

SoftMAC: Layer 2.5 MAC for VoIP Support in Multi-hop Wireless Networks

Haitao Wu¹, Xin Wang², Yunxin Liu¹, Qian Zhang¹, Zhi-Li Zhang³

¹Microsoft Research Asia (MSRA), Beijing, China, {t-hwu, yunliu, qianz}@microsoft.com

²Tsinghua University, China, wangxin02@mails.tsinghua.edu.cn

³University of Minnesota, zhzhang@cs.umn.edu

Abstract- In this paper, we present the challenges in supporting VoIP services over multi-hop wireless networks using commercial IEEE 802.11 MAC DCF hardware, and propose a novel software solution, called Layer 2.5 SoftMAC. Our proposed SoftMAC resides between the 802.11 MAC layer and IP layer to coordinate the real-time and best-effort packet transmission among neighboring nodes in a multi-hop wireless network. To effectively support VoIP services, our SoftMAC architecture employs three key mechanisms: 1) distributed admission control for regulating the load of real time-traffic, 2) rate control for minimizing the impact of best-effort traffic on real-time traffic, and 3) non-preemptive priority queuing for providing high priority service to VoIP traffic. To evaluate the efficacy of these mechanisms, we conduct extensive simulations using the network simulator NS2. We also implement our proposed SoftMAC as a Windows Network Driver Interface Specification (NDIS) driver over Network Interface Card (NIC) driver, and build a multi-hop wireless network testbed with 32 wireless nodes equipped with 802.11 a/b/g combo cards. Our evaluation and testing results demonstrate the effectiveness of our proposed software solution.

1. Introduction

Due to their low costs, ease of deployment, increased coverage, and enhanced capacity (e.g., via spatial reuse), multi-hop wireless networks such as mesh networks that utilize inexpensive and readily available 802.11 wireless interfaces are touted as the new frontier of wireless networking. In addition to traditional data services, multi-hop wireless networks have the potential to deliver exciting new real-time services such as Voice over IP (VoIP) [1], streaming music or video, providing a competitive alternative to cellular networks, in particular, in areas where the latter are not available. However, there are several challenges in effectively realizing real-time services over multi-hop wireless networks.

First of all, unlike cellular networks where bandwidth needed for a voice flow is reserved (e.g., via CDMA), in an 802.11-based wireless network, all nodes share and compete for the same media (spectrum). Hence transmissions from neighboring nodes may interfere with each other, causing collision. IEEE 802.11 (in DCF—distributed coordination function—mode) employs CSMA/CA based media access control (MAC) to reduce collisions. It works well in wireless LAN (WLAN), but poorly in multi-hop wireless network [9], where collisions happen due to the transmitters are out the carrier sense range of each others. Meanwhile, although VoIP payloads themselves consume relatively small amount of air time, the overheads introduced by 802.11 DCF MAC header

and PHY preamble, MAC ACK, and collision avoidance can be fairly significant, consuming valuable wireless capacity. This problem is further compounded in a multi-hop wireless network where packets are relayed across multiple hops, at each of which they may experience interference or collision, resulting in enlarged air time cost on wireless link. Without carefully controlled traffic volume to reduce collision, a multi-hop wireless network deploying DCF can be easily overwhelmed.

Furthermore, real-time services such as VoIP are likely to co-exist with data services over a multi-hop wireless network. *Delay-sensitive* real-time (RT) traffic such as VoIP packets must compete with delay-insensitive “best-effort” (BE) data traffic for access to shared media using 802.11 DCF MAC. Although QoS-enhanced 802.11 MAC mechanism, eDCF, has been developed for WLAN, which works by appropriately controlling Contention Window (CW) and Inter-Frame Spacing (IFS), it does not provide adequate service differentiation for support of real-time traffic in multi-hop wireless networks, because of the hidden terminal and other interference problems. Thus, the prioritized traffic volume regulation is necessary to provide service differentiation to VoIP, e.g., reduce contention of VoIP by cutting down interfering BE traffic.

To support delay-sensitive real-time services—in particular, VoIP, the focus of our study—over 802.11-based multi-hop wireless networks, it requires accurate contention and interfering traffic information. Such information is unlikely being obtained from implicit overhearing or adaptively try and error. Thus, in this paper we argue that it is necessary to regulate and control transmission of VoIP packets and BE data packets for reducing collision and meeting QoS requirements by explicit *coordination* among nodes. The challenging problem is *to implement the necessary coordination among nodes in a distributed manner using the “off-the-shelf” standard 802.11 MAC interfaces, namely, without requiring modification of existing hardware.*

To address this problem, we propose a novel unified software framework for distributed coordination, called *Layer 2.5 SoftMAC*, which resides between the standard 802.11 MAC layer (Layer 2) and IP layer (Layer 3) to regulate and control the amount of traffic (both real-time and BE) delivered to 802.11 DCF MAC interfaces. The proposed *Layer 2.5 SoftMAC* consists of three main components: i) a distributed

This work was finished when Xin Wang was visiting MSRA as intern.

Zhi-Li Zhang is supported in part by the National Science Foundation under the grants ITR-0085824 and CNS-0435444.

admission control (AC) module ii) a rate control (RC) module and iii) a priority queueing module. The AC module regulates the amount of VoIP traffic that is “admissible” in a “neighborhood” by exchanging traffic load information among neighboring nodes and by measuring channel conditions such as collision rate; it also “reserves” bandwidth for a VoIP flow along its path by piggybacking bandwidth reservation information in route queries and replies used in on-demand routing protocols such as DSR [34]. The RC module controls transmission of BE traffic so that the collision probability and impact to real time traffic on other nodes is under control. In priority queueing module, non-preemptive priority is always provided to VoIP traffic at each node. A key feature of our Layer 2.5 SoftMAC is that it achieves distributed coordination *without* requiring neither tight clock synchronization nor fine-grained transmission scheduling among neighboring nodes, both of which are difficult to implement in multi-hop wireless networks. Considering the variable air time cost for a payload on wireless link due to time varying collision and transmission error probability as well as link capacity, our distributed AC and RC algorithms employ the novel notion of “*fraction of air time*” to regulate the admissible traffic loads among neighboring nodes and control the transmission of real-time and BE packets. We implement our Layer 2.5 SoftMAC as a NDIS driver in Windows OS platforms. The software based design can be ported to other OS systems and is capable of handling new hardware interfaces and MAC mechanisms (e.g., 802.11e) with only software upgrade.

The rest of this paper is organized as following. Section 2 briefly describes the problems in support of VoIP over multi-hop wireless networks. We present our Layer 2.5 SoftMAC architecture in Section 3. The novel notion fraction of air time is illustrated in Section 4, and the detailed schemes and algorithms are described in Section 5. We discuss the system design and implementation of the SoftMAC architecture in Section 6, and present simulation evaluation and experimental testing results in Section 7. The related work is discussed in Section 8, and finally, Section 9 concludes the paper.

2. Challenges in Support of VoIP Services

In this section we use examples to illustrate the challenges in providing VoIP services in multi-hop wireless networks, in particular, the need for explicit coordination among neighboring nodes to control the network load and regulate real-time and BE traffic in support of VoIP.

As mentioned earlier, collision due to interference is a major issue that affects the performance of wireless networks. Collision introduces larger air time cost and delay in packet transmission due to the exponential back-off mechanism used in 802.11 DCF. Meanwhile, the time varying transmission error probability on wireless link and auto rate selection makes the link capacity change dynamically. To quantify the time varying cost, we introduce the notion of *air time*, i.e., the time cost introduced at physical medium to deliver packet from a source node to a destination node. In addition to the actual packet transmission time, it also includes the “overhead” time for carrier sensing, back-off, MAC ACK, retransmission, etc.

Clearly, because of the CSMA/CA and back-off mechanisms used in 802.11 DCF, the air time of a packet delivered over a wireless channel is determined by how busy the channel is as well as the number of collisions it experiences. Due to real-time nature of VoIP packets, their end-to-end delay must be bounded. The delay consists of two parts: queueing(buffering) and (re)transmission. Both require control of the network load to reduce the channel “busy time” and the collision rate. To illustrate, we use measurement data from a simple example conducted using the NS-2 simulator [33].

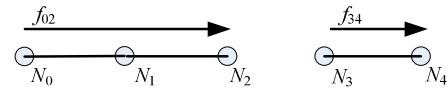


Fig.1 Example for admission control and service differentiation

Fig.1 shows a simple five-node wireless network, where each node is equipped with an 802.11b NIC and the channel rate is 11Mbps. For simplicity, we assume that the communication range is equal to the interference range. We use $f_{(i,j)}$ to denote the flow from node N_i to node N_j , and use $l_{(i,j)}$ to denote the link from node N_i to node N_j . Note for easy illustration, we use real time flow with arbitrary packet size and rate instead of multiple VoIP flows. We consider the following two scenarios.

In the first scenario, flow $f_{(3,4)}$ is an existing real-time flow in the network, which sends out packets of 1000 bytes every 4ms, i.e., at a rate of 250packets/s. Its average end-to-end delay is about 1.3ms. Now another real-time flow $f_{(0,2)}$ (with the same flow statistics as $f_{(3,4)}$) arrives, attempting to access the network. Without any explicit load control, by examining the residual air time left by flow $f_{(3,4)}$ we find that adding this new flow into the network causes it to experience an average end-to-end loss rate of 43.8% and an average end-to-end delay of 356ms, resulting very poor performance. On the other hand, $f_{(3,4)}$ is affected only slightly, with an average end-to-end delay of 1.4ms and negligible (0.0%) loss rate. The poor performance suffered by $f_{(0,2)}$ is due to two major factors: i) links $l_{(0,1)}$ and $l_{(1,2)}$ directly conflict with each other; and more importantly ii) node N_3 is a “hidden terminal” (outside the sensing range of N_1) that interferes the packet reception at N_2 , causing packet collision that leads to backoff and retransmission at N_1 . As a result, the MAC frame collision probability over the link $l_{(1,2)}$ is 51.7%.

In the second scenario, we assume that a real-time flow $f_{(0,2)}$ comes first, periodically sending packets of 50bytes at a rate of 100 packets/s. The observed average end-to-end delay is 1.6ms and zero packet loss. Then a greedy BE flow $f_{(3,4)}$ arrives, sending many packets of 1500bytes, the maximal packet size. We observe that the packet loss ratio and packet airtime of RT $f_{(0,2)}$ is determined by the transmission rate of flow BE $f_{(3,4)}$. If it transmits at 250packets/s, then the end-to-end loss ratio of flow $f_{(0,2)}$ is 0% and the average end-to-end delay is 18ms. However, if flow $f_{(3,4)}$ transmits at 330packets/s, then the average end-to-end loss ratio and delay of flow $f_{(0,2)}$ become 38.2% and 547ms. It demonstrates the effectiveness of rate control for BE traffic to the performance of RT traffic. The degraded performance is

because the packet transmission at N_3 (the hidden terminal) interferes with packet reception at N_2 .

Moreover, we have done experiments on our testbed to verify the feasibility of VoIP over multi-hop wireless networks. We observe that when the number of hops for VoIP flows is around 4 to 6, which is typical in a mesh network, around 8 simultaneously VoIP flows can be supported. The major issues we have found are the large delay and high packet losses caused by collision and buffer overflow (due to long channel busy time and collision back-off).

Our simulation and experimental results demonstrate that while it is feasible to support VoIP services over multi-hop wireless networks, it is imperative that the network load be appropriately controlled – we must keep the channel busy time and collision rate below certain thresholds to reduce the delay and packet loss rate, and thus ensure acceptable VoIP quality. In controlling the network load, we need to consider both RT and BE traffic and their interaction as well as their impact on the quality of existing VoIP flows.

3. SoftMAC Architecture

We propose a novel software framework, called Layer 2.5 SoftMAC, to support VoIP services over multi-hop wireless networks utilizing off-the-shelf commercial 802.11 a/b/g NICs and standard MAC DCF. Layer 2.5 SoftMAC lies between 802.11 MAC layer and IP layer: it leverages the (limited) coordination by 802.11 MAC DCF, and enhances it via software mechanisms for support of VoIP services. The key idea behind our SoftMAC is to employ “coarse-grained” control mechanisms (e.g., distributed admission control and rate regulation) to coordinate and regulate network load and packet transmission of both real-time (RT) and BE traffic among neighboring nodes in a distributed manner. The objective is to keep the channel busy time and collision rate below appropriate levels, and thus ensure acceptable VoIP quality.

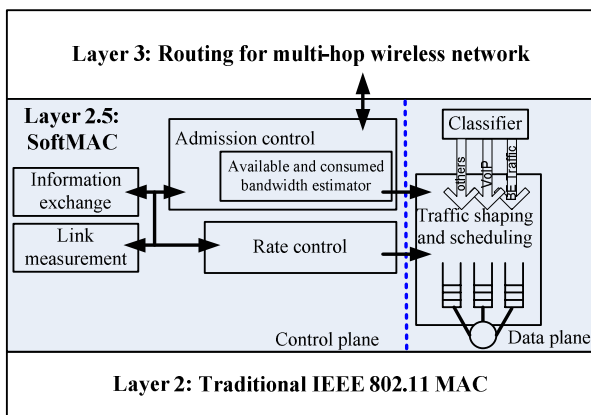


Fig.2 SoftMAC architecture and components for VoIP

Fig.2 provides a schematic depiction of our proposed Layer 2.5 SoftMAC architecture, which is divided into two planes: the control plane and data plane. There are four modules in the control plane and works as following. Each node will exchange its RT and BE traffic information explicitly to its neighbors by

broadcasting, and measure the current link status such as capacity and packet loss ratio. With all the information, we can estimate the consumed and available bandwidth for admission control, and the rate of BE traffic will be controlled by a rate control mechanism at each node. Based on the decisions of control plane modules, priority queueing and traffic shaping are used in the data plane to regulate the rate of packet transmission. The key components of SoftMAC are thus admission control (AC) and rate control (RC) in the control plane, and the priority queueing mechanism in the data plane.

The distributed AC module decides whether the available resource is sufficient to support new connections. Unlike in a wired network where the utilization of a given link is determined by the bandwidth consumption of existing flows traversing the link, in a multi-hop wireless network determining the “link utilization” is no longer as trivial. To address this problem, we introduce the novel notion of *fraction of air time* to represent the utilization of a physical channel. It maps the bandwidth requirement at the application layer to the actual air time required at a wireless link, taking header overheads, time varying link capacity and packet collision ratio into account. As a result, more accurate available bandwidth estimation is obtained.

The rate control module regulates the rate of BE packets to reduce their impact on existing VoIP flows. More specifically, a portion of the residual air time left by the real-time traffic is allocated to BE traffic in a distributed fashion. The actual air time cost at each link will vary with the currently measured link capacity and packet loss ratio, so the traffic shaper at SoftMAC needs to adjust the BE transmission rate accordingly.

In the priority queueing module, non-preemptive priority is always provided to VoIP traffic at each node, so BE packets can only be passed to the MAC layer when there is no VoIP packet waiting to be served. In a sense, we move data buffering from the 802.11 MAC layer to layer 2.5 SoftMAC in order to regulate packet transmission.

Note that the control mechanisms employed by SoftMAC are fairly coarse-grained. In particular, it does *not* perform distributed *per-packet* transmission scheduling among neighboring nodes to control when packets are transmitted by the MAC so as to avoid collision. Such fine-grained distributed per-packet scheduling requires *tight* clock synchronization among neighboring nodes, which can be hard to achieve. Instead our distributed control mechanisms only control when and how many RT and BE packets are passed to the MAC layer of a node; neighboring nodes still compete for the wireless channel using CSMA/CA to decide when individual packets are transmitted. Our coarse-grained SoftMAC provides compatibility and extensibility with future more powerful wireless MAC and PHY mechanisms. For example, if 802.11e is available, our SoftMAC can easily take advantage of its ability to differentiate and regulate transmission of packets of different priorities among neighboring nodes in the carrier sensing range, thus simplifying our priority queueing and traffic shaping functions. In general, our SoftMAC can be readily upgraded to incorporate enhanced control functions and accommodate new hardware technologies.

4. Fraction of Air Time (FAT)

The admission control and rate control in SoftMAC requires quantified metric to denote consumed (including allocated for reservation) and residual resource. In this section, we first explain the difference of transmission time cost for a packet on a wireless link and that on a wired link. Then based on such understanding, we introduce the notion fraction of air time as the metric to quantify the resource – air time on wireless link. The consumed and the residual resource are estimated taking the time varying wireless link status and interference into consideration.

First, unlike wired networks, where the concept of capacity over a (wired) link is a well-defined constant and can be easily obtained, the notion of “capacity” of a wireless link is a time varying value due to auto rate adaptive modulation selection [24, 25]. As a result, the time needed for transmitting a frame of the same size by a node to the same neighbor at different time instances, or to different neighbors, will be different, depending on the channel conditions. Second, the ambient interference and contention-based shared media access control leads to collision, which increases the air time cost on a wireless link and reduces its effective capacity. This is different from (wired) Ethernet where collision detection (CD) can significantly reduce the collision overhead; and more importantly once a frame is transmitted collision-free, its transmission time is determined solely by its frame size and the link capacity. Therefore, to estimate the air time consumed to transmit a packet over a wireless link, we must take into account the time varying channel conditions (e.g., capacity as reflected by frame loss rate) of the link as well as the traffic load at interfering links.

We introduce the novel concept of *fraction of air time (FAT)* – in two related forms, *consumed* and *residual* FAT – to represent the normalized utilization or available capacity resource of a wireless link in multi-hop wireless network. Based on the notion of (packet) air time (cost) introduced earlier – the time required to successfully transmit a packet over a wireless link that includes the actual packet transmission cost as well as the MAC ACK time cost, retransmissions cost, etc. – the (*consumed/residual*) fraction of air time is defined as the ratio of the total air time *consumed/available* in a given time interval to the length of the interval. The length of the time interval should be sufficiently large relative to the air time cost of a packet (of maximal size). In this paper we choose time intervals of 1 second.

We first illustrate the concept of the consumed FAT and use an example to how to estimate it. Consider a VoIP flow from node i to node j over link $l_{(i,j)}$. For simplicity, we assume that the VoIP flow is constant bit rate with fixed packet size, PL (in bits), and packet inter-arrival time, t_{int} (in seconds). The *consumed* FAT (of the VoIP flow) at link $l_{(i,j)}$ is simply,

$$r_{(i,j),RT} = t_{(i,j),PL} / t_{int} \quad (1)$$

where $t_{(i,j),PL}$ denotes the air time cost for a packet with size PL at link $l_{(i,j)}$. We now discuss how to estimate $t_{(i,j),PL}$. Assume that the current packet loss probability of link $l_{(i,j)}$ is

$p_{(i,j)}$. We can estimate the air time cost of transmitting a packet of size PL over link $l_{(i,j)}$ with a packet loss rate $p_{(i,j)}$ as follows:

$$t_{(i,j),PL} = \sum_{k=1}^m (p_{(i,j)})^{k-1} (1 - p_{(i,j)}) (T_s + \sum_{l=1}^{k-1} T_c) + (p_{(i,j)})^m m T_c \quad (2)$$

where m is the maximal number of (re)transmissions, T_s , and T_c are the air time cost of a successful and failed transmission of a frame of payload size PL , respectively. The value of m defined in 802.11 is 4 for the basic access method, and 7 for the RTC/CTS access method. Estimation of T_s and T_c requires knowledge of physical link parameters, e.g., the overhead introduced by the backoff, the frame header size, and the link rate $C_{(i,j)}$ of link $l_{(i,j)}$. For example, the air time cost T_s for the basic access method is given by [27, 28, 29],

$$T_s = t_{backoff} + t_{headers} + t_{ACK} + PL / C_{(i,j)} + t_{SIFS} \quad (3)$$

Note that using (2)(3) as an approximation to the air time cost $t_{(i,j),PL}$ may potentially overestimate the actual cost, since the backoff time is actually “shared” by all nodes in contention. To compensate for this, we could refine the estimation by dividing the backoff time by the number of neighboring nodes. Note that although the cost of collisions may also be shared among all packet transmissions in collision, this effect is partially absorbed by the different frame loss probability experienced by different nodes (over the conflicting links). For simplicity, in this paper we will simply use (2) for packet air time cost estimation. In summary, we see that the notion of consumed FAT not only takes into account the actual flow rate and protocol overheads (headers, ACK, etc), but also the time varying access overheads (collision, link loss rate) and the link capacity.

We now illustrate how to estimate the residual FAT (rFAT) of a wireless link, which intuitively measures the available FAT of a wireless link that is not “used” by existing interfering RT traffic and thus can be used by new RT flows. Consider a wireless link $l_{(i,j)}$ and let $r_{(i,j),RT}$ be its consumed FAT. On the first thought, one would think that the residual FAT is simply $1 - r_{(i,j),RT}$, as it would be in a wired link. This is in fact not correct, as in a multi-hop wireless network, how much capacity is available on $l_{(i,j)}$ for admitting new flows is not only determined by the existing flow(s) on link $l_{(i,j)}$, but also by the existing flows on other links or nodes that may be in conflict (i.e., interfere) with transmission on link $l_{(i,j)}$, i.e., interfering traffic at neighboring links or nodes. Different from the case in synchronized network where two traffic with only their receivers in the interference range of each other aren’t in conflict, in asynchronous 802.11 networks, the required reverse ACK enlarges the interference range of a link and makes the above case in conflict. Hence to estimate the residual FAT of $l_{(i,j)}$, we need to estimate the total load at the neighbors of node i or node j that may potentially be in conflict with $l_{(i,j)}$. For this purpose, we introduce two auxiliary notions: for node k , we define the nominal residual FAT of node k , $nrFAT_{k,RT}$ as follows:

$$nrFAT_{k,RT} = \max \left\{ 0, 1 - \sum_{m \in N(k) \text{ or } m \in N(k)} r_{m,RT} \right\} \quad (4)$$

where $N(k)$ denote the set of neighbors of node k , which includes nodes that are in the communication range of node k . Since 802.11 DCF requires reverse MAC ACK to confirm successful unicast data transmission, the neighborhood relationship is thus symmetric. Note the estimation in (4) can be further refined by taking the traffic at the nodes in the interference range of node k , i.e., superset of its neighbor, into account.

Intuitively, $nrFAT_{k,RT}$ represents the (fraction of) potentially available capacity that are not used by the existing interfering (RT) flows at the neighbors of node k and thus is potentially available for node k for admitting new (RT) flows. However, to determine whether a new (RT) flow can be accommodated at one node, we must also take into account the impact of air time consumption at this node on its neighboring nodes, as its transmission interferes and reduces the residual FAT on its entire neighbors. Hence, in order to avoid admitting too much new load at node k that may adversely impact its neighbors, we define the residual FAT of node k , $rFAT_{k,RT}$, to be the minimum among node k and its neighbors:

$$rFAT_{k,RT} = \min(nrFAT_{k,RT}, nrFAT_{h,RT}), \forall h \in N(k) \quad (5)$$

Given the above two notions, we can now define the *residual FAT* of link $l_{(i,j)}$ to be

$$rFAT_{(i,j),RT} = \min\{rFAT_{i,RT}, rFAT_{j,RT}\} \quad (6)$$

which provides an approximate estimate (upper bound) of “available” FAT on link $l_{(i,j)}$ for admitting a new flow while without adversely affecting existing flows at nodes i, j and their neighbors.

In the following we use the example in Fig.3 to illustrate how to estimate the rFAT for link $l_{(1,2)}$. Fig. 3 represents a matrix consisting of 12 nodes, and the consumed FAT at each link $(r_{(i,j),RT})$ is shown in the figure. For ease of exposition, we assume that the interference range is the same as the transmission range.

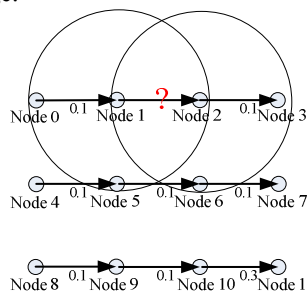


Fig.3 Example for available bandwidth (FAT) estimation

The first step is to estimate the $nrFAT$ for node 1 and node 2. Transmission on neighbor nodes may conflict with the traffic on the current node. For example, when node 2 is receiving data from node 1, transmission by node 6, e.g., an ACK to confirm the received data frame from node 10, results in collision at node 2. Using (4) to exclude all the interfering traffic, we have

$$nrFAT_{1,RT} = 1 - r_{(0,1),RT} - r_{(2,3),RT} - r_{(4,5),RT} - r_{(5,6),RT} = 0.6$$

and similarly we have $nrFAT_{2,RT} = 0.6$.

The next step is to estimate rFAT for node 1 and node 2. To ensure the network is not overwhelmed, the $nrFAT$ at each node should be kept as non negative. For example, we have $nrFAT_{1,RT} = 0.6$ and $nrFAT_{5,RT} = 0.4$, if node 1 really takes 0.6, then node 5 is overwhelmed, so actually the residual FAT that node 1 can further use is,

$$rFAT_{1,RT} = \min(nrFAT_{1,RT}, nrFAT_{0,RT}, nrFAT_{2,RT}, nrFAT_{5,RT}) = 0.4$$

Similarly, we have $rFAT_{2,RT} = 0.2$.

Finally, considering that if node 1 just uses its $rFAT$ at 0.4, then the $nrFAT$ for nodes that are node 1 and 2’s neighbor will reduce 0.4, which leads to the suffering of node 6 since its $nrFAT$ is only 0.2. So the rFAT for link $l_{(1,2)}$ cannot be greater than the $rFAT$ at both node 1 and node 2, i.e.,

$$rFAT_{(1,2),RT} = \min(rFAT_{1,RT}, rFAT_{2,RT}) = 0.2.$$

From the above discussion and example, we see that the notion of *residual fraction of air time* (rFAT) of a wireless link provides us with an approximate estimate of the available resource at a wireless link that can be used to admit new RT flows. It not only takes into account the “reduced” capacity on the wireless link due to existing interfering RT traffic in its neighborhood, but also attempts to minimize potential impact of new flows on existing RT traffic in the neighborhood. The estimation formula (6) for rFAT of a wireless link is rather a *conservative* one, as existing traffic may interfere with multiple neighboring nodes, and thus its impact may be counted in multiple neighboring nodes in (4). Although it may be possible to derive a more accurate estimate of rFAT, such estimation may require more accurate estimation of traffic load at neighboring nodes or even nodes in interference range, and their relationship such as the overlapped time cost, incurring more communication and computational overheads. In addition, it may not be amenable to distributed computation of rFAT. For the simplicity, we decide to use (6) as an approximate estimate for rFAT of a wireless link, as our goal of this paper is primarily to demonstrate the feasibility of supporting VoIP using our proposed SoftMAC architecture. We plan to study the problem of more accurate estimation of rFAT in future work. Note also that in addition to RT traffic, other traffic in high priority, such as routing, SoftMAC signaling and capacity measurement, can also be taken into account to make more accurate estimation for rFAT. In the next section, we will describe how to apply the notion of residual FAT in 1) *distributed* admission control (AC) in to decide whether there is sufficient resource to admit new RT flows, which in advance takes the FAT consumption in multi-hop traverse into account; and in 2) rate control (RC) to ensure that BE traffic will only use residual capacity left by RT traffic.

5. Key Modules in SoftMAC

In this section we describe the control mechanisms and algorithms used in the three key modules in our proposed SoftMAC architecture: 1) distributed AC for regulating the real-time (VoIP) traffic load, 2) RC for minimizing the impact of BE traffic on RT traffic, and 3) priority queueing for VoIP and BE traffic.

5.1 Admission Control for VoIP traffic

To perform admission control to regulate the VoIP traffic load, each node needs to estimate the rFAT of its incident wireless links (shared channels). Moreover, when a new VoIP flow request arrives at the node, the required (i.e., consumed) FAT for the flow on the outgoing link is estimated using the flow request information (e.g., flow rate, packet size) and the current channel status. Based on the estimate, the AC module checks whether the residual FAT is sufficient to support the flow. If the decision is affirmative, then the new reservation information will be broadcasted to make the reservation take effect. In our design, the reservation message is piggybacked with the routing discovery or query messages for fast pruning. The reservation is kept in “soft state” and is removed after a timeout or by an explicit release message.

5.1.1 Method for admission control

For admission control, the key step is to determine whether the resource is sufficient to support the new VoIP flow, i.e., whether the residual FAT is larger than the consumed FAT. In order to perform AC correctly, when estimating the consumed FAT of the new flow on a link, we must also take into account the impact (i.e., reduced FAT) on the adjacent links along its path that it directly interferes. We illustrate this point through the sample example in Figure 4. Suppose node 2 needs to check whether the available bandwidth is sufficient to support a new VoIP flow from the source node 0 to the destination node 5. Its consumed FAT on link $l_{(0,1)}$, $l_{(1,2)}$ and $l_{(3,4)}$ will also reduce the residual FAT of node 2. This is because a VoIP flow consists of a stream of packets that must be transmitted continually along its path; in other words, packets of the same flow will compete for the resource simultaneously. Hence when node 2 estimates its *total consumed FAT* of the new flow on $l_{(2,3)}$, it should take into account the interference of packets transmission from the same flow on the three adjacent links, $l_{(0,1)}$, $l_{(1,2)}$ and $l_{(3,4)}$.



Fig.4 Example for multi-hop bandwidth consumption

Formally, let f denote the VoIP flow that we will apply AC, and $l_{(i,j)} \in f$ denote that flow f will traverse on link $l_{(i,j)}$. Let $CFAT_{(i,j),f}$ denote the consumed FAT of flow f on link $l_{(i,j)}$, estimated using eqs. (1) and (2). Therefore, the *total consumed FAT in advance* of flow f on link $l_{(i,j)}$, denoted by $TCFAT_{(i,j),f}$, is given by

$$TCFAT_{(i,j),f} = \sum_{m \in N(i) \text{ or } n \in N(i)} CFAT_{(m,n),f}, l_{(i,j)} \in f, l_{(m,n)} \in f \quad (7)$$

So node i checks whether $TCFAT_{(i,j),f} \leq rFAT_{(i,j),RT}$ is satisfied, where the link rFAT is derived from (6). The details regarding the AC implementation, such as how to piggyback the reservation request in routing, are described in Section 6.

5.1.2 Method to release unused resource (FAT)

The reserved resource for a VoIP flow must be released under either of the following two cases: 1) when AC fails at some node, the nodes that have already reserved the resource for the VoIP flow should release it; 2) when the VoIP flow terminates or when the route for the flow changes, the reserved resource along the original route should also be released. Clearly relying only on the *explicit release* signaling message, it cannot guarantee that the nodes with the reserved resource will always release it correctly. For this reason, we also employ a time-out based *implicit* resource release mechanism. If a node observes that there is no packet arriving from a reserved VoIP flow after a T_{res-TO} value, it will remove the reservation and release the corresponding resource. Hence, the reservation is in *soft state*: it will be cleared after a timeout.

5.2 Rate Control for BE traffic

The function of rate control is to allocate the residual FAT left by the existing RT traffic (as well as newly admitted RT traffic) to BE traffic in a distributed way. Each node distributes its residual air time to links in its interference range according to their BE *weights* – the selection of BE weight for each link will be discussed later. Each node announces the bandwidth per unit weight, and each link can only transmit BE traffic according to the minimal value received.

Formally, we assign a BE weight, denoted by $w_{(i,j),BE}$, to each link, which satisfies

$$nrFAT_{i,RT} \geq \delta_i \sum_{m \in N(i) \text{ or } n \in N(i)} w_{(m,n),BE}$$

where δ_i denotes the normalized bandwidth per unit weight.

Therefore, we have

$$\delta_{i,max} = nrFAT_{i,RT} / \sum_{m \in N(i) \text{ or } n \in N(i)} w_{(m,n),BE} \quad (8)$$

Each node i will broadcast $\delta_{i,max}$ to control the BE traffic to minimize interference with the RT traffic. The consumed FAT for BE traffic at link $l_{(i,j)}$ is then controlled by using the minimal received value $\delta_{i,max}$,

$$r_{(i,j),BE} = \min(\delta_{i,max}, \delta_{j,max}) \times w_{(i,j),BE}, \forall k \in N(i), l \in N(j) \quad (9)$$

In other words, the capacity resource used for BE traffic will be limited by $r_{(i,j),BE}$. However, each node still needs to estimate the actual sending rate for BE packet using formulas similar to eqs. (1) and (2), where the average BE packet size is used. Now we discuss how to choose the BE weight for each link. In this work, we only consider the long time scale BE traffic and assume all the BE flows are greedy, so we use the number of BE flows at each link as its BE weight. Another possible choice for BE weight is the averaged queue length. However, we do not consider this choice in this work. We will study this and other choices for BE weight in the future work.

5.3 Priority Queueing for Service Differentiation

Priority queueing is used to provide VoIP traffic with high priority and BE traffic with low priority. The signaling traffic for resource reservation, link status measurement and other

information update as well as the routing traffic is given the highest priority. Hence we use a three-queue priority scheduling at each node for packet scheduling and buffer management. Note here we use priority scheduling for its simplicity in implementation. If needed, other scheduling algorithms can also be used in SoftMAC architecture.

6. System Implementation

In this section we describe the implementation of SoftMAC, which is built as a NDIS driver on Windows OS. We have implemented the three key modules according to the architecture shown in Section 3. To provide priority queueing at SoftMAC, the buffer at layer two (MAC layer) is removed. We measure the link status, including the current link capacity $C_{(i,j)}$ and loss rate $p_{(i,j)}$ to estimate the residual and consumed FAT of each link for admission control and rate control. In addition, we also incorporate our admission control functionality in the DSR routing protocol for fast pruning. Each node broadcasts its traffic information to neighbors for information exchange. We describe these procedures one by one below.

6.1 Priority Queueing and Traffic Shaping

We implement priority queueing for VoIP support. Currently three types of queues are supported. The TOS field in the IP header is used to indicate the service level of the flow a packet belongs to. The implementation uses a strict priority based scheduler as discussed in Section 5.3. To implement priority queueing at SoftMAC, we remove buffering at the MAC layer to obtain local per-packet level priority control, i.e., buffering is done only at SoftMAC. Hence the protocol driver will pass a packet down the MAC layer only if the previous packet is not in the pending status. In other words, packets will be passed down to the MAC layer one by one, and there are never multiple packets in the NIC's buffer. Clearly this will introduce some inefficiency in the NIC's performance in terms of its throughput. Fig.5 shows the test results of TCP/UDP throughput comparison of our buffering scheme and the original NIC buffering scheme over an IEEE 802.11a wireless link. From the figure, we see that the overhead is less than 14 percents. With this performance penalty, our SoftMAC is able to control the priorities of packet transmission.

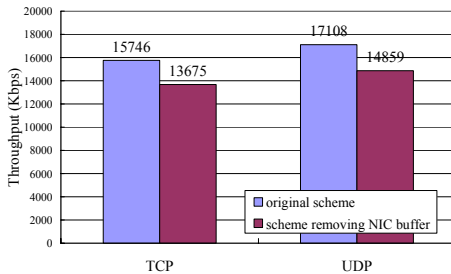


Fig.5 Performance degradation introduced by SoftMAC

We also use a token bucket to control the BE traffic rate according to eq. (9) on each link.

6.2 Measurement of Frame Loss Probability (FLP)

In our implementation, each node sends out broadcast packets to exchange link status, traffic load and other information, we measure the broadcast packet reception ratio to deduce the FLP $p_{(i,j)}$ used in eq. (2) for each link $l_{(i,j)}$. Although the MAC layer modulation of broadcast frames is different from that of unicast frames, we expect the FLP measured by broadcast frames to be fairly close to that of unicast frames, as is supported by the previous work [7,8]. This is because the frame loss due to collision with hidden terminals contributes to the major part of frame losses in a multi-hop wireless network. Furthermore, since the Network Allocation Vector (NAV) at the nodes which are in the communication range of a source node are set by the data packet transmission, MAC ACK frames experience much lower collision probability than that of data frames. Hence we can ignore the loss probability of ACK frames and use the broadcast packet loss ratio seen by a receiving node as the FLR on the link from the source of broadcast to the said receiving node. The metric is *directional*, i.e., the metric from node x to y may be different from node y to node x . This is indeed observed in our experiments using our testbed. In [7,8], the authors measure and multiply the ratio on both the forward and the reverse directions, which results in a symmetric metric. The test results in [7] also show that the ACK delivery ratio is under-estimated, resulting in overestimation for FLR. In our scheme broadcast packets are sent out every half a second, and we measure the frame loss ratio using a 5-second sliding window.

6.3 Measurement of Physical Link Capacity

We measure the actual link capacity by probing when the NIC is working at auto rate mode. For wired networks, schemes such a packet-pair [31] (two back-to-back probe packets) have been used to measure the capacity of a link. For CSMA/CA based DCF, packet-pair based scheme is likely to underestimate the channel capacity, especially when the physical bandwidth is large [8]. This is due the overhead introduced in MAC and PHY layer headers, backoff, and so forth, which cannot be ignored. Assuming the minimal overhead introduced by the MAC and PHY layers is stable, we use the difference of the two minimal valued observed in samples with different packet size to remove the impact of overhead. More specifically, we measure the physical bandwidth by probing as following: the sender sends out three probe packets back-to-back, and the packet sizes of the probes are small (PL_s), small (PL_s) and large (PL_l), respectively. The interval between the first and second probes is denoted as T_{int-ss} and the second and the third probes as T_{int-sl} . Then the bandwidth $C_{(i,j)}$ for link $l_{(i,j)}$ can be estimated as

$$C_{(i,j)} = (PL_l - PL_s) / (\min(T_{int-sl}) - \min(T_{int-ss})) \quad (10)$$

where the $\min()$ function denotes the minimal value from the most recent 10 consecutive samples obtained in last 5 seconds. The advantage of this method is that it reduces the impact of non-deterministic overhead introduced in transmission.

In our implementation, each node will pick a neighbor randomly and send probe packets to measure the bandwidth every second. Fig.6 shows the link capacity measurement results in our testbed where we manually configure two nodes with LinkSys dual band A+G NIC working on 802.11a in the ad hoc mode and fix the link transmission rate to a specific value with auto rate disabled. Since the physical rate follows the discrete set defined in the standard, we observe that the measured result is accurate enough for the current capacity estimation.

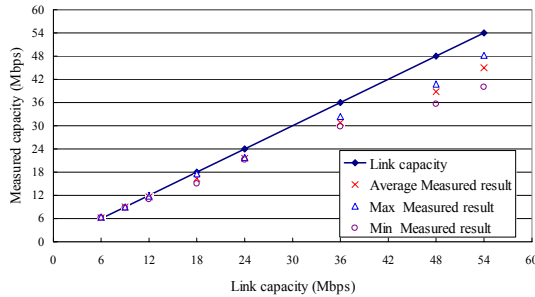


Fig.6 Physical link capacity measurement in SoftMAC

6.4 Integration of Admission Control into DSR

To speed up the AC and flow resource reservation, we use a hop-by-hop AC combined with the Dynamic Source Routing (DSR) routing protocol. In DSR, a node broadcasts a route-request to find a valid route to a destination, and the destination replies with a route-reply to notify the route to the source. We add the reservation information of a new flow (i.e., the consumed FAT of the flow estimated using eqs. (1)(2) to the DSR routing header of route-requests and replies to perform AC. Upon receiving a route request, the AC module at a node checks whether sufficient resource is available *only for fast pruning*, e.g., to eliminate routes which do not have sufficient resource for admitting the new flow. In other words, at this stage, *no reservation is made*. Only upon receiving a route-reply will the reservation for the new flow be made along the route specified by the route reply, as explained in more detail below.

When an intermediate node along a potential route to the destination receives the route-request, the AC module makes the admission control decision using the partial routing information collected so far in the DSR header. It needs to check whether the destination is its neighbor or neighbor's neighbor to estimate the TCFAT (eq. (7)) of the new flow. Such information can be obtained by overhearing the periodical broadcast from neighbors. If available bandwidth for all the potential outgoing links is not sufficient, the node will drop the route request silently and do not continue the rebroadcast of the route request.

Upon receiving a route-reply, the route information in the DSR header is used to decide whether the available bandwidth for the link is sufficient. If there is sufficient bandwidth, then the node sends the route-reply to the next upstream node, reserves the resource for the new flow and broadcasts the reservation information to other nodes. If there is not sufficient

bandwidth, the route reply is discarded, and a route reply failure message is generated and sent back to the destination node. This way the destination can send a route-reply on another route, if it exists.

6.5 Broadcast for Information Exchange

As we have described earlier, each node needs to estimate the rFAT for each outgoing link to perform distributed AC and RC. To do so, each node must tell its neighbors how much resource has been allocated and how much is left to other nodes. Following information is broadcasted by each node i broadcasts: $nrFAT_{i,RT}$, $rFAT_{i,RT}$, $\delta_{i,max}$, $r_{(i,j),RT}$, $r_{(j,i),RT}$, $w_{(i,j),BE}$ and $w_{(j,i),BE}$ where $j \in N(i)$. Two modes of broadcast are used: 1) periodic broadcast -- each node uses a timer and broadcasts its information every $T_{broadcast}$ (0.5 second); 2) trigger based update -- every time a node accepts and reserves the consumed FAT for a new VoIP flow, or releases the consumed FAT allocated to an existing VoIP flow, it broadcasts its updated information immediately.

7. Simulation and Experimental Results

In this section, we present the results of simulation and experimental evaluation of the proposed SoftMAC architecture. For simulation evaluation, we have implemented SoftMAC using the ns-2[33] simulator and its wireless extension developed at CMU. For experimental testing, we have implemented SoftMAC as a NDIS driver for Windows OS, and built a small indoor testbed. The codec used for the VoIP application is GSM 06.10, which has a data rate of 50 frames/s, i.e., one frame every 20 ms. The payload per frame generated by the codec is 33 bytes. Including the 12 bytes application header, 8 bytes UDP header, and 20 bytes IP header, the total payload per frame is 73bytes.

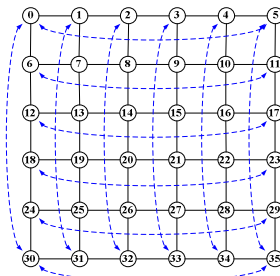


Fig.7 Grid topology and traffic pattern

7.1 Simulation Results

In our simulation evaluation, we construct a mesh network on a grid with 36 nodes (Fig.7), where the solid lines represent the wireless links between nodes, while the dashed line represents the potential traffic flows, which are always generated from the nodes on the borders along one of the six rows or columns. The distance between neighboring nodes, communication range, and CS range are set as 24m, 25m and 30m respectively. The channel model is TwoRayGround. The network parameters are all set for 802.11a DCF with 24Mbps physical rate and RTS/CTS disabled. We use two CBR flows

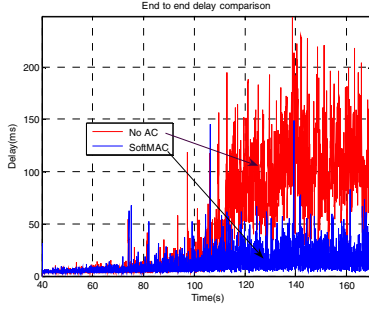


Fig. 8 End-to-end delay comparison

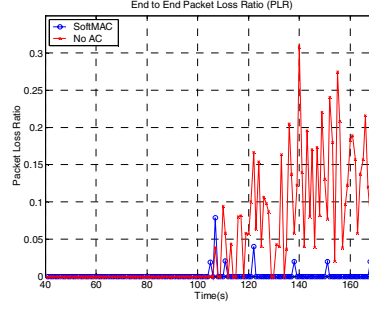


Fig. 9 Packet loss ratio comparison

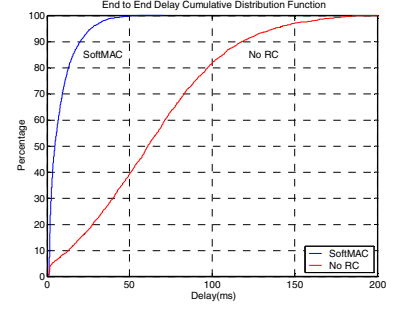


Fig. 10 End-to-end delay of VoIP

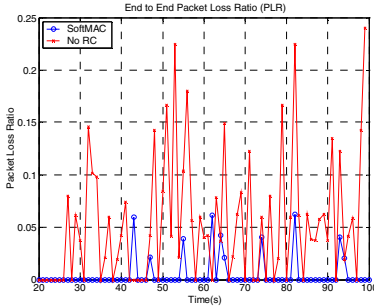


Fig. 11 End-to-end packet loss ratio of VoIP

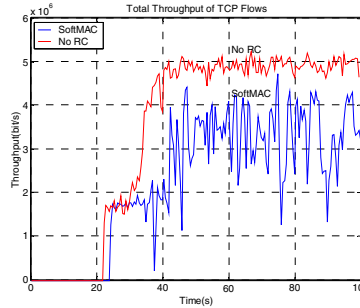


Fig. 12 Total throughput of TCP

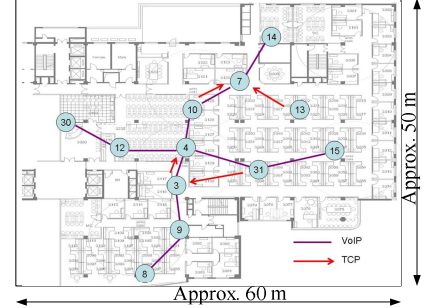


Fig. 13 Experiment topology

to emulate the traffic of a VoIP application using the parameters of the aforementioned VoIP codec.

7.1.1 Admission control for VoIP

Totally 24 VoIP flows are generated and injected into the network one by one every 4 seconds starting from second 24 (in simulation time), in the following the order: first 12 flows starting from the top row to the bottom, then the leftmost column to the right column, i.e., $f(0,5)$, $f(6,11)$,... $f(30,35)$, $f(0,30)$, $f(1,31)$,... $f(5,35)$ (here $f(i,j)$ denote the VoIP flow between node i and node j), and then another 12 flows in the same order. All flows last until the end of simulation. We continue the simulation for 50 seconds after the last flow is injected at second 116. Ideally, to measure the quality of VoIP applications, one should use metrics such as MOS (mean opinion score) [38]. For simplicity, however, we resort to the practical guideline for VoIP network performance given in [1], where it suggests that for satisfactory VoIP quality, the end-to-end one-way delay be kept below 150 ms – excluding the delay budget for codec, packetization, etc., the delay budget for the one-way network delay is about 80 ms – and the packet loss rate be kept below 10%, even with error concealment algorithms [35,36]. Using these performance metrics, we first perform the simulation without SoftMAC AC and then the simulation with SoftMAC AC. We find that without AC, after 23 VoIP flows are injected, the performance of the all VoIP flows is acceptable. However, when the 24th VoIP flows are injected, the performance of four VoIP flows degrades significantly, resulting unacceptable VoIP quality. On the other hand, with SoftMAC AC, only 22 VoIP flows are accepted. So the decision of AC is fairly accurate, one less than what is maximally admitted – this is likely due to the conservative

estimation of rFAT we used. Figs. 8 and 9 show the delay and loss performance comparison using one affected VoIP flow (between node 18 and 23 starting from time 36s) as example, where the x-axis shows the simulated time. Without SoftMAC AC, we see that after the 24th flow is injected (at time 116s), both the delay and loss rate increase significantly, frequently exceeding 80 ms and 10% loss rate, respectively, while before the injection of the 24th flow, the performance is still acceptable. With SoftMAC AC, both the delay and loss performance are kept at the acceptable level, as there are a total 22 flows in the network after time 108.

7.1.2 Rate control for BE traffic

To evaluate the efficacy of rate control (as well as priority queuing) for BE traffic, we first place 3 VoIP flows, $f(0,5)$, $f(12,17)$ and $f(24,29)$, in the network. Then we inject 12 BE flows – ftp downloads using TCP SACK – every 2 seconds into the network, one per row and one per column of the grid. We perform simulations *with* and *without* SoftMAC RC.

Fig. 10 shows the cumulative delay distributions for VoIP flows with and without SoftMAC RC. We see that with SoftMAC RC, 100% of one-way packet delays are within 80 ms, while without SoftMAC, more than 30% of packets have one-way delay of above 80 ms. Fig. 11 shows the loss performance of VoIP flows over time. We see that without SoftMAC RC, VoIP flows periodically experience losses above 10%, while with SoftMAC RC, the loss rates for the VoIP flows are always kept below 10%. Lastly we compare of the total throughput of TCP for the BE flows, which is shown in Fig. 12. We see that SoftMAC RC is able to keep a cap on the throughput on the BE traffic to minimize their impact on the VoIP flows. The oscillation in the total TCP throughput

under SoftMAC RC is caused by the priority queueing mechanism.

7.2 Experimental Results on Testbed

We have built a 32-node wireless testbed, which is located on one floor of a fairly typical office building. All the nodes are placed in fixed locations and are not moved during testing. Each node is a DELL PC, and equipped with one LinkSys Dual-band A+G card with 802.11 a/b/g support. All the nodes operate in IEEE 802.11a mode. The autorate selection on the cards is enabled and RTS/CTS is disabled. Our implementation uses the Mesh Connectivity Layer (MCL) [37] and its driver is installed on each PC in our testbed.

Firstly, we use a simple 5-hop chain topology (path 8-9-3-4-10-7 only in Fig.13) to verify the accuracy of AC in real implementation. All the VoIP flows are generated between node 7 and node 8 along the path. With AC disabled, we collect the results for 7~10 VoIP flows concurrently running (Fig.14). We observe that the system can serve 8 simultaneous VoIP flows with acceptable quality; while with more than 8 VoIP flows in the network, the delay of all VoIP flows increases dramatically. With AC enabled, only 7 VoIP flows are accepted, and the performance is similar to that with only 7 VoIP flows in the AC-disabled case in Fig.14. This demonstrates that our AC works quite accurately on a real testbed.

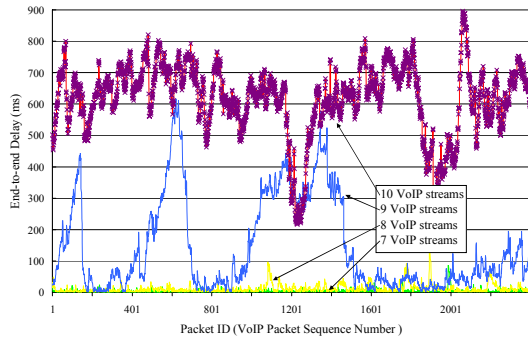


Fig.14 End-to-end delay comparison for AC

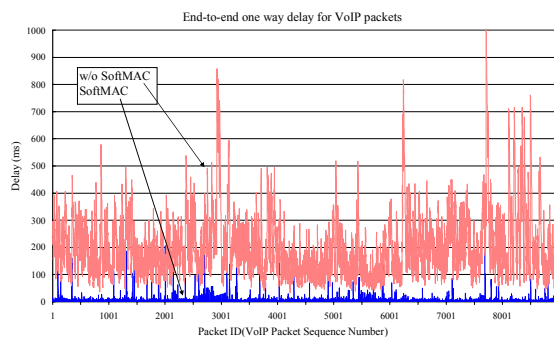


Fig.15 End-to-end delay comparison for RC

Secondly, to evaluate the efficacy of RC in SoftMAC, we collect experimental results on 11 nodes of our testbed as shown in Fig. 13. Two VoIP flows are set up on path 30-12-4-31-15 and path 9-3-4-10-7-14. Four TCP flows, $f(3,4)$, $f(10,7)$,

$f(31,3)$ and $f(13,7)$, are added as best-effort traffic. The end-to-end one-way delay of VoIP packets of the VoIP flow $f(9,14)$ are measured under two scenarios – with SoftMAC driver enabled and disabled. Fig. 15 shows the comparison of the two scenarios. The results demonstrate that SoftMAC significantly reduces the end-to-end one-way delay of VoIP packets by controlling the BE TCP traffic rate and providing priority queueing. We also observe that the voice quality is very smooth with SoftMAC, while it is bursty without SoftMAC. In addition, with SoftMAC, the TCP throughput is decreased by about 30~50% so as to provide better quality for VoIP.

8. Related Work

Although we are not aware of any other work that provides an integrated solution to support VoIP applications over a multi-hop wireless network, various studies have addressed some relevant pieces of the problem: service differentiation, AC, bandwidth sharing for BE, overlay MAC, etc.

Most of the prior works to support service differentiation and quality-of-service for VoIP over wireless networks focus on WLAN [2,3], e.g., IEEE 802.11e [5] and the proposal in [10]. As explained earlier, such WLAN based MAC and corresponding QoS enhancement schemes do not work well for multi-hop wireless networks. For ad hoc networks, there are various scheduling mechanisms (see, e.g., [11-13]), all of which either require modification of 802.11 MAC DCF, or simply assume the availability of 802.11e-like MAC [5].

In terms of distributed AC, Yang et al. [15] propose a contention-aware admission control protocol (CACP) for mobile ad hoc network by querying all nodes within the carrier sensing range to determine whether a new flow can be supported. I.D.Chakeres et al. [16] show that CACP reserves bandwidth unnecessarily, and instead propose the perceptive admission control (PAC) to estimate the available bandwidth by adjusting the carrier sensing range to measure the channel busy time. Unfortunately, such capability is not supported by the current hardware and thus requires modification of MAC. Several schemes [17~19] have been proposed for estimating available bandwidth in a wireless ad hoc network. Q.Xue et al. [20] combine the AC and bandwidth reservation in routing called AQOR. However, the available bandwidth is overestimated in previous work since the capacity estimation is performed from per node rather than per link point of view. Another problem of previous work is that the wireless link dynamics, such as time varying link capacity due to the autorate setting on NIC [24,25], frame loss(collision) probability, etc., are not considered, which motivates the FAT metric proposed in this paper.

The problem of fair bandwidth sharing in wireless networks are addressed in [21,22]. In SWAN [23], the authors use an AIMD algorithm for local BE traffic rate control, and a probing-based AC scheme at sender for real time traffic. However, probing for available bandwidth is not accurate and using AIMD with decision based only on self MAC delay can not guarantee the impact to the real time traffic on other nodes. Paper [15] shows that the average per-hop delay by reservation based CACP is much lower that of SWAN.

A.Rao and I.Stoica [32] propose a software-based overlay MAC framework to improve the performance of DCF in multi-hop wireless networks. While it is similar to SoftMAC as a software overlay between IP and MAC, their focus is on improving the throughput of BE traffic. In addition, their solution assumes local clock synchronization and employs a slotted allocation scheme that operates at a time scale of, e.g., 20 ms (slot size). The performance achieved thus hinges on the accuracy of synchronization. Furthermore, the overhead introduced by software implementation constrains the slot size, which makes such granularly not well-suited for VoIP applications with stringent delay requirement.

Therefore, all the related works and even their simple combination can not provide real time VoIP support over multi-hop wireless network with commercialized NIC, which motivates the system work of SoftMAC.

9. Conclusions

In this paper, we proposed and presented a novel software solution, called Layer 2.5 SoftMAC, to effectively support VoIP applications in multi-hop wireless networks using commercial IEEE 802.11 MAC DCF hardware. Our proposed SoftMAC resides between the 802.11 MAC layer and IP layer to coordinate – in a distributed fashion and without resorting to tight clock synchronization – the real-time and best-effort packet transmission among neighboring nodes in a multi-hop wireless network. The proposed SoftMAC architecture employs three key mechanisms: 1) distributed admission control for regulating the load of real time-traffic, 2) rate control for minimizing the impact of best-effort traffic on real-time traffic, and 3) non-preemptive priority queueing for providing high priority service to VoIP traffic. We have implemented our proposed SoftMAC architecture as a NDIS driver over NIC driver, and built a multi-hop wireless network testbed with 32 wireless nodes equipped with 802.11 a/b/g combo cards. Through extensive simulations using the network simulator NS2 and experimental testing on the testbed, we have demonstrated the efficacy of our proposed software solution. We plan to expand our testbed and perform more extensive testing. In addition, we plan to further explore several design issues (e.g., the BE weight, mobility, etc) and improve the performance of SoftMAC.

References

- [1] Bur Goode. Voice over Internet Protocol (VoIP), *Proceedings of the IEEE*, Vol.90, No.9, Spt. 2002
- [2] S.Garg and M.Kappes. On the throughput of 802.11b Networks for VoIP, *Avaya TR. ALR-2002-012*, March 2002
- [3] M.Veeraraghavan, N.Cocker, and T.Moors. Support of voice services in IEEE 802.11 wireless LAN, *INFOCOM '01*, April 2001
- [4] IEEE standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, ISO/IEC 8802-11:1999(E)
- [5] B.P. Crow and J.G. Kim. IEEE 802.11 Wireless Local Area Networks, *IEEE Comm.*, Sept. 1997
- [6] IEEE draft for Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: MAC Enhancements for Quality of Service (QoS), IEEE Std 802.11e/D4.0, November 2002
- [7] D.De Couto, D.Aguayo, J.Bicket, and R.Morris. High-throughput path metric for multi-hop wireless routing, In *Mobicom*, 2003
- [8] R.Draves, J.Padhye, and B.Zill. Routing in multi-radio, multi-hop wireless mesh networks, In *Mobicom*, 2004
- [9] S.Xu and T.Saadwi. Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks. *IEEE Comm.*, Jun. 2001
- [10] Q.Pattara-akikom and P.Krishnamurthy. Distributed mechanism for quality of service in wireless LANs, *IEEE Wireless Comm.*, June 2003
- [11] V.Karodia and C.Li. Distributed priority scheduling and medium access in ad hoc networks. *Wireless Network*, 8, 455-466, 2002
- [12] Y.Yang and R.Kravets. Distributed QoS Guarantees for Realtime Traffic in Ad Hoc Networks. *UIUCDSCS-R-2004-2446*, June 2004
- [13] K.C. Wang. Quality of service assurances in multihop wireless network, Dotor Dissertation, Univ. of Wisconsin-madison, 2003
- [14] G.Allard, L.Georgiadis, P.Jacquet, and B.Mans. Bandwidth reservation in multihop wireless networks: complexity, heuristics and mechanisms, *International Journal of Wireless and Mobile Computing*, May 2004
- [15] Y.Yang and R.Kravets. Contention-aware admission control for ad hoc networks. TR. 2003-2337, UIUC, April 2003
- [16] I.D.Chakeres and E.M.Belding-Royer. PAC: perceptive admission control for mobile wireless networks. TR. UCSB-CS-2004-17, Univ. of California, May 2004
- [17] M.Li, B.Prabhakaran, and S.Sathyamurthy. On flow reservation and admission control for distributed scheduling strategies in IEEE 802.11 wireless WLAN, *MSWiM'03*, San Diego, Sept. 2003
- [18] D.H.Cansever, A.M.Michelson, and A.H.Levesque. Quality of service support in mobile ad-hoc IP networks. In *MLCOM* 1999, Oct 1999.
- [19] L.Cerdo, J.Barcelo, and J.Garcia. Reservation Scheme Satisfying Bandwidth QoS Constrains for Ad-hoc Networks, *DAC TR. UPC-DAC-2004-24*, 2004
- [20] Q.Xue and A.Ganz. Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks, *Journal of Parallel and Distributed Comput.*, 63(2003), 154-165
- [21] Z.Fang and B.Bensaou. Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks, *IEEE Infocom'04*, 2004
- [22] T.Nandagopal, T.E.Kim, X.Gao and V.Bharghavan. Achieving MAC layer fairness in wireless packet networks, in *ACM Mobicom* 2000
- [23] G.Ahn, and A.T.Campbell. A.Veres, L.Sun. Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (SWAN), in *IEEE Trans. On Mobile Comput.*, Sept. 2002
- [24] G.Holland, N.Vaidya, and P.Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks, *Mobile Computing and Networking*, 2001
- [25] B.Sagdehi, V.Kanodia, A.Sabharwal, and E.knightly. Opportunistic media access for multirate ad hoc networks, in *Mobicom* 2002
- [26] J.Li, C.Blake, D.De Couto, H.Lee, and R.Morris. Capacity of ad hoc wireless networks, In *Mobicom* 2001
- [27] H.Wu, Y.Pong, et al. Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement. In *INFOCOM* 2002
- [28] G.Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. In *IEEE JSAC.*, V18, N3, March 2000
- [29] M.M.Carvalho and J.J.Garcia-Luna-Aceves. A scalable model for channel access protocols in multihop ad hoc networks, In *Mobicom'04*
- [30] F.Alizadeh-Shabdiz and S.Subramaniam. Analytical models for single-hop and multi-hop ad hoc networks, *BROADNETS'04*, Oct. 2004
- [31] S.Keshav. A control-theoretic approach to flow control. In *SIGCOMM*, 1991
- [32] A. Rao and I. Stoica. An Overlay MAC Layer for 802.11 Networks, In *Mobisys*, 2005
- [33] "NS2", URL <http://www.mash.cs.berkeley.edu/ns/>
- [34] D.B. Johnson, D.A. Maltz, and Y. Hu. The Dynamic Source Routing protocol for mobile ad hoc networks (DSR). Internet draft, April 2003. <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt>
- [35] C.Boutremans and J. Boudec. Adaptive Joint Playout Buffer and FEC Adjustment for Internet Telephony, In *IEEE INFOCOM'03*, 2003
- [36] H. Sanneck. Concealment of lost speech packets using adaptive packetization, in *Proceedings IEEE Multimedia Systems*, June 1998.
- [37] Microsoft Mesh Connectivity Layer (MCL) Software, <http://research.microsoft.com/mesh>
- [38] Athina P. Markopoulou, Fouad A. Tobagi, Mansour J. Karam, Assessment of VoIP Quality over Internet Backbones, In *IEEE INFOCOM*, 2002