice traffic than the data traffic. The carrier sense history information is maintained for the past six cycles. A sender node that does not receive a feedback message from the receiver for 10 consecutive cycles (Missed Feedback Threshold) attempts slot reconfiguration. Slot reconfiguration is also attempted when the feedback message sent by the receiver indicates more losses than a certain threshold (10–20% in our simulations).

We evaluate the performance of Sticky CSMA/CA and compare it with IEEE 802.11b and IEEE 802.11e in terms of the total number of VoIP calls supported with the required QoS, i.e., bounded packet loss, average delay, and delay jitter. We explore the impact of background traffic (both TCP and UDP) on the call carrying capacity and estimate the achievable throughput for these flows. Note that we do not address the starvation of background data flows due to voice traffic in our simulations, since the objective is to understand the effect of these flows on the voice call traffic and vice-versa. Our results can be used in practical systems to determine the maximum number of voice calls that can be admitted, while sustaining a given data throughput. Call admission control, resource constrained routing, and end-to-end signaling for call establishment are higher layer functionalities that are beyond the scope of this paper.

We impose a delay constraint of 50 ms for voice packets, since this constraint allows ample time for the transfer of voice packets over the backbone network to ensure a total end-to-end delay of 150 ms.

We consider a VoIP call as useful only if 95% of the packets sent in both directions are delivered within the delay bound. We use these constraints to model the QoS requirements of VoIP calls [23]. The simulation results have been averaged over 10 runs.

We generate the irregular grid topology by uniformly varying the placement of nodes within a square of side 40 m around the corresponding positions in the regular grid topology. This arrangement of nodes ensures that, with a high probability, neighboring nodes are within the TR of each other so that links between them still exist, but with different link qualities. We use a different uniform random grid topology with each simulation run.

## 5.2. Line topology

Consider a mesh network that consists of a single line of five nodes as shown in Fig. 8. Nodes 1 and 5 act as gateways for the network. All the calls are routed out of the network through these gateways. Thus, one of the two gateways in the network is always the terminal point of the CBR flows corresponding to a VoIP call in the network.

As mentioned in Section 2, IEEE 802.11e has an optional feature, called Transmission Opportunity (TXOP), that allows contention-free bursting for the corresponding access category (AC) after a node has won medium contention. TXOP means that a node that has won medium contention can continuously send data up to a maximum allowed transmission duration for that AC without having to contend again. As shown in Fig. 10, over the line topology, the total number of calls supported by

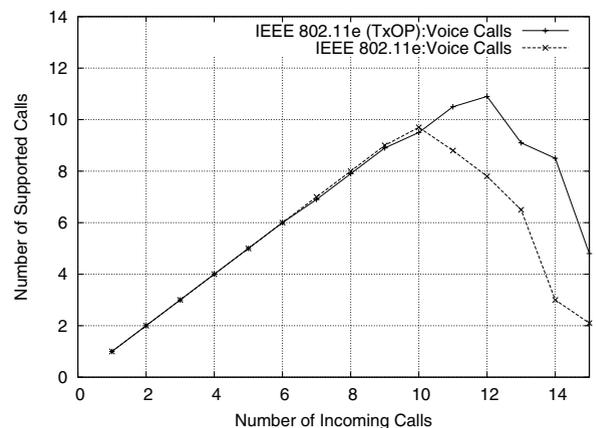


Fig. 10. The effect of TXOP on the total number of voice calls supported over a five node line topology.

IEEE 802.11e improves from 9 to 11 with TXOP. We incorporate TXOP support in our IEEE 802.11e simulations to increase the IEEE 802.11e voice call carrying capacity.

Fig. 11a shows the average number of voice calls supported as a function of the number of incoming calls. We generate voice calls by randomly picking the nodes (other than gateways) in the network and assigning the nearest gateway as the terminal node for each call. Hence, calls traverse either a single hop or multiple hops to the gateway depending on the source node. Each data point in Fig. 11a corresponds to the average number of calls supported for a fixed number of incoming calls. Observe from Fig. 8 that Node 3 can carrier sense all the nodes in the network. Since the links from Node 3 are effectively shared by all the other nodes in the network, Node 3 is a bottleneck node. From Fig. 11a, we observe that the total number of calls supported by Sticky CSMA/CA is 16, whereas IEEE 802.11e

and IEEE 802.11b support a maximum of 11 and 9 calls respectively. The gains in the VoIP call carrying capacity for Sticky CSMA/CA result from reduced contention overhead, reduced average backoff time, and the elimination of per-packet ACKs. Notice that in the absence of a call admission control mechanism, the performance degrades when the number of incoming calls exceeds the total capacity of the network. Lack of resource reservation and medium contention for every packet transmission result in reduced total call carrying capacity for IEEE 802.11b/e.

Fig. 11b shows the average number of supported voice calls when there is background TCP traffic consisting of payloads of varying sizes (maximum segment size of 1460 bytes). In this evaluation scenario, we have two TCP flows in the network, originating from gateway nodes 1 and 5, and terminating at nodes 2 and 3, respectively. This TCP traffic scenario models the long-lived FTP and other TCP

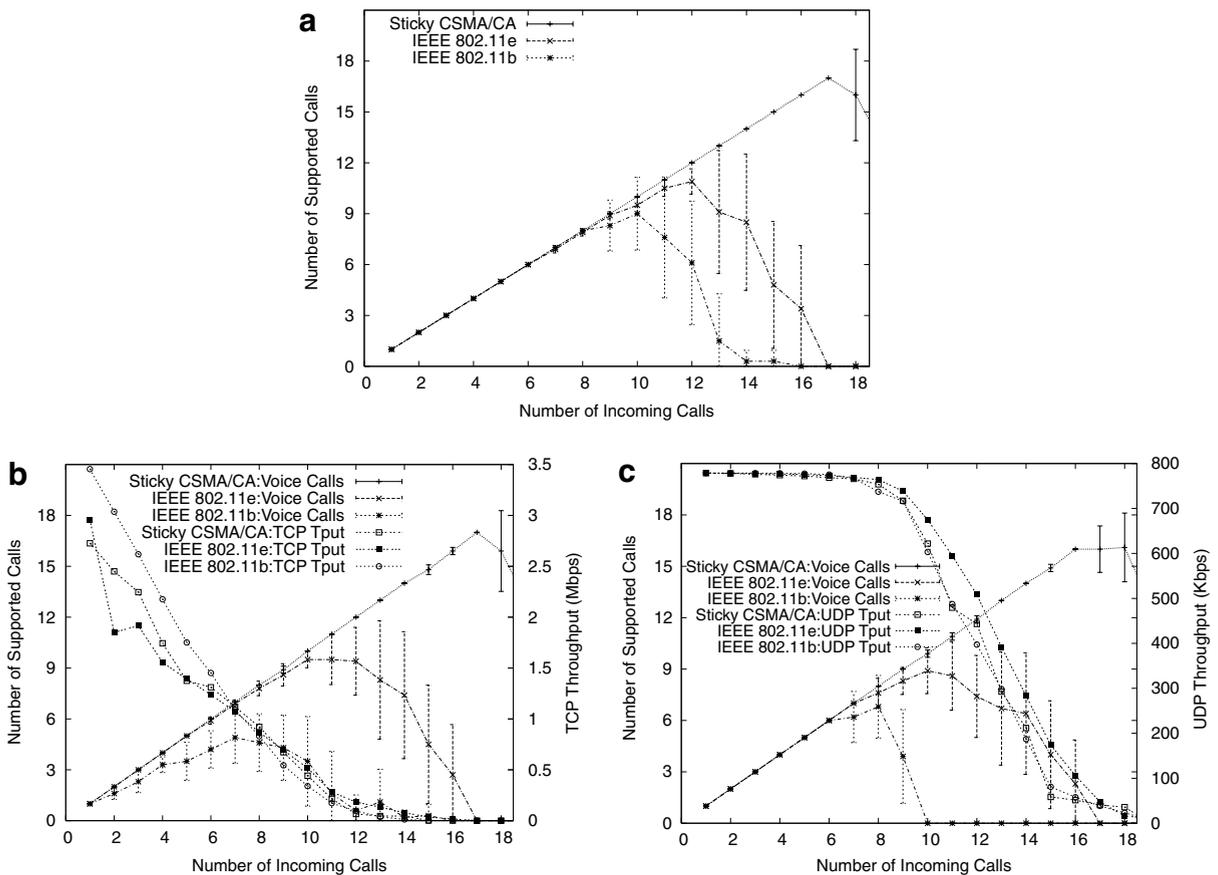


Fig. 11. Sticky CSMA/CA call carrying capacity for the line topology: (a) voice traffic only, (b) voice and TCP background traffic and (c) voice and UDP background traffic. All figures show 95% confidence intervals for the average number of supported calls.

based content download applications from the gateway nodes that connect to the Internet. The data traffic workload ensures that all the nodes in the network have backlogged low priority data queues, and that the nodes are directly affected by the background TCP traffic dynamics. The capacity of Sticky CSMA/CA, in terms of the total number of supported calls, remains the same even in the presence of background TCP traffic of varying payload sizes. This is because of the effective priority given to voice packets over data transmissions in the entire carrier sensing range (CSR) of a node. IEEE 802.11e (with TXOP) supports a higher number of voice calls as compared to IEEE 802.11b because of service differentiation, but the total number of calls supported by IEEE 802.11e is still significantly less than that supported by Sticky CSMA/CA. Due to interference and lack of prioritization in IEEE 802.11b, the number of supported voice calls reduces drastically with TCP traffic, since different nodes with voice or data packets contend with each other for medium access. For data traffic in Sticky CSMA/CA, the bandwidth left over by the real-time flows is sufficient to maintain average data throughput slightly lower than IEEE 802.11b but relatively higher than IEEE 802.11e.

With fewer voice calls in the network, the effect of interference caused by heavy data traffic in terms of loss of many calls for IEEE 802.11b is apparent in Fig. 11b. IEEE 802.11e is able to recover from these losses by multiple fast retransmissions of voice packets aided by shorter contention windows for voice. The TXOP feature helps in recovering from delays due to retransmissions by transmitting the accumulated voice packets together in bursts. Sticky CSMA/CA effectively copes with the interference caused by data traffic with the help of techniques discussed in Section 3.3.2, which results in approximately 200% additional call carrying capacity as compared to that of IEEE 802.11b, and 65% additional call carrying capacity as compared to that of IEEE 802.11e.

Fig. 11c shows the average number of supported voice calls in the presence of background UDP traffic. We use three UDP flows, each with a data rate of 256 Kbps and a payload size of 1460 bytes. These UDP flows originate from the three non-gateway nodes in the network, and are routed to the nearest gateway node. As in the TCP background traffic scenario, this data traffic workload ensures that all the nodes in the network have low priority data packets to send, and that they have to deal with the back-

ground UDP traffic dynamics to sustain voice calls without performance degradation. We observe from Fig. 11c that the voice call performance is not affected by the background traffic, and the capacity in terms of the total number of calls supported remains the same. UDP throughput for Sticky CSMA/CA starts to decrease after the number of incoming calls increases to an extent that the leftover gaps between the voice calls are not sufficient to sustain the UDP flows. Sticky CSMA/CA UDP throughput is comparable to that of IEEE 802.11b/e.

Fig. 12 shows the instantaneous TCP throughput for Sticky CSMA/CA with increasing number of incoming calls over the line topology. Observe that TCP throughput decreases as the number of calls increases, demonstrating that the priority mechanism provides effective service differentiation between background data traffic and voice. Data flows adapt to the voice traffic in the network depending on the leftover capacity.

### 5.3. Grid topologies

Consider a 25 node regular grid topology as shown in Fig. 9. All VoIP calls in the system are routed out of the network through the gateways. The gateways are also assumed to be the sources of long-lived TCP or UDP flows. Though the grid topology provides more opportunities for spatial reuse, it is also characterized by increased contention and interference among nodes. We compare the performance of Sticky CSMA/CA with IEEE 802.11b and IEEE 802.11e in terms of the total number of VoIP calls supported with the required

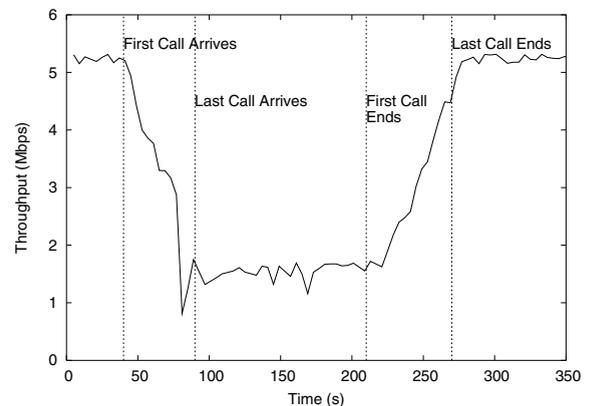


Fig. 12. Instantaneous throughput of a FTP flow in a representative simulation for sticky CSMA/CA.

QoS over the grid topology. We explore the impact of background traffic (both TCP and UDP) on the call carrying capacity.

Fig. 13a shows the average number of voice calls supported as a function of the number of incoming calls. We generate calls by picking nodes in the network at random and assigning the nearest gateway as the terminal node for each call. Ties between the gateways are broken arbitrarily. For UDP or TCP flows, the gateway nodes are chosen as the sources, and the destination nodes are randomly picked among the nodes serviced by the gateway. We see that the total number of calls supported by Sticky CSMA/CA is 25, whereas IEEE 802.11e supports a maximum of 18 calls and IEEE 802.11b supports only 15 calls. The increase in the number of calls supported by Sticky CSMA/CA and IEEE 802.11b/e over the grid topology is because of increased spatial reuse of the medium.

Fig. 13b shows the average number of supported voice calls as a function of the number of incoming calls in the presence of background TCP traffic with varying payload sizes (maximum segment size of 1460 bytes). In this scenario, we generate the background TCP traffic by initiating one TCP flow from each gateway node and choosing a random non-gateway node from its service area as the destination, routing packets via preconfigured shortest path routes. Thus, the background TCP traffic dynamics differ over different simulation runs with different seeds. We observe that the maximum call carrying capacity for Sticky CSMA/CA remains the same as that without the background TCP traffic. The effect of high interference in the grid topology is significantly more pronounced for IEEE 802.11b/e in terms of majority of incoming calls being dropped. Sticky CSMA/CA is able to cope with this increased interference except for a small

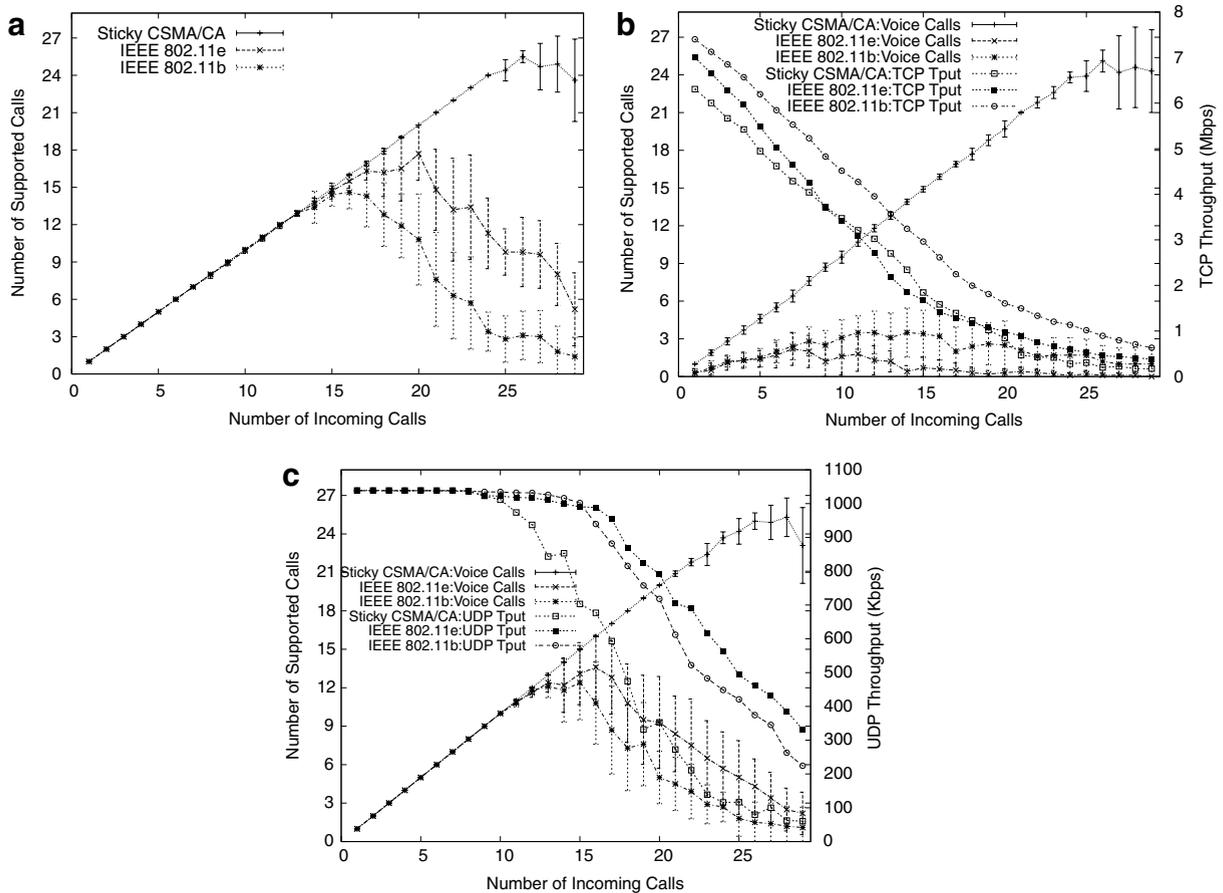


Fig. 13. Sticky CSMA/CA call carrying capacity for the regular grid topology: (a) voice traffic only, (b) voice and TCP background traffic and (c) voice and UDP background traffic. All figures show 95% confidence intervals for the average number of supported calls.

probability of call drop in some cases with fewer incoming calls. As the number of VoIP flows increases, the interference problem is alleviated as the time gaps between VoIP flows decrease, reducing the overall data throughput. The voice call carrying capacity of Sticky CSMA/CA in this scenario is many times higher than that of IEEE 802.11b/e.

The increased data throughput observed for IEEE 802.11b can be attributed to the lack of prioritization of voice over background data because of which data packets achieve much higher throughput at the expense of voice calls. This lack of service differentiation adds to the effect of increased interference in the grid topology and further degrades the voice call performance for IEEE 802.11b. With fewer calls in the network, TCP throughput for Sticky CSMA/CA is slightly less than that for IEEE 802.11e, but this increased throughput for IEEE 802.11e is at the expense of most of the incoming voice calls being dropped. Note that the voice call

performance of Sticky CSMA/CA does not change significantly in the presence of TCP packet transmissions of varying sizes. This demonstrates that the CS Tables maintained by the nodes in the network contain the relevant periodic high priority flow information and are not corrupted by the data packet transmissions even under heavy TCP traffic.

Fig. 13c shows the average number of supported calls when there is background UDP traffic in the network. The background UDP traffic is generated in a similar manner as the background TCP traffic, with each UDP flow having a data rate of 256 Kbps and a payload size of 1460 bytes. Again, a capacity gain is observed. The sharp fall in data throughput as compared to IEEE 802.11b/e after 10 voice calls can be attributed to the time gaps between the voice transmissions being too small for data packets to be transmitted. Note that the data throughput is approximately equal for Sticky CSMA/CA and

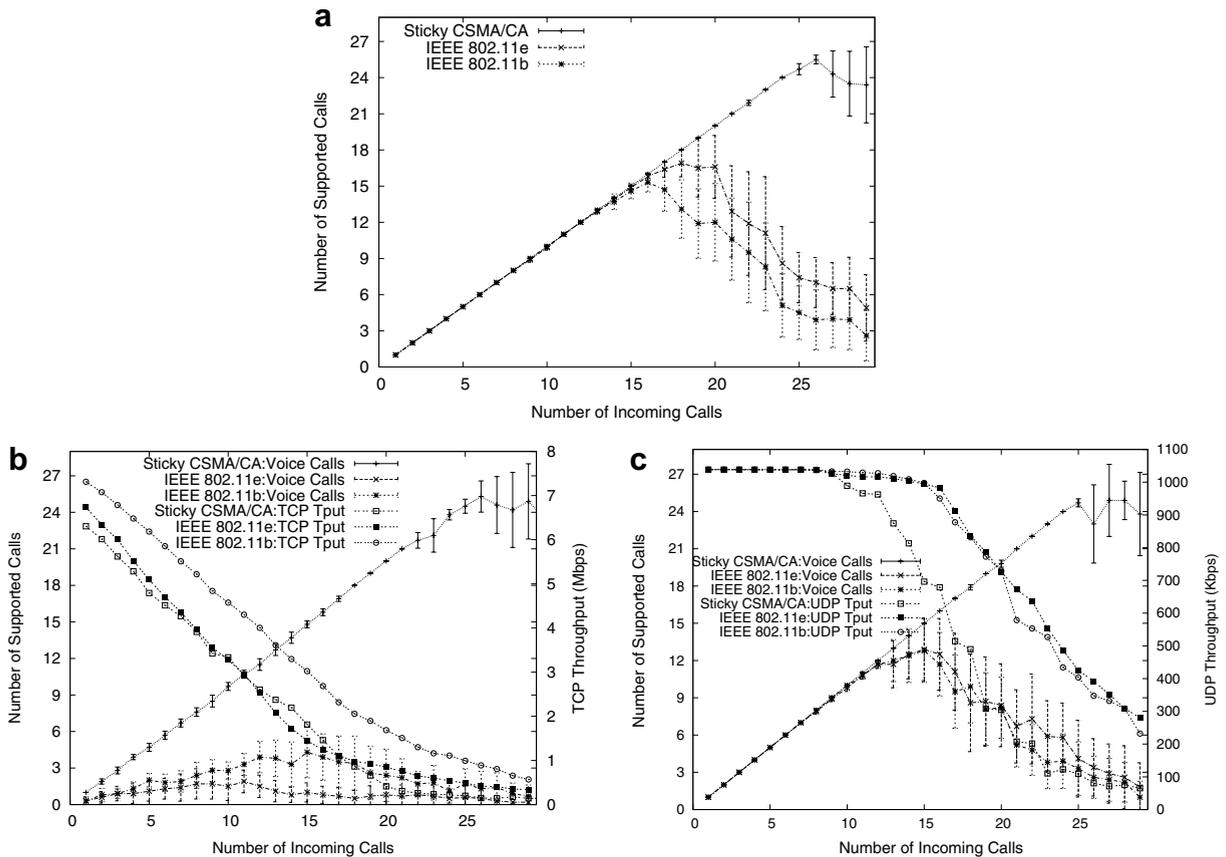


Fig. 14. Sticky CSMA/CA call carrying capacity for the irregular grid topology: (a) voice traffic only, (b) voice and TCP background traffic and (c) voice and UDP background traffic. All figures show 95% confidence intervals for the average number of supported calls.

IEEE 802.11b/e until the number of voice calls in the network reaches the maximum that can be supported by IEEE 802.11b/e. The loss of background data for more incoming calls is the cost of supporting significantly more high priority voice calls.

Fig. 14 illustrates performance comparison in terms of voice call carrying capacity of the network in voice only case, and in presence of background TCP/UDP traffic over the irregular grid topology. We observe that the results obtained for the irregular grid topology and the regular grid topology show very similar trends.

We make the following observations regarding the supported calls results for different simulation runs for all the test topologies: the variation of the results for different simulation seeds is minimal until the number of calls approaches the system’s call-carrying capacity for Sticky CSMA/CA and IEEE 802.11b/e for the voice only case. A similar trend is observed for voice call results with UDP background traffic, except that the results start to vary earlier (by approximately two calls) than the voice only case. For TCP background traffic, the variation of the number of supported calls is higher than that observed for the voice only and the UDP background traffic cases for all the protocols. Interestingly, for both IEEE 802.11b and 802.11e over the regular and irregular grid topologies, the most dominant result for all data points (over most seeds) is zero calls, and the deviations are non-zero numbers that account for the very low average call carrying capacity for both IEEE 802.11b/e. This is because of high interference and contention that result in heavy packet losses for both IEEE

802.11b/e. Also, Sticky CSMA/CA voice call results vary comparatively more than the voice only case for low number of incoming calls (up to five) because of increased interference effects from background data traffic.

5.4. QoS performance: delay and delay jitter

We compare the QoS performance of Sticky CSMA/CA and IEEE 802.11b/e in terms of the average end-to-end delay and delay jitter for voice calls. Figs. 15–17 illustrate the average end-to-end delay and delay jitter for Sticky CSMA/CA and IEEE 802.11b/e for the line topology, the regular grid topology and the irregular grid topologies, respectively. The traffic in the network consists of only voice calls. The delay performance for Sticky CSMA/CA remains fairly constant with increasing number of voice calls. We notice the steep rise in delay for IEEE 802.11b and IEEE 802.11e when the number of incoming calls exceeds the network capacity. Because of the TDM-like synchronization achieved by Sticky CSMA/CA, the average delay jitter is orders of magnitude less compared to that obtained for IEEE 802.11b/e.

Performance trends of Sticky CSMA/CA in terms of the voice call carrying capacity and the offered QoS are very similar over the line topology, the regular grid topology and the irregular grid topologies. This observation implies that the performance of Sticky CSMA/CA scales over more general, irregular mesh network topologies. In order to gain more insight into Sticky CSMA/CA

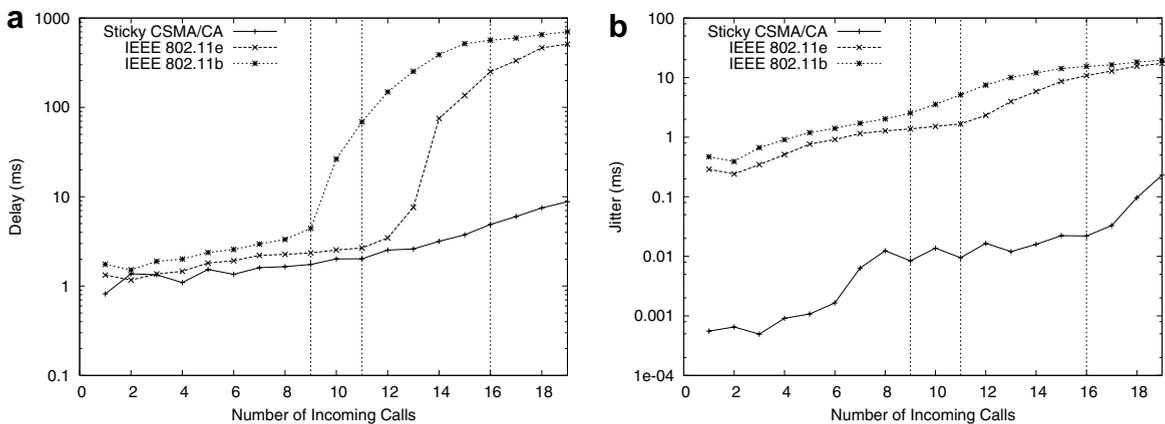


Fig. 15. Average end-to-end delay and delay jitter for voice calls over the line topology: (a) end-to-end delay and (b) delay jitter.

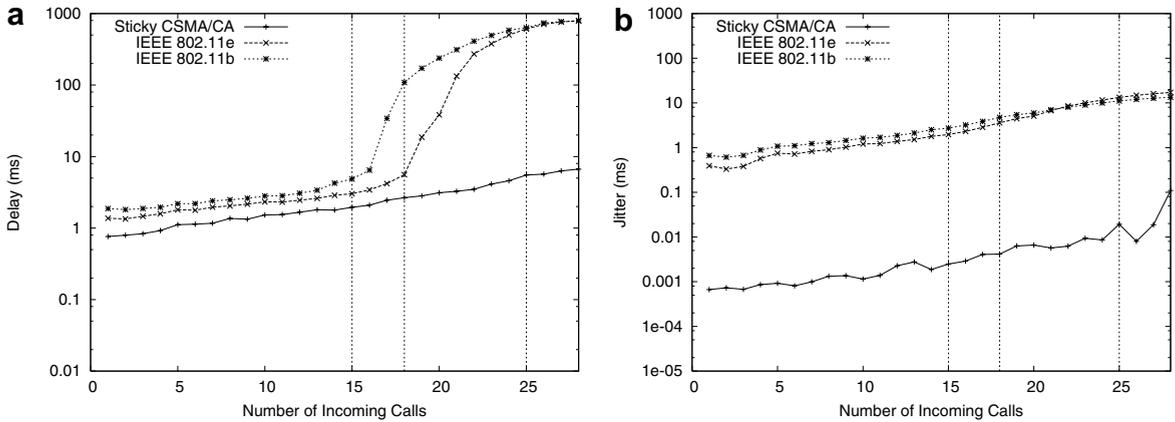


Fig. 16. Average end-to-end delay and delay jitter for voice calls over the regular grid topology: (a) end-to-end delay and (b) delay jitter.

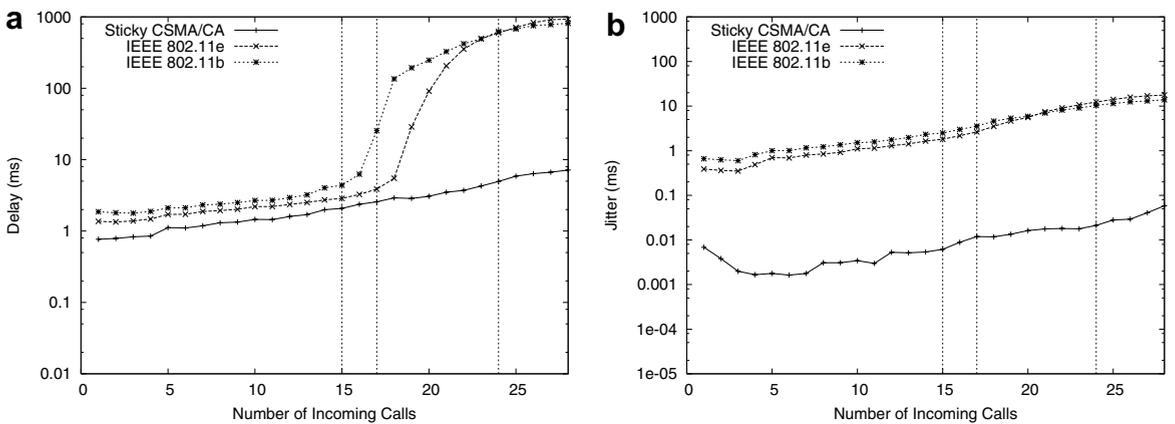


Fig. 17. Average end-to-end delay and delay jitter for voice calls over the irregular grid topology: (a) end-to-end delay and (b) delay jitter.

performance over different practical mesh network scenarios, we next study the effect of increasing number of hops on Sticky CSMA/CA and 802.11b/e voice call performance. We then look at the voice call performance over networks employing silence suppression to optimize bandwidth consumption of voice calls.

5.5. Effect of increasing number of hops

In order to evaluate the effect of paths with large number of hops on Sticky CSMA/CA voice call performance, we consider a mesh network with nine nodes arranged in a single-line topology (see Fig. 18). We include background TCP traffic to model realistic network traffic loads. The voice call and data traffic load generation methodology is similar to that used for the previous evaluations in this



Fig. 18. Nine node line topology: the edge nodes 1 and 9 act as gateway nodes.

section. Thus, there are voice calls that traverse two, three or four hops over the network depending on the nodes chosen as sources of voice calls for each simulation run. Note that for calls traversing four hops, the gateway node is out of the carrier sense range of the source node, and thus it does not have complete carrier sensing information of the VoIP call in its CS Table. There are two TCP flows in the network which traverse three hops (with gateway nodes 1 and 9 as sources and nodes 4 and 6 as destinations). Fig. 19 compares the total number

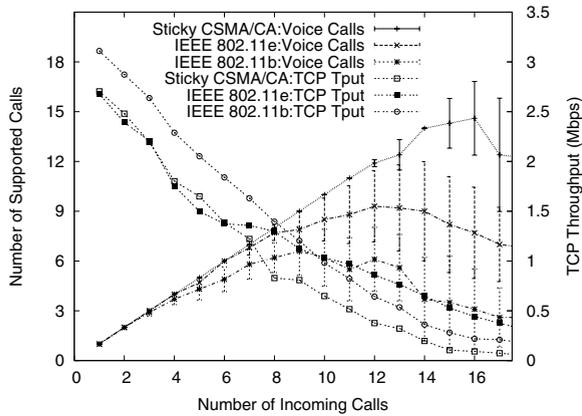


Fig. 19. Voice call support for nine node single line topology network.

of supported calls and aggregate TCP throughput for Sticky CSMA/CA and IEEE 802.11b/e. We observe that Sticky CSMA/CA has a higher call carrying capacity in this scenario. This is consistent with the results reported for other network topologies, and shows that Sticky CSMA/CA performs better than IEEE 802.11b/e even for paths with more hops.

### 5.6. Silence suppression

Silence suppression is an important configurable feature that is used to reduce the bandwidth usage of VoIP flows. End nodes that employ silence suppression do not send packets during the silence periods in the VoIP call thereby saving bandwidth. Many common audio codec standards along with the Realtime Transport Protocol (RTP) support

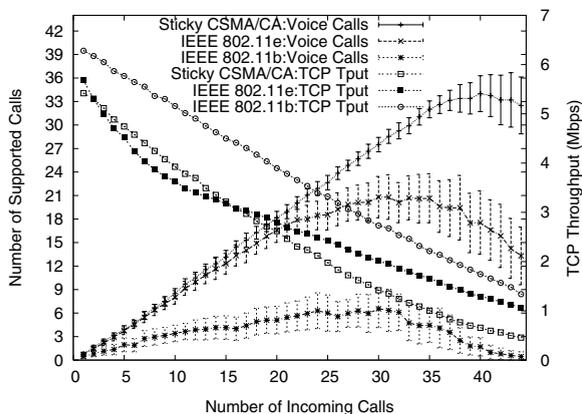


Fig. 20. Voice call support for regular grid topology network with silence suppression enabled.

silence suppression [24]. We compare the performance of Sticky CSMA/CA with IEEE 802.11b/e protocols over an example scenario of the regular grid topology with background TCP traffic. We use Brady's On-Off Markov Modulated Fluid model [25] to model voice calls with talk spurts and silence periods. Brady's model has been commonly used in the VoIP literature because of its simplicity and ability to capture the dynamics of telephonic conversations with reasonable accuracy. In this model, the silence periods and the talk spurt periods of a voice source are exponentially distributed random variables with means of  $1.35S$  and  $IS$ , respectively. The probability  $p$  that the sending end is active is given by 0.43.

Fig. 20 compares the voice call carrying capacity of Sticky CSMA/CA with IEEE 802.11b/e when silence suppression is used. We observe that Sticky CSMA/CA supports a higher number of voice calls. The underlying reason for this performance gain is that Sticky CSMA/CA is able to exploit the predictability of packet transmissions over talk spurt durations, thereby reducing medium contention and packet loss due to collisions.

## 6. Discussion and conclusions

In this paper, we have introduced Sticky CSMA/CA, a MAC framework that can exploit regularity in real-time traffic to achieve implicit synchronization over wireless mesh networks, thereby enabling TDM-like QoS for such applications. For the mix of voice and data traffic considered in our numerical results, Sticky CSMA/CA locks onto a periodic transmission schedule that matches the period of the VoIP flows. This periodicity in transmission enables each node to accurately predict the real-time traffic generated by nodes that are capable of interfering with it, thereby drastically reducing the contention seen by real-time flows. Thus, Sticky CSMA/CA mechanism provides quasi-deterministic QoS to a real-time flow once it is established, while its slot reconfiguration mechanism enables robust recovery from losses due to unexpected interference patterns resulting from hidden terminals. Through analysis and simulations, we have shown that Sticky CSMA/CA significantly increases the total voice call carrying capacity of mesh networks as compared to the conventional CSMA/CA based schemes such as IEEE 802.11b/e. While we impose a strict priority for real-time flows over delay-insensitive data traffic, the new contention mechanisms

introduced for data packet transmissions ensure that the data traffic effectively fills the bandwidth left over after serving the VoIP flows. Our simulation results for different scenarios show that the total voice call carrying capacity of Sticky CSMA/CA is much higher than that of IEEE 802.11b/e, while its data throughput remains comparable to that of IEEE 802.11b/e.

It is worth pointing out that, while IEEE 802.11e is intended to provide QoS to real-time flows by giving them priority over delay-insensitive traffic, this service differentiation has little effect when a number of real-time flows of equal priority contend for the medium. In particular, our performance evaluations over the grid topology (characterized by high interference and contention for the medium) demonstrate that IEEE 802.11e yields much lower voice call carrying capacity and higher delay/delay jitter than Sticky CSMA/CA. In contrast, Sticky CSMA/CA achieves fine-grained resource allocation among nodes that have real-time packets to transmit, while enforcing strict service differentiation among real-time and delay-insensitive traffic.

While our performance evaluation was restricted to VoIP flows that naturally exhibit periodicity that can be exploited by Sticky CSMA/CA, the promising results that we obtain motivate research on extending these ideas to real-time flows that may not be naturally periodic. In particular, our ongoing research focuses on efficient support of variable bit rate (VBR) traffic such as video using Sticky CSMA/CA. We propose to impose a basic unit of resource allocation in the network as a periodic connection with packets of a fixed size, thus satisfying the basic requirements for Sticky CSMA/CA: implicit synchronization and prediction of network activity via Carrier Sense Tables. Nodes initiate and terminate such connections based on their bandwidth needs, analogous to the arrival and departure of voice connections in the scenario considered in this paper. The specific criteria for opening and closing connections depends on the applications being supported. For VBR video, for example, it may be possible to use an approach similar to Renegotiated CBR, which was proposed in [30] for supporting video over ATM networks. This approach is based on splitting a VBR flow into a base CBR stream with minimal bandwidth, and renegotiating for additional bandwidth in terms of additional CBR streams whenever required. The buffer occupancy levels at the application layer can be used to deduce the need for additional bandwidth and to estimate

the bandwidth required. Significant cross-layer interaction is required to be able to provide a deterministic QoS. This interaction includes admission control to decide whether or not a new VBR flow should be admitted while ensuring that bandwidth renegotiations for existing flows are successful, end-to-end signaling for flow establishment and bandwidth renegotiations, and buffering at the application layer along with an intelligent algorithm to determine if bandwidth renegotiation is required for an existing VBR flow. The efficacy of such techniques for supporting VBR video over wireless mesh networks is the subject of our current research.

More broadly, Sticky CSMA/CA could potentially be used as a fundamental building block of a cross-layer architecture to provide comprehensive QoS solutions for wireless mesh networks. Specifically, Sticky CSMA/CA can be exploited by intelligent routing, transport and application layer schemes to provide deterministic QoS guarantees to applications with different QoS requirements. For example, a wireless mesh routing protocol can leverage the medium utilization information maintained in the CS Tables of nodes to search for QoS capable routes that can help in efficient resource utilization. Admission control algorithms can utilize the CS Table maintained at each node to obtain an accurate estimate of the available resources in the network, to be able to decide whether a new flow should be admitted, and to provide deterministic QoS guarantees to existing flows. In order to enforce the desired resource allocation among different traffic classes and to prevent starvation, thresholds on maximum allocated bandwidth for each traffic class can be decided by the network designer and can be enforced through admission control schemes. By enabling local (per-hop) resource reservations, supporting service differentiation, and providing an estimate of available resources in the network, Sticky CSMA/CA supports QoS constrained routing and admission control, which are essential components of any QoS architecture.

### Acknowledgments

This work was supported by the National Science Foundation under grants ANI-0220118 and NSF NeTS Award CNS-0435527, and by the Institute for Collaborative Biotechnologies under grant DAAD19-03-D-0004 from the US Army Research Office.

## References

- [1] Y. Sun, I. Sheriff, E.M. Belding-Royer, K.C. Almeroth, An experimental study of multimedia traffic performance in mesh networks, in: WiTMeMo'05: Workshop on Wireless Traffic Measurements and Modeling, Seattle, WA, 2005.
- [2] F. Kelly, Modelling communication networks, present and future, *Philosophical Transactions of the Royal Society of London A354* (1707) (1996) 437–463.
- [3] T. Henderson, D. Kotz, I. Abyzov, The changing usage of a mature campus-wide wireless network, in: Proc. 10th Annual International Conference on Mobile Computing and Networking (MobiCom'04), Philadelphia, PA, 2004.
- [4] Cisco 7900 Series IP Phones (2006). URL <http://www.cisco.com/en/US/products/hw/phones/ps379/>.
- [5] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std. 802.11, January 1997.
- [6] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, IEEE Std. 802.11e-2005, November 2005.
- [7] J.L. Sobrinho, A.S. Krishnakumar, Real-time traffic over the IEEE 802.11 medium access control layer, *Bell Labs Technical Journal* 1 (2) (1996) 172–187.
- [8] N.H. Vaidya, P. Bahl, S. Gupta, Distributed fair scheduling in a wireless LAN, in: Proc. 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, 2000.
- [9] S.J. Golestani, A self-clocked fair queueing scheme for broadband applications, in: Proc. IEEE INFOCOM'94, Toronto, Canada, 1994.
- [10] C.R. Lin, M. Gerla, Asynchronous multimedia multihop wireless networks, in: Proc. IEEE INFOCOM'97, Kobe, Japan, 1997.
- [11] A. Rao, I. Stoica, An overlay MAC layer for 802.11 networks, in: MobiSys'05: Proc. of the 3rd International Conference on Mobile Systems, Applications, and Services, Seattle, WA, 2005.
- [12] I. Rhee, A. Warrier, M. Aia, J. Min, Z-MAC: a hybrid MAC for wireless sensor networks, in: SenSys'05: Proc. of the 3rd International Conference on Embedded Networked Sensor Systems, San Diego, CA, 2005.
- [13] C.-H. Lin, H. Dong, U. Madhow, A. Gersho, Supporting real-time speech on wireless ad hoc networks: inter-packet redundancy, path diversity, and multiple description coding, in: Proc. 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH'04), Philadelphia, PA, 2004.
- [14] A. Vasan, R. Ramjee, T. Woo, ECHOS enhanced capacity 802.11 Hotspots, in: Proc. IEEE INFOCOM'05, Miami, FL, 2005.
- [15] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher Speed Physical Layer (PHY) Extension in the 2.4GHz Band, IEEE Std. 802.11b/D5.0, April 1999.
- [16] K. Sanzgiri, I.D. Chakeres, E.M. Belding-Royer, Determining intra-flow contention along multihop paths in wireless networks, in: Proc. 1st Annual International Conference on Broadband Networks (BROADNETS'04), San Jose, CA, 2004.
- [17] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, *IEEE Journal on Selected Areas in Communications* 18 (3) (2000) 535–547.
- [18] Y.C. Tay, K.C. Chua, A capacity analysis for the IEEE 802.11 MAC protocol, *Wireless Networks* 7 (2) (2001) 159–171.
- [19] H.S. Chhaya, S. Gupta, Performance modeling of asynchronous data-transfer methods of IEEE 802.11 MAC protocol, *Wireless Networks* 3 (3) (1997) 217–234.
- [20] M.M. Carvalho, J.J. Garcia-Luna-Aceves, A scalable model for channel access protocols in multihop ad hoc networks, in: Proc. 10th Annual International Conference on Mobile Computing and Networking (MobiCom'04), Philadelphia, PA, 2004.
- [21] Qualnet Network Simulator, version 3.9 (2005). URL <http://www.scalable-networks.com>.
- [22] MeshDynamics-High Performance Metro Mesh Networks with Structured Mesh (TM) (2006). URL <http://www.meshdynamics.com>.
- [23] A.P. Markopoulou, F.A. Tobagi, M.J. Karam, Assessing the quality of voice communications over internet backbones, *IEEE/ACM Transactions on Networking* 11 (5) (2003) 747–760.
- [24] R. Zopf, Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN), RFC 3389 (September 2002).
- [25] P.T. Brady, A model for generating on-off speech patterns in two-way conversation, *The Bell System Technical Journal* 48 (9) (1969) 2445–2472.
- [30] M. Grossglauser, S. Keshav, D.N.C. Tse, RCBR: a simple and efficient service for multiple time-scale traffic, *IEEE/ACM Transactions on Networking* 5 (6) (1997) 741–755.



**Sumit Singh** received the bachelors degree in Electrical Engineering from the Indian Institute of Technology, Bombay, in 2002. He worked as a software engineer in the Network Systems Division at Samsung Electronics, Bangalore from 2002 to 2004. He is currently pursuing the Ph.D. degree in Electrical and Computer Engineering at the University of California, Santa Barbara, where he is working with Prof. Upamanyu Madhow and Prof.

Elizabeth M. Belding-Royer. His research interests include quality of service and scalability issues in wireless networks.



**Prashanth Aravinda Kumar Acharya** received his B.E. degree in Computer Engineering from National Institute of Technology, Suratkal, India in 2002. From 2002 to 2004, he was with Texas Instruments, Bangalore, India as a Software Design Engineer. He is currently pursuing his Ph.D. in Computer Science at the University of California, Santa Barbara. He is a member of the Moment Research Lab directed by Prof.

Elizabeth M. Belding-Royer. His research interests include Quality of Service, multi-media streaming and multi-radio wireless mesh networks.



**Upamanyu Madhow** received his bachelor's degree in electrical engineering from the Indian Institute of Technology, Kanpur, in 1985. He received the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois, Urbana-Champaign in 1987 and 1990, respectively.

From 1990 to 1991, he was a Visiting Assistant Professor at the University of Illinois. From 1991 to 1994, he was a research scientist at Bell Communications Research, Morristown, NJ. From 1994 to 1999, he was on the faculty of the Department of Electrical and Computer Engineering at the University of Illinois, Urbana-Champaign. Since December 1999, he has been with the Department of Electrical and Computer Engineering at the University of California, Santa Barbara, where he is currently a Professor. His research interests are in communication systems and networking, with current emphasis on wireless communication, sensor networks and multimedia security.

He is a Fellow of the IEEE, and recipient of the NSF CAREER award. He has served as an Associate Editor for

Spread Spectrum for the IEEE Transactions on Communications, and as an Associate Editor for Detection and Estimation for the IEEE Transactions on Information Theory.



**Elizabeth M. Belding-Royer** is an Associate Professor in the Department of Computer Science at the University of California, Santa Barbara. Her research focuses on mobile networking, specifically ad hoc and mesh networks, multimedia, monitoring, and advanced service support. She is the author of over 50 papers related to mobile networking and has served on over 40 program committees for networking conferences. She was

the TPC Co-Chair of ACM MobiCom 2005 and IEEE SECON 2005, and is currently on the editorial board for the IEEE Transactions on Mobile Computing. She is the recipient of an NSF CAREER award, and a 2002 Technology Review 100 award, awarded to the world's top young investigators. See <http://www.cs.ucsb.edu/~ebelding> for further details.