E-MAC: An Elastic MAC Layer for IEEE 802.11 Networks

Qing Wei, Imad Aad[‡], Luca Scalia, Jörg Widmer, Philipp Hofmann[*], Luis Loyola[**]
DOCOMO Euro-Labs, Nokia Research Center Lausanne[‡], IABG mbH[*], SkillupJapan Corporation[**]
lastname@docomolab-euro.com, imad.aad@nokia.com[‡], hofmannp@iabg.de[*], luchoja2006@googlemail.com[**]

Abstract

We present a system for real-time traffic support in infrastructure and ad-hoc IEEE 802.11 networks. The proposed Elastic MAC (E-MAC) protocol provides a distributed transmission schedule for stations with real-time traffic requirements, while allowing a seamless coexistence with standard IEEE 802.11 clients, protecting best-effort 802.11 traffic from starvation by means of admission control policies. Our scheduling decisions are based on an "elastic" Transmission Opportunity (TXOP) assignment which allows for efficient wireless resource usage: whenever a real-time station does not use the assigned TXOP, the other real-time stations can take over the unused access opportunity, thus preventing the well known inefficiencies of static TDMA schemes. Unlike other TDMA-based solutions for 802.11, E-MAC does not require a tight synchronization among the participating clients, thus allowing its implementation on commodity WLAN hardware via minor software changes at the client side, and no changes at the access points. We studied the performance of our mechanism via ns-2 simulations and a mathematical model, showing that it outperforms IEEE 802.11e in terms of throughput, delay, and jitter. We finally provide a proof of concept through the results obtained in a real testbed where we implemented the E-MAC protocol.

1. Introduction

In the last few years we have witnessed an incredible proliferation of wireless data networks: WiFi/WiMAX mesh networks for metropolitan access, ADSL/WiFi routers for domestic indoor networking, Bluetooth personal area networks for device integration and personal area networks (GPS, earphones, etc.), 3G-HSPA services for ubiquitous connectivity, and 4G-LTE deployments for high-speed data services. This trend also feeds the demand of the users for new value-added data services: in the coming years, real-time video traffic will exceed P2P traffic in terms of bandwidth usage, thus requiring important reworking of wireless protocols to support the required QoS (Quality of Service).

This is particularly true for the unlicensed ISM 2.4 GHz band: the lack of regulations and the presence of numerous uncoordinated devices has made this portion of spectrum difficult to control, thus making the provisioning of voice and video applications extremely challenging. Throughout the literature, several approaches to provide QoS support can be found. They can be split into two types: those with statistical guarantees and those with strict guarantees.

Statistical guarantees are usually based on contention among competing stations to access the channel, as for example in 802.11e [18]. High priority flows (or stations) are assigned high-priority channel access parameters (e.g., AIFS or contention window) to gain access to the channel more often than low-priority flows. On the one hand, these approaches are relatively easy to implement, deploy, and manage, thus boosting their success in the market. On the other hand, the guarantees they provide are statistical, which causes problems in case hard QoS guarantees are required.

Strict guarantees are based on reservations (e.g., 802.11 infrastructure PCF/HCCA - Hybrid Centralized Channel Access mode [17] or TDMA). The reservations can be tailored to the QoS requirements of different applications. However, they are rather complex to implement and manage, which so far has hindered their deployment.

Besides the drawback of implementation complexity, strict guarantees rely on "medium reservations" that are often done on a periodic basis: a given station reserves the channel for transmitting (P bytes) every T seconds. However, this periodicity may not match the real-time traffic pattern generated by the data sources. Consider for instance voice traffic, where the communication channel is typically idle 1/3 of the time. Several voice codecs, e.g., the ITU-T G.711 μ-Law codec [20], optimize bandwidth usage by applying silence suppression, leaving several reserved time-slots empty. One may try to adapt the period of the slot reservation to the voice codec pattern to optimize bandwidth efficiency or to reduce delays, but never both at a time. To reduce the average packet delay, short slot intervals must be used, but for such an over-provisioned reservation many of the reserved slots may be empty, therefore drastically reducing the efficiency, i.e., the ratio of the used slots to the total number of time slots. Conversely, increasing efficiency is likely to increase packet delays which is undesirable for real-time applications.

With these considerations in mind, we design an *elastic MAC* (E-MAC) protocol to provide strict QoS guarantees for real-time traffic, with "elastic" reservations to allow for empty-slot reuse. A possible option is to adopt the centralized scheduling proposal HCCA of the 802.11e amendment. With HCCA, after the transmission of the beacon the access point (AP) reserves the channel for a specific amount of time, during which it polls the real-time stations. The polling is performed on the basis of Traffic Specifications (TSPECS), given by the stations through radio resource requests. This mechanism prevents the AP from scheduling clients that do not have real-time packets to transmit, while allowing it to order the polling list according to the specific traffic deadlines of the users. The remaining part of the superframe – the interval between two consecutive beacons – is left for legacy channel access contention. However, due to its implementation complexity, no HCCA-based APs can be found in the market. We design a protocol that meets the requirements for strict QoS guarantees and empty-slot reuse, for both infrastructure and ad hoc mode. It is compatible with existing 802.11 devices and deployed 802.11 networks and hotspots. To show the viability and efficiency of our approach, we implement it in a real testbed.

The paper is organized as follows. In Section 2, we give an overview of related work. In Section 3 we describe the design goals and the protocol in detail. The mathematical model is presented in Section 4. The protocol is evaluated both through simulations in Section 5 and in a testbed in Section 6. Finally, we discuss various issues and future work in Section 7 and conclude the paper in Section 8.

2. Related Work

In the past decade there has been considerable research in wireless networks with particular focus on QoS [7, 15, 33]. Existing approaches cover a wide range of applications, requirements, and assumptions, however the lack of feasibility is a common drawback of many past works. With the availability of several open-source Linux-based 802.11 MAC drivers, developed thanks to the reverse-engineering work made by user communities [19, 22, 24, 27], there is a significant increase of experimental work on QoS-oriented 802.11 solutions.

One common aspect of these works was the adoption of deterministic QoS provisioning mechanisms directly at the MAC layer. In particular, a lot of research targeted TDMA-based MAC protocols able to support PCF-like contention free channel access without incurring the unpredictable delays and overhead of a centralized polling scheme [8, 10, 12, 13, 16, 25, 26, 28, 30, 31]. The rationale behind this design choice lies in the deficiency statistical prioritization mechanisms supported in the 802.11e amendment. Apart from the inherent 802.11e limitations on achieving the expected QoS guarantees, it has been shown in [5, 11] that several commercially available WLAN devices do not exhibit standard compliant MAC behavior. Lower contention window sizes, absence of backoff mechanism, incorrect AIFS timing, NAV neglection and so on have made the deployment of 802.11e QoS-oriented mechanisms infeasible in real-world scenarios, where fair bandwidth sharing between the devices is expected. Even for standard compliant network deployments, it has been shown in [9] that IEEE 802.11e QoS-oriented enhancements can provide the desired delay performance only when the number of real-time traffic sources is very low. As shown in [6], with a larger number of real-time traffic sources, the statistical prioritization offered by the 802.11e mechanism becomes insufficient to preserve the delay requirements. This has lead to the common understanding that, in order to effectively support traffic prioritization over 802.11 legacy hardware, deterministic approaches have to be followed. Most of the proposed work explicitly designed non-802.11 compliant access schemes, whereas coexistence with legacy 802.11 traffic can be achieved at the cost of introducing a PCF-like reservation mechanism, together with its related inefficiencies [32]. Furthermore, as shown in [23], the introduction of a contention free period can also have the adverse effect of excessively delaying real time traffic, thus requiring a more elastic traffic scheduling mechanism able to follow the traffic generation pattern. [34] proposes a flexible scheduling scheme by adaptively segregating the realtime traffic and non-realtime traffic. However, that approach is still based on PCF and thus has some complexity issues.

The implementation of TDMA-like schemes on top of commodity 802.11 hardware requires modification of the driver source code: low-level 802.11 functions and parameters like the exponential backoff, the physical and virtual carrier sense, the slot-time duration and IFS size need to be modified, disabled, or reconfigured. In addition, the lack of central coordination and the distributed nature of these access schemes inevitably requires the introduction of tailored synchronization mechanisms able to align the time slots of all clients to some time reference.

The author of [30, 31] have built TDMA-like protocols and scheduling policies on the basis of the SoftMAC framework [25]. Peer synchronization is achieved by means of guard times between consecutive time slots and a wired connection to a central node is used to achieve a precision of 25 μs.

In 2P [8], the authors provide a customized TDMA MAC protcol for interference mitigation in a multichannel environment. A node is in transmission mode for a specific time period that is globally known, and then explicitly notifies the end of its transmission period to each of its neighbors using marker packets. A receiving node waits for the marker packets from all its neighbors before switching over to transmission mode. In the event of a loss of a marker packet, a receiving node uses a timeout to switch into the transmission mode. The 2P protocol suffers from performance impairments in lossy environments, where marker packets can be easily lost. The same authors, in [26], adopt a looser synchronization scheme tailored for lossy environments. The approach resembles the NTP principles, that implicitly correct the offset between the beginning of the transmission and the beginning of the time slot.

The authors in [12] propose a TDMA MAC protocol able to exploit elastic TDMA transmission scheduling thanks to an out-of-band synchronization mechanism able to achieve a precision of the order of a few microseconds.

In [28] Rao and Stoica propose an overlay TDMA MAC layer on top of 802.11 hardware to overcome the typical 802.11 MAC performance impairments. They fix the slot size to 10 msec and use a leader node to generate the "clock" to synchronize all the nodes in the network. Time stamps and latency estimation are appended to each packet header to compute the clock skew at each receiver.

Finally, [16, 10] propose TDMA-based MACs explicitly tailored to achieve VoIP and video traffic improvements over the 2.4 GHz band. The TDMA scheme is similar to IEEE 802.16. A superframe structure, divided in uplink and downlink phases is used. A beacon packet informs the stations of the transmission schedule for the duration of the overall frame.

3. E-MAC Protocol Description

This section describes the basic characteristics of the E-MAC protocol. For simplicity, the following description refers to a typical hotspot scenario, where all the traffic transits through an AP. However, the E-MAC protocol is independent of the 802.11 operation mode (ad-hoc or infrastructure). We consider a network scenario in which $n_{\rm be}$ legacy IEEE 802.11 best-effort stations share the channel with $n_{\rm rt}$ E-MAC real-time stations. Of course, E-MAC real-time stations can also generate other traffic types at the same time. The non-RT traffic from the E-MAC real-time station will be put into different queues and treated separately, just like another independent contending 802.11 station. Without losing generality, we focus on the real-time traffic from E-MAC real-time stations in this paper. The total number of active stations is thus $n = n_{\rm be} + n_{\rm rt}$. We start from a high-level overview of the E-MAC protocol procedures, and subsequently provide a more detailed discussion of the E-MAC protocol characteristics.

The E-MAC protocol is compatible with 802.11 standard compliant devices, ensuring inter-operability with legacy stations and APs.

3.1. Overall E-MAC Characteristics

The E-MAC protocol divides the channel access into two phases: a slotted TDMA-like access phase, available only to E-MAC enabled real-time (RT) stations, and a legacy 802.11 contention phase, available to all the contending stations and arbitrated according to the DCF access rules.

Framing control and synchronization: The length of the TDMA access phase and the legacy DCF access phase is regulated by a specific E-MAC station, which is referred to as the "Maestro station". The Maestro station guarantees the loose synchronization of the E-MAC stations to the start of the contention-free phase and specifies the rules for admission control. This ensures a predictable level of fairness within the overall system: capping the traffic offered by RT stations during the contention free period, and letting them contend fairly (using best-effort channel access parameters) with best-effort (BE) stations during the contention-based interval. The admission rules are used to divide the resource among the RT stations and guarantees a minimum length for the DCF phase which prevents low-priority best effort traffic from starvation.

Scheduling and resource utilization: One of the main features that differentiates the E-MAC protocol from similar channel access mechanisms like HCCA or PCF is the organization of the transmission schedule within the contention-free period. During the contention-free period, the transmission sequence is organized in a distributed manner by the RT stations.

Each E-MAC station overhears the admission rules as well as the highest sequence number S of the active E-MAC stations before join the transmission. If this E-MAC station gets admitted according to the rule, it is assigned the sequence number $S + 1$. The sequence number decides the backoff time of each E-MAC station, which implicitly decides their transmission order.

The loose resource reservation via backoff and the distributed transmission schedule have an important impact on the resource utilization efficiency: different from the other reservation-based access schemes, which inevitably waste the resources (slots) previously scheduled but subsequently not utilized, each MAC-enabled station can take over the transmission opportunity that her predecessor has skipped after short time interval (i.e., the difference of their respective backoff times), thus shortening the contention-free period and extending the duration of the contention-based period.

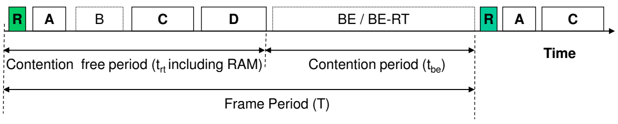


Figure 1. Conceptual model

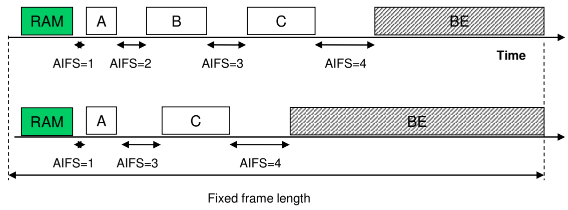
3.2. Frame Period, Contention-Free Period and Contention-Based Period

One round of contention-free and contention-based access together form what we refer to as frame period (T).One frame period starts with the transmission of a Reserved Access Marker (RAM) by the Maestro station: the role of the RAM is to define the duration of the frame period, the rules for admission control (e.g., the maximum amount of allowed RT traffic from each RT station and the minimum reserved contention period) and synchronizing all the RT stations to the beginning of the contention-free period ($t_{\rm rt}$ in Fig.1, including the RAM itself). Supposing there is no idle time between $t_{\rm rt}$ and the contention period $t_{\rm be}$, T = $t_{\rm rt}$ + $t_{\rm be}$ (Fig. 1). During the contention-free period, packets from different RT stations access the channel sequentially according to the agreed schedule (e.g., A, B, C, D in Fig. 1). For simplicity, we assume each RT station is only allowed to send one packet in $t_{\rm rt}$. If a RT station (e.g., B in Fig. 1) misses its chance to send its packet, for example because it does not have packet to send, the next RT station in the schedule (e.g., C in Fig. 1) takes over after waiting for an additional timeslot. After all RT stations transmit their admitted packets in the contention-free period, BE stations compete for access to the channel during the contention period.

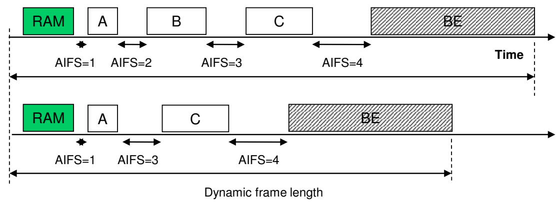
The frame period can be configured according to different design choices. One can opt for a fixed frame length structure or for a dynamic frame length structure. It is also possible to use a fixed ratio R between the contention free period and the contention based period. Fig. 2(a) and Fig. 2(b) show the fixed and the dynamic frame period structure. For simplicity, in the remainder of the paper we use a fixed frame length and a fixed minimum contention period.

3.3. Self-Organization and "Maestro" Station

Before starting its transmissions, a real-time station overhears the channel for a given amount of time (RAM timeout) in order to join a group that is already using E-MAC. The RAM packet is broadcasted to all the nodes every frame period T. It has the role of partitioning the entire frame period in the contention-free and contention-based periods,



(a) Frame period with fixed length



(b) Frame period with dynamic length

Figure 2. Fixed frame length (a) vs. dynamic frame length (b)

synchronizing all the RT stations to the start of each new contention free period. It also has the purpose of "pushing" the best-effort stations to the contention based period. To this end, the duration field of the RAM is set to a SIFS+$(n_{\rm rt}+1)\,t_{\rm slot}$. Real time stations do ignore the NAV field. The RAM is sent with the highest priority, i.e., after a SIFS time plus one IEEE 802.11 time slot $t_{\rm slot}$ (SIFS $=10\mu$s and $t_{\rm slot}=20\mu$s for 802.11b). This allows the Maestro node to deterministically take the control of the channel, preventing the best-effort stations from accessing it, as further discussed on section 3.5.

The Maestro node maintains a table of all active real-time stations, including their MAC address, sequence number i (explained in Subsection 3.4) and transmission time required for their real-time packets. Additionally, the time of the last real-time packet transmission of the respective station is stored. Note that other stations should also maintain such a table in case they become the Maestro (see Section 3.7). Compared to the polling process of PCF, the cost to maintain such a table at the Maestro station can be ignored. The Maestro node broadcasts information about the total number of real-time stations $n_{\rm rt}$ and the total time required for transmitting all real-time packets including the RAM, which is $t_{\rm rt}$. If all real-time stations transmit their packets, the total required transmission time is (cf. Fig. 3):

$$t_{\rm rt}=\mathrm{SIFS}+\mathrm{t_{slot}}+\mathrm{t_{ram}}+\sum_{\mathrm{i=1}}^{\mathrm{n_{rt}}}(2\mathrm{SIFS}+2\,\mathrm{t_{slot}}+\mathrm{t_{data,i}}+\mathrm{t_{ack}}) \tag{1}$$

In case the Maestro station leaves, due to the lack of a RAM packet there will be no more real-time transmissions.

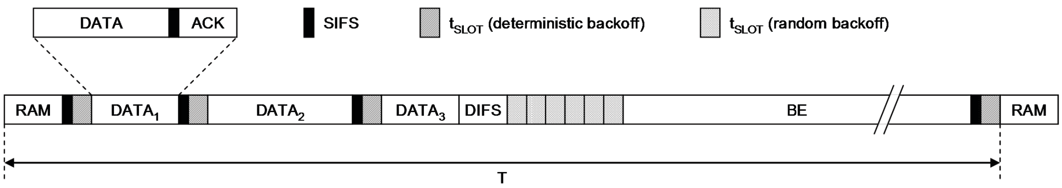


Figure 3. Basic system model

If the station immediately following the Maestro in the schedule neither receives a RAM packet nor any real-time packets for a duration of a RAM timeout interval (which is at least as large as a frame period), it will take over the role of Maestro, and the schedule is adjusted accordingly. If the station following the Maestro does not receive the RAM packet due to a packet loss, but the Maestro station did not leave, other real-time stations that did receive the RAM will still send their packets according to the schedule. They can be overheard by the station following the Maestro, and in this event, it will refrain from taking over the Maestro's role.

The transmission duration of RAM (t_{ram}), data packets ($t_{\mathrm{data},i}$), and ACK (t_{ack}) depends on the current channel rate and thus may vary. How to cope with variable data transmission times, different data rate requirements, and mobility is discussed in Section 7.

To maintain compatibility with conventional 802.11 devices, the RAM is a normal 802.11 data frame with specific information in the payload. It is transmitted using MAC layer broadcast and is not acknowledged by other stations. Conventional stations are not aware of the notion of frame periods and RAMs and may not finish their packet transmission before the end of a frame period, thus overlapping the next frame period by a time of ΔT. In this case, the Maestro station sends the RAM with a delay ΔT and schedules the next RAM after duration $T - \Delta T$ (shortening the BE traffic period) to compensate for this delay, keeping the average frame period duration T constant.

3.4. Sequence Establishment and Admission Control

The Maestro station assigns itself a sequence number of one. A new real-time station which wants to join the existing E-MAC group assigns itself a sequence number i ($i = 2, 3, \ldots$), without involving the Maestro, by simply adding one to the total number of real-time stations n_{rt} previously advertised by the Maestro in the RAM.

If a real-time station wants to join and is not the Maestro, it first has to check whether sufficient transmission time in T is available to accommodate its real-time packets. If $t_{\mathrm{rt}} + \delta_{\mathrm{guard}} + \delta_{\min}^{be} + 2\,\mathrm{SIFS} + t_{\mathrm{slot}} + t_{\mathrm{ack}} + t_{\mathrm{data}} + \leq T$, the real-time station may join. Otherwise, it has to refrain from transmitting real-time packets, contending instead using BE priority. Here $\delta_{\min}^{be}$ is the minimum reserved contention period for BE traffic (of course the degraded RT traffic can also use this period) and δ_{guard} is used to accommodate occasional retransmissions of real-time packets due to channel errors and interference. The $\delta_{\min}^{be}$ guarantees the opportunity for BE stations to transmit their fair share of packets. Note that $\delta_{\min}^{be}$ and δ_{guard} are only for admission control purposes and do not actually represents reservation periods.

Once the real-time station is admitted to the E-MAC group, the self-selected sequence number i will be used to configure its *fixed* backoff time to $t_{\mathrm{back},i} = (i-1)\, t_{\mathrm{slot}}$. Choosing the backoff this way results in a fixed transmission sequence, avoiding collisions among real-time stations. However, there is a small probability that two stations join at the same time and hence select the same backoff. This results in a collision (detected by absence of an ACK). To resolve this conflict, the two colliding stations wait for a duration $r\,T$ (r being a random integer number, e.g., between 1 and 10) before trying to join again. The use of this $r\,T$ is triggered only upon occasional collisions between two (or more) RT stations that happen to join at the same time. It should not be confused with the *fixed* backoff mentioned earlier, which is always used in order to define the RT station's order in the RT transmission sequence.

3.5. Slot Reuse

Real time stations begin their contention-free access upon the reception of a RAM. If a real-time packet is already waiting in the buffer upon receiving the RAM, the real-time station starts decrementing its backoff after the channel gets idle for AIFS ($= \mathrm{SIFS} + t_{\mathrm{slot}}$) time. If another real-time station is transmitting, the backoff is frozen until the channel becomes idle again. Hence, if all real-time stations have a packet in their buffer upon receiving the RAM, any two consecutive real-time packets are being transmitted with an idle time of AIFS between them. This case is illustrated in Fig. 3.

If a real-time station does not have a packet to send, it skips its turn. The subsequent station in the sequence will then transmit next with an idle time of AIFS$+t_{\mathrm{slot}}$ after the previous transmission. Generally, if k consecutive stations refrain from transmitting a RT packet after the RAM, the idle time between two packets becomes AIFS+$k\, t_{\mathrm{slot}}$. This idle time might be longer than the DIFS of legacy 802.11 stations for large k (or n_{rt}) resulting in possible collisions between RT and BE stations. This event is prevented by the real-time stations by setting the duration field of the packet to $2\,\mathrm{SIFS} + t_{\mathrm{ack}} + (n_{\mathrm{rt}} + 1)\, t_{\mathrm{slot}}$, which in turn sets the Network Allocation Vector, NAV, of legacy 802.11 stations. The last RT station in the sequence announces a duration (for the NAV of legacy 802.11 stations) of SIFS+t_{ack}. The duration field is ignored by real-time stations. Hence, best-effort stations will refrain from transmitting until all active real-time stations have transmitted.

3.6. BE Traffic Preservation and RT Fairness Considerations

Stations may generate real-time packets at the application layer at any time during a frame period. The first packet of a given station is assigned high priority, as described before. However, if a second packet of the same station arrives during the same frame period, it is either queued until the next frame period or contends as BE. We refer to such "degraded" packets as BE-RT packets in the rest of the paper. BE-RT packets are treated the same as the BE packets in the contention-based period. However, upon hearing a new RAM, a station with a waiting BE-RT packet can "promote" it back to high priority, thus being able to transmit it during the RT period. Promoting BE-RT packets to RT avoids that packets of the same flow get reordered at the MAC layer. Without promotion, the older BE-RT packet could be transmitted after the current RT packet.

There is no guarantee of minimum throughput for each BE station, but there is a minimum t_{be} for all contending stations (i.e., BE packets and RT-BE packets) in each frame period. For simplicity, we assume that RT stations are allowed to transmit only one high priority RT packet per frame period. The general case of different stations with different RT traffic requirements is discussed in Section 7. Stations can have (unrestricted) BE traffic sources in addition to RT traffic sources, in which case the two traffic types should be managed separately by two different queues.

3.7. Releasing Reservations

If a real-time station has finished its session, the previously reserved resources must be released. If a station has not transmitted a real-time packet for a duration of $l\,T$ (l being a predefined integer valid for all stations, e.g., 100), the Maestro supposes that it has left the real-time session. The Maestro then informs the other real-time stations about this fact in the next RAM together with the sequence number of the station that left. Then, all real-time stations with a higher sequence number can decrement their fixed backoff by one.

If a station has not transmitted a real-time packet for more than $l\,T$ although it has not finished its real-time session yet, it has to re-join as if it were a new station, before transmitting the next real-time packet.

In case the Maestro finishes its real-time session, it adds the number of remaining RAMs it will still broadcast to the last j (e.g., 10) RAMs. Thus, other real-time stations know when they have to decrement their sequence number. Furthermore, the real-time station with sequence number 2 then knows when it has to take over the role of the Maestro. In the event of sudden Maestro disconnection (e.g. mobility) there is an additional timeout, after which the next station in the schedule takes over the role of the Maestro.

3.8. Difference to TDMA and 802.11e

In comparison with TDMA, our mechanism has the advantage of slot reuse, making it more efficient since no time slots are wasted, for example in case of silence suppression by the corresponding (e.g., voice) application codecs.

The differentiation based on traffic categories defined by IEEE 802.11e does not give any guarantees for real-time traffic since at high load there is a high number of collisions even for real-time flows. Hence, under high load traffic the delay performance of IEEE 802.11e deteriorates. Moreover, previous work [18] showed that in heavily loaded networks, low priority traffic has extremely low transmission probability when using EDCA, an effect called starvation of low-priority applications. Conversely, the proposed E-MAC guarantees a minimum data rate and a very low delay for all real-time stations almost irrespective of the network load while avoiding the starvation of best-effort stations. In summary, E-MAC has the following advantages compared to IEEE 802.11e:

- Almost no collisions for real-time stations during the contention-free period, due to the order imposed by the sequence of backoff values. In 802.11e, high-priority stations still suffer from increasing collisions when the number of real-time stations increases.
- Strict throughput and delay guarantees for admitted real-time traffic, due to the "reservation" of periodic slots. In contrast, 802.11e offers only statistical guarantees.
- A very low guaranteed delay even under heavy-load traffic conditions. In IEEE 802.11e the delay performance deteriorates as the number of high-priority stations increases.
- Better protection for best-effort traffic, due to the limitation on RT transmission and frame period, and the

specification of minimum contention period $\delta_{\min}^{be}$. In contrast, real-time stations in 802.11e can consume the whole channel capacity depending on the data rates at the sources.

4. Mathematical Analysis

As the RT stations are also allowed to contend during the contention-based period, the analysis focuses on a) the contention-free period where n_{rt} RT stations transmit in a TDMA-like way, and b) a contention-based period where all the $n_{\mathrm{rt}} + n_{\mathrm{be}}$ stations contend for channel access (including n_{rt} BE-RT stations). The analysis focuses on the behavior of the network under saturated conditions, i.e., at any time instant both RT and BE stations have at least one packet in their transmission buffer.

4.1. Throughput Analysis

During the contention-free period all RT stations transmit their packets in a TDMA-like way. Under the assumption of saturated conditions the RT stations always have a packet to transmit so they also participate in the contention period. Hence, all $(n_{\mathrm{rt}} + n_{\mathrm{be}})$ stations participate in the contention during the contention period. After the end of the current time frame, any BE-RT packet which could not be sent during the contention period is promoted to RT priority again and transmitted during its corresponding time slot in the contention free period.

As all RT stations are always transmitting, there are no time-slot takeovers during the contention-free period. Thus, the separation between the end point and the start point of any two consecutive RT data packets is $AIFS + t_{\mathrm{slot}}$. Assuming that acknowledgments are used and that the packet size P_{rt} and the data rate R_{rt} has the same value for all RT stations, t_{rt} can be calculated as previously explained in Eq. (1). As the frame period has a fixed length T, the length of the contention-based period is $t_{\mathrm{be}} = T - t_{\mathrm{rt}}$.

As in previous work [4], [14], we assume that during the contention-based period, the probability of a packet collision p is constant and does not depend on the number of previous transmissions. Our analysis follows the work based on mean values carried out by Lin et al. [21]. During the contention-based period, the number of transmissions experienced by each packet follows a geometric distribution with probability of success $1 - p$. As the contention window doubles in size after every retransmission, the average contention window size W for the $n_{\mathrm{rt}} + n_{\mathrm{be}}$ stations is given by:

$$W = (1-p)\,CW_{\min} + p\,(1-p)\,2CW_{\min} + p^2\,(1-p)\,2^2CW_{\min} + ... + p^m\,(1-p)\,2^mCW_{\min}$$

$$= CW_{\min}\,(1-p)\,\frac{\left(1-(2p)^{m+1}\right)}{1-2p} \qquad (2)$$

In Eq. (2), the parameter m is the maximum number of allowed retransmissions. Consequently, the probability of transmission during the idle time period of the contention period can be calculated as:

$$\tau = \frac{1}{W} = \frac{(1-2p)}{(1-p)\left(1-(2p)^{m+1}\right)CW_{\min}}. \qquad (3)$$

The probability that a packet transmitted by a station during the contention-based period collides is equivalent to the probability that at least one of the other stations transmits in the same idle slot and hence is given by:

$$p = 1 - (1-\tau)^{n_{\mathrm{rt}}+n_{\mathrm{be}}-1} \qquad (4)$$

From (4), τ can be also expressed as:

$$\tau = 1 - (1-p)^{1/(n_{\mathrm{rt}}+n_{\mathrm{be}}-1)} \qquad (5)$$

Using numerical methods, the probability of collision p can be calculated from Eq. (3) and Eq. (5), and hence the average contention window W can be obtained from Eq. (2). On average, the separation between the end point and the start point of two consecutive packets during the contention period is given by $SIFS + t_{\mathrm{ack}} + DIFS + t_{\mathrm{slot}}\,W/\left(n_{\mathrm{rt}} + n_{\mathrm{be}} + 1\right)$ where t_{ack} is the time length of an ACK frame. Based on the assumption of backlogged queues at all stations, the average number of packets transmitted by a RT station F_{rt-be} that can fit into a single contention-based period and the average number of packets from a best-effort station that fit into the contention-based period is:

$$F_{rt-be} = F_{be} = \frac{T - t_{\mathrm{rt}}}{t_{\mathrm{data}} + SIFS + t_{\mathrm{ack}} + DIFS + t_{\mathrm{slot}}\,\frac{W}{(n_{\mathrm{rt}}+n_{\mathrm{be}}+1)}} / (n_{\mathrm{rt}} + n_{\mathrm{be}}) \qquad (6)$$

Hence, the average throughput for one RT station reaches:

$$S_{rt} = \frac{P_{rt} + F_{rt-be}P_{rt}}{T} \qquad (7)$$

The average throughput for a BE station is:

$$S_{be} = \frac{F_{be}P_{be}}{T} \qquad (8)$$

4.2. Delay Analysis

Since in saturated conditions buffers of both RT and BE stations are full, the average delay for BE packets can be calculated based on the average inter-transmission period between two consecutive packets from the same BE station and the buffer size B in packets is:

$$Delay_{be} = B/(S_{be}/P_{be}) = B\ \frac{T}{F_{be}} \tag{9}$$

Similarly, the average time elapsed between two consecutive BE-RT packets transmitted by the same station can be approximated as:

$$Delay_{rt} = B/(S_{rt}/P_{rt}) = B\ \frac{T}{1 + F_{rt-be}} \tag{10}$$

As can be seen in Eq. (9) and Eq. (10), two consecutive BE packets from a given station have an inter-transmission period generally greater than the frame period length T while in case of RT packets the inter-transmission period is guaranteed to be less than T. The latter feature represents a great advantage for RT applications since other statistical QoS-guarantee mechanisms such as 802.11e cannot guarantee delay for RT packets, especially under heavy traffic loads.

5. Performance Evaluations with Simulations

To evaluate the performance and scalability of our protocol we perform ns-2 simulations with the following parameters:

- Stations have a transmission range of 250 m.
- The network area size is 200 m × 200 m, therefore all nodes are within receive range of each other.
- The channel model is two-ray-ground.
- The number of RT and BE stations varies according to the scenario in consideration.
- All values are averaged over 400-second simulation runs.
- We consider a warmup period for the first 70 s, with low traffic load, to make sure all nodes have the ARP entry of the AP. The final measurements do not include this warmup phase.
- No routing protocol is used (single hop to the AP).

The channel capacity is set to 2Mbit/s and no RTS/CTS is used. Default values of AIFS(DIFS), CWmin and CWmax are used for 802.11e and 802.11g simulations. For the packet delay computation, we use a buffer size B of 50 packets.

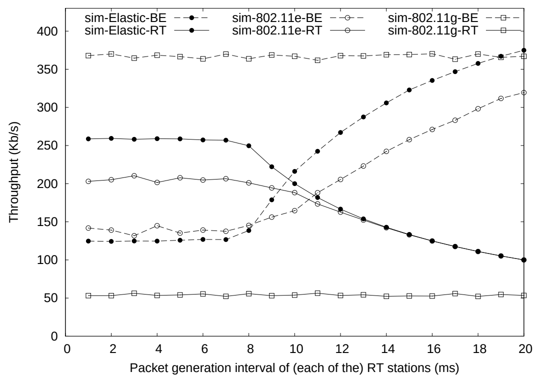


Figure 4. Throughput, as RT stations get more greedy

5.1. Protecting BE traffic from starvation

A very useful feature of E-MAC is that, even under heavy-load traffic conditions, it does not starve BE stations. Fig. 4 shows the throughput for both RT and BE stations with respect to the RT packet generation interval. As shown in Fig. 4 for small intervals (i.e., high packet generation rates), E-MAC yields a higher throughput for RT stations than 802.11e while keeping a minimum guaranteed data rate and without decreasing the throughput of BE stations, which is almost the same as in 802.11e. As the interval increases (i.e., the generation rate decreases) RT stations obtain exactly what they require in both E-MAC and IEEE 802.11e, but BE stations get a considerably higher throughput with E-MAC than with IEEE 802.11e. The reason for the latter behavior is that E-MAC yields a total throughput that is higher than that of 802.11e due to the scheduled RT transmissions in E-MAC. Therefore there are less collisions and retransmissions. Fig. 5 shows the throughput obtained by each of the three RT stations under saturated conditions when the number of BE stations increases for both simulations and mathematical analysis. It can be observed in Fig. 5 how E-MAC keeps the minimum guaranteed data rate (200 Kbit/s) for any number of BE stations. At the same time, BE stations do not starve but always get their fair share of the BE period in the frame period. Moreover, it can be seen that, as predicted in Eqs.(7) and (8), the throughput obtained by RT stations in E-MAC is equivalent to the share of the channel obtained by the BE stations (normalized by the packet's time length) in addition to the minimum guaranteed data rate. The latter result has been verified by the simulation results in Fig. 5.

Results from both simulations and mathematical analysis show that E-MAC guarantees a minimum data-rate for RT stations while protecting BE stations from starvation,

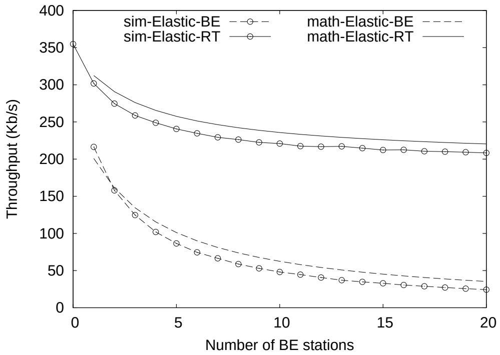


Figure 5. Throughput, with increasing number of BE stations. (3 RT stations, each can saturate the channel alone)

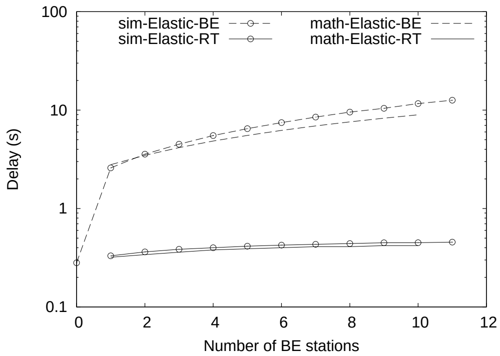


Figure 6. Packet delays, with increasing number of BE stations. (3 RT stations, each can saturate the channel alone)

even in the presence of a high number of BE stations. Fig. 6 depicts the delay obtained by E-MAC for both RT and BE packets. It shows how the delay for RT packets is guaranteed regardless of the number of BE stations as predicted in Eq.(10). This complements the results obtained through mathematical analysis regarding the delay guarantees for RT traffic provided by E-MAC.

5.2. Throughput, Delay, and Jitter

The performance comparison of 802.11, 802.11e, and elastic MAC is performed with three RT stations and a variable number of BE stations or vice versa. RT stations send a packet with 250-byte payload every 10 ms (i.e., at a rate of 200 Kbit/s) and BE stations send a 1400 bytes packet every 5.5 ms (i.e., at 2 Mbit/s). The RT packet generation interval is set to 10 ms. The average throughput with respect to the number of BE stations is illustrated in Fig. 7. IEEE 802.11e can provide a throughput of 200 Kbit/s for each RT station only if the number of BE stations is one. However, RT stations do not get their requested data rate when the number of BE stations increases. Using E-MAC however, RT stations always get 200 Kbit/s throughput independent of the number of BE stations.

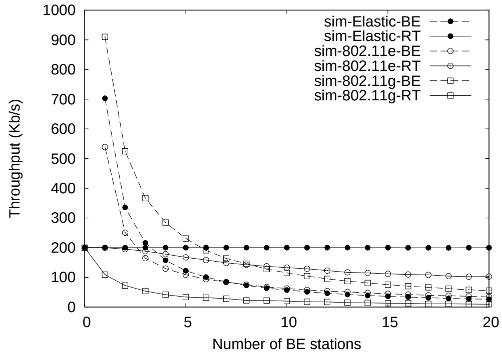


Figure 7. Throughput (per station) with variable number of BE stations

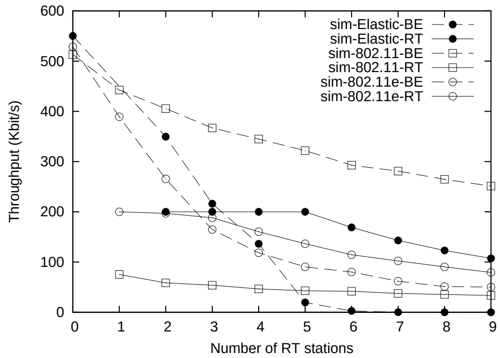


Figure 8. Throughput (per station) with variable number of RT stations

The average throughput with respect to the number of RT stations is illustrated in Fig. 8. It shows how E-MAC yields a higher throughput than 802.11e as the number of RT stations increases. This throughput is guaranteed to remain fixed unless some RT stations do not adhere to the admission control, therefore going beyond the system capacity of 5 RT stations in the example (we show this for the sake of understanding).

The average packet delay with respect to the number of BE stations is illustrated in Fig. 9. Using E-MAC, the simulated average packet delay of BE stations increases to 10

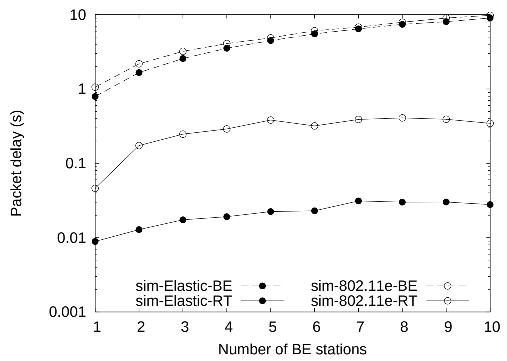


Figure 9. Packet delay with variable number of BE stations

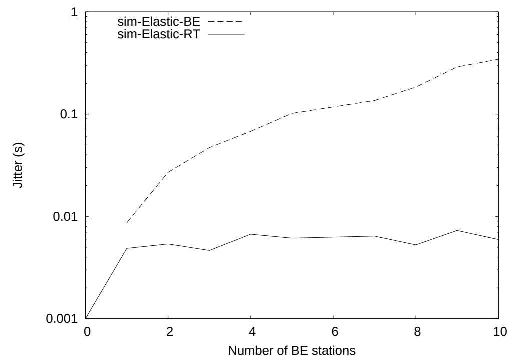


Figure 10. Packet jitter with variable number of BE stations

s when increasing the number of BE stations to 10. In contrast, the average packet delay of RT stations is always below 30 ms. Furthermore, the experimental results show that 802.11e is not able to guarantee low transmission delays for RT stations as opposed to our approach. With three BE stations, the average packet delay of RT stations is about 600 ms with 802.11e compared to 5 ms with E-MAC. The guarantee of such low delays for RT packets is a key feature of E-MAC that, to the best of our knowledge, no other QoS scheme compatible with IEEE 802.11 provides. The small increase of the delay for RT packets in Fig. 9 is due to synchronization problems, which tend to produce collisions between the last RT packets during the RT period and the first BE packets of the BE period as the number of BE stations increases. According to [29], we calculate the jitter after reception of packet i as exponentially weighted moving average of packet delay differences

$$J(i) = J(i-1) + \frac{|D(i-1,i)| - J(i-1)}{16}, \qquad (11)$$

where $D(i-1,i)$ is the difference of the transmission delay of two successively received packets. The average jitter depending on the number of BE stations is illustrated in Fig. 10. The simulated jitter of BE stations goes up to 350 ms when increasing their number to 10. RT stations using E-MAC on the other hand only experience a very low jitter of less than 10 ms independent of the number of BE stations. The very low jitter provided by E-MAC represents another great advantage when compared to 802.11e, which performs poorly in terms of jitter as the number of BE stations increases.

6. Testbed Implementation

To complement the analytical and simulative studies, we implemented the E-MAC protocol on current off-the-shelf WLAN hardware.

	802.11e		E-MAC	
	(CWmin, CWmax)	AIFS	(CWmin, CWmax)	AIFS
AC_BE	(4,10)	3		
AC_BK	(4,10)	7		
AC_VI	(3,4)	2	(0,0)	sta_cnt+3
AC_VO	(2,3)	2		

Table 1. WME parameter settings of E-MAC and IEEE 802.11e

We chose Proxim Orinoco 802.11b/g wireless LAN cards. As they are based on an Atheros chipset, these cards are supported on Linux OS thanks to the MadWifi [24] driver, version 0.9.4 at the time of the implementation. The MadWifi driver supports the 802.11e MAC amendments, thus providing four data queues respectively for BK (background), BE (best effort), VI (video) and VO (voice) traffic. We place the RT traffic into the VI queue in our implementation.

Table 1 shows the AIFS, CWmin, and CWmax values used in 802.11e and E-MAC. Without loss of generality, we select the infrastructure mode in our testbed. For the E-MAC operation, the frame period is set to 20 ms.

6.1. Distributed Packet Scheduling

The Madwifi driver offers a limited access to the backoff registers[1]. Then, to implement our scheduling policy, we set both the minimum and maximum values of the contention window to 0 and dynamically set the individual AIFS values to distributively establish the transmission order. Based on

[1] At the time we patched the driver, Atheros had not yet released the source code of its HAL driver APIs.

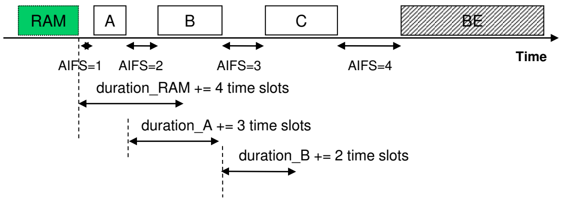


Figure 11. Decremental-Push

the RAM advertisement, each RT station sets it AIFS value to be equal to an AIFS value plus as many time slots as its self-assigned sequence number. It is important to notice that each transmission which follows the RAM will update the duration field of the legacy 802.11 stations to the duration field of the new packet. Then, in order to prevent best-effort traffic from backing off during the contention free period, the duration field of the real-time packet is set such to ensure that the last RT station can still transmit. Further details about the "pushing mechanism" are given in the next subsection.

It is worth noticing that there is a subtle difference between using backoff (in the protocol design) and AIFS (in the practical deployment) to implement the sequential distributed scheduling: since the backoff is decreased by one unit whenever the channel is idle, it would leave a single time slot between two consecutive RT transmissions. The use of AIFS would separate RT transmissions by an increasing number of time slots. Considering a moderate number of RT stations (e.g., 10) and the duration of a single time slot (e.g., $20\mu s$ in the 802.11b case), the additional overhead introduced by adopting the AIFS approach, compared to the backoff one, is negligible ($((10 \times 11)/2 - 10) \times 20\mu s = 900\mu s$).

As mentioned earlier, when a RT station gets more RT packets than allowed in the current frame period, the station either degrades the packets and contends as BE, or it defers them to the next frame period. The easiest way to degrade a given packet is to move it to the BE queue, which contends with the normal access rules. This, however, causes some reordering among packets of the same flow, since they are placed in different queues.

6.2. Pushing BE stations using the NAV

The pushing mechanism we implemented in E-MAC guarantees that the non-degraded RT traffic is sent before the BE traffic in each frame period. We realized this scheme by modifying the duration field of both the RAM and of the data packets sent by the E-MAC stations.

We implemented what we call the "Decremental-pushing" algorithm (Fig. 11): both the Maestro station and the real-time stations protect the transmission of all the following E-MAC nodes by setting the duration field to a value that guarantees each of them to access the channel, even though some of them may skip its reserved schedule. Each E-MAC station will protect the slots until the end of the contention free period, whose duration was previously advertised in the RAM.

There are different protection policies that can be used. In a purely decremental fashion, each E-MAC station selects a duration value that protects all the nodes that follow it. This implies that this duration value gets smaller and smaller as the schedule sequence proceeds. Another approach is to set the duration field to a fixed value, to only protect the immediately following E-MAC transmission. This is a more complicated design choice, that might more efficient in terms of bandwidth usage. However, here we select the purely decremental approach, which guarantees a good trade-off between resource utilization and complexity.

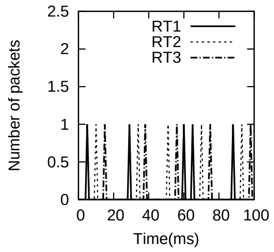
6.3. Distributed Packet Scheduling Analysis

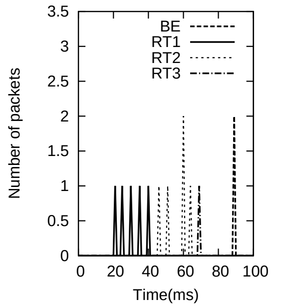
In order to assess the effectiveness of the AIFS-based schedule, we set up a testbed in which three RT stations and one BE station send uplink data traffic towards an access point. Each RT station sends 1000 byte UDP data packets at a rate of 50 pkt/s. The BE station sends 1400 byte UDP packets at a rate of 500 pkt/s. The MGEN traffic generator was used to implement the traffic sources. We considered two scenarios: in the first one, only the RT traffic is present. In the second scenario, BE background traffic saturates the channel. Fig. 12 shows the sequence of packets as received at the AP. When the channel is not saturated (no background BE traffic), the E-MAC protocol provides a deterministic packet order, as clearly visible on Fig. 12(a). One slight reordering of the packet sequence at time 60s comes from the occasionally delay of packets from the application layer queue. Nevertheless, the correct transmission schedule is immediately corrected in the next frame period. In the case of 802.11e of course there is no clear ordering in the channel access of RT stations, as visible in Fig. 12(c).

When the channel is saturated with BE traffic, accumulation phenomenons[2] are observed in both E-MAC (Fig. 12(b)) and 802.11e (Fig. 12(d)). Nevertheless E-MAC still keeps the transmission sequence as RT1, RT2, RT3 and at last BE as soon as the burst ends. We believe that this effect is due to hardware inefficiencies. It may happen in fact that the Ethernet card is not ready to accept new packets from the software queue. This causes these packets to

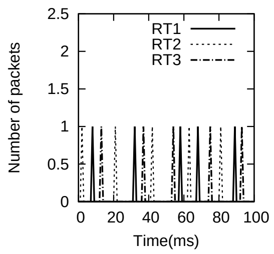
[2]In case Ethernet card hardware is not ready to accept new packets from the software queue, the packets will get accumulated at the kernel level and be transmitted all at once later as soon as the station gets the channel access.

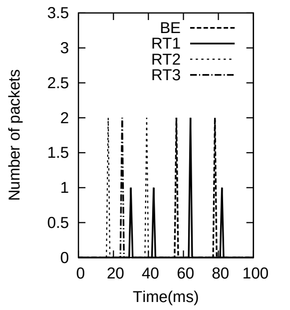
accumulate at the kernel level, for being later transmitted all at once as soon as the station gets the channel access.





(a) E-MAC transmission sequence: non-saturated channel (b) E-MAC transmission sequence: saturated channel





(c) 802.11e transmission sequence: non-saturated channel (d) 802.11e transmission sequence: saturated channel

Figure 12. Sequential Packet schedule in saturated and non-saturated conditions.

To complete the AIFS-based distributed scheduling process, we consider a dynamic network scenario in which the E-MAC nodes subsequently join the network. We set up an experimental scenario in which one BE station starts transmitting at time 0s, and stops at time 40s. Three RT stations sequentially join and start transmitting at time 10s, 15s and 20s, respectively finishing after 50s, 55s and 60s. We generated the same traffic pattern as in the previous experiment. Since the data-rate is set to 2Mbit/s, the BE traffic is able to saturate the entire channel capacity. Fig. 13 shows the throughput performance for all the tested stations when the 802.11e based access and the E-MAC based access are used. As soon as each RT station joins, as expected the throughput of the BE station decreases in both cases. However, as visible from the figures from time 20s to 40s in Fig. 13, E-MAC guarantees stable throughput to all the RT stations, while 802.11e gives very unsteady throughput for RT stations because of the probabilistic contention mechanism. This type of unsteady throughput might cause undesirable jitter and delays to RT traffic, thus affecting the user perceived QoS.

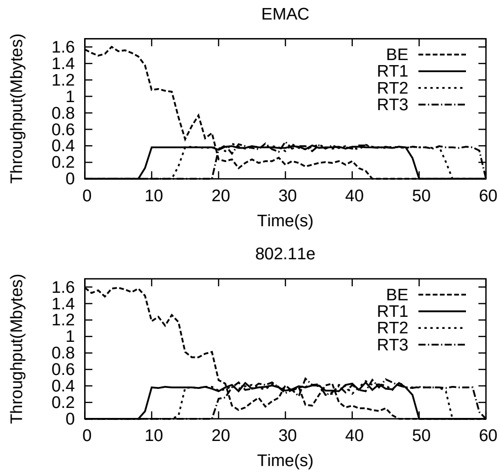


Figure 13. Throughput Behavior with IEEE 802.11e and E-MAC

6.4. Throughput, Delay, and Jitter

To assess the E-MAC protocol effectiveness, we performed a thorough assessment of the E-MAC behavior in terms of QoS-related parameters. Our investigation has been carried out in a different testbed scenario, in which five nodes contended for the transmitting to an access point. All the machines were synchronized using the Network Time Protocol (NTP), in order to get a precise estimation of delay and jitter.

Table 2 shows the average throughput of RT stations with different numbers of BE stations. In this test each RT station sends 500 bit/s UDP traffic and BE stations each send 2000 bit/s UDP traffic to AP. The tests are performed for (3RT, 1BE), (3RT, 2BE), and (2RT, 3BE) respectively. The UDP buffer size is 108 KBytes and the packet size is 1400 bytes. Values are averaged over 100-second long tests. It can be easily seen that the average throughput of RT stations is kept around 500 bit/s in the E-MAC case. However, using 802.11e, the RT throughput drops when the number of BE stations increases. The tiny drop of throughput in the (3RT, 2BE) case compared to the (2RT, 3BE) case for the E-MAC protocol comes from the small overhead due to the AIFS value of the third RT station. However, 802.11e suffers a throughput drop that is twice as high in the same case. The same settings are used for the jitter evaluation. Table 3 shows that E-MAC produces lower jitter (less than 20ms) for RT traffic independent of the number of BE stations. As the number of BE stations increases, 802.11e lacks the mechanism to guarantee low jitter for RT traffic. Obviously, the jitter increases as the number of RT stations increase for both 802.11e and E-MAC protocol. However, the penalty

Throughput (Kbit/s)	(3RT, 1BE)	(3RT, 2BE)	(2RT, 3BE)
802.11e	448.00	306.67	332.50
E-MAC	498.67	487.67	500.00

Table 2. Average throughput of RTs using variable number of BE stations

Jitter (ms)	(3RT, 1BE)	(3RT, 2BE)	(2RT, 3BE)
802.11e	19.80	89.04	37.86
E-MAC	13.46	19.35	13.66

Table 3. Average jitter of RTs using variable number of BE stations

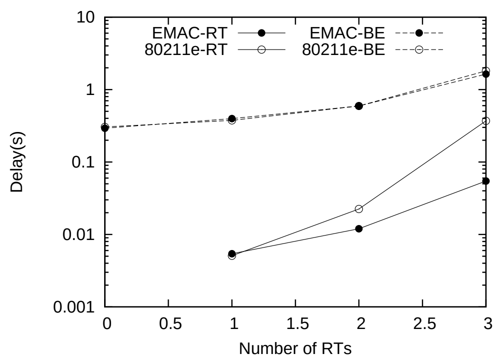


Figure 14. Delay using variable number of RT stations

is much less in E-MAC (around 6 ms) than for 802.11e (around 50ms). In 802.11e, RT traffic has higher transmission probability than BE traffic. However, within the same traffic class, the transmission opportunities are shared. Therefore, the increased number of RT stations causes a high probability of collision and leads to random backoffs. In contrast, E-MAC adopts a fixed transmission sequence which greatly reduces the probability of collision.

Similar results were observed in the delay performance evaluation. There, each RT station sends 1000-byte packets every 20ms to the AP and 1 BE station sends 1400-byte packets every 2ms to the AP. The channel is saturated by BE traffic. This test is repeated with different number of RT stations. Values are averaged over 200-second tests. Fig. 14 shows that E-MAC gives similar delays for BE stations compared to the 802.11e ones. However, the average delay for RT stations is much smaller. Using 2 RT stations, the delay for 801.11e is twice as much as that of E-MAC. With 3 RT stations the ratio is larger than 6 times. The average delay of 802.11e with 3 RT stations (369.34 ms in our test), already exceeds the recommended maximum one-way voice packet delay by ITU-T G.114 which is 150 ms. In comparison, using E-MAC results in 54.55 ms delay for the same case. Because of the limited number of available notebooks, we were not able to test cases with more RT stations. However, we can already see from this test that E-MAC has a much lower delay for RT traffic than 802.11e.

6.5. User Perceived QoS with E-MAC

The final target of E-MAC is to improve the user experience. However, the absolute value of throughput, delay and jitter do not really represent the real experience of the user. Therefore, we measured the user perceived QoS in our testbed. The most popular method for video quality evaluation is based on the computation of PSNR (Peak Signal to Noise Ratio). It compares the maximum possible signal energy to noise energy pixel by pixel for each video frame. For each frame of $M \times N$ pixels, $PSNR = 10 \log \frac{255^2}{MSE}$, where $MSE = E[|r(m,n) - r'(m,n)|^2]$. $r(m,n)$ can be the image luminance function, brightness, etc.

We performed two tests, a coarse one and a fine one. In the coarse one we use VLC [1] to stream and decode the video in TS-format (Transport Stream). We compare the number of decoded audio blocks/video frames at the receiver to the number of encoded audio blocks/video frames at the sender to get an idea of the effectiveness of E-MAC protocol. In the fine test, we calculate average PSNR of each video frame. The latter case uses the open source evaluation framework SVEF [3], which is based on JSVM (Joint Scalable Video Model) software to stream and to compute the PSNR of H.264 video stream as well as compensate for the lost frames at the receiver. The coarse test is set up as follows: 1 RT station sends one video stream to the AP. The video is coded with rate 512kb/s in TS-format. The channel bandwidth is set to 2Mb/s. At the same time, 1 BE station sends UDP packets at 2Mb/s to the AP, competing with the RT station. Each test case is repeated 10 times. In both tests, the channel is saturated by BE traffic from one BE station. This will cause some packet loss for RT traffic.

We performed both tests with a different number of MAC layer retransmissions. IEEE 802.11 MAC uses conventional ARQ to control the link layer error rate. If no ACK is received after a given period of time, the sender retransmits the packet after an exponential back off. This process is repeated for 10 times in the MadWifi driver before the station drops the packet. This is handled differently in E-MAC. In E-MAC the lost packets are retransmitted immediately after the waiting period of the ACK timeout. This is convenient for real-time traffic streams, where each packet has strict delay/jitter constraints. However, the absence of a random back off in E-MAC might cause the retransmitting RT station to delay other contending RT stations as well as BE

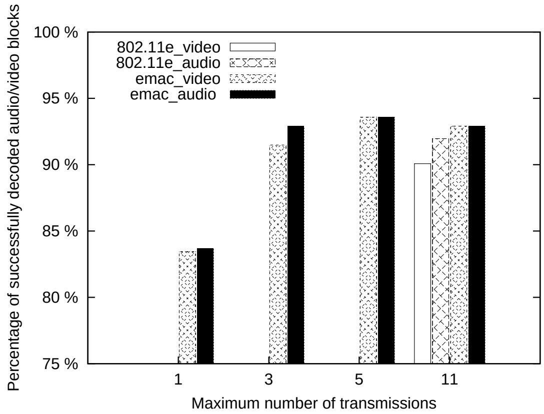


Figure 15. Decode rate of audio/video blocks with different number of transmissions

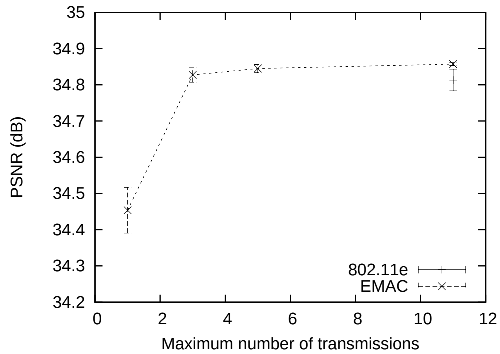


Figure 16. PSNR with different number of transmissions

stations from accessing the media, thus increasing the unfairness towards the BE traffic. However, disabling retransmissions is not a viable solution to retain fairness. Because of the high correlation between the video frames, the loss of one important frame can cause the failure in decoding several video frames and greatly affects user perceived quality. Here, our intention is to check the influence of the number of retransmission on the user perceived QoS.

Fig. 15 shows the results of the coarse test. It provides the amount of packet loss with different number of retransmissions. When the number of retransmission increases, the percentage of successfully decoded audio packets/video blocks increases as expected. E-MAC with 2 retransmissions (which means a maximum of 3 transmission in total, including the first transmission) can already ensure correct decoding of 90% of all the audio packets/video blocks. More audio packets can be correctly decoded compared to video blocks. More importantly, E-MAC with 2 retransmissions already outperforms traditional 802.11e with 10 retransmissions.

According to the requirement of SVEF, AVC coded video is used for the fine test. The decoder buffer size is set to 1 second, which means that the packets received with more than 1 second delay will be dropped. SVEF will replace a missing frame with the previous completely received frame for PSNR calculation. We note that the results are slightly biased towards high PSNR since we had to drop the tests where I-frames are corrupted (JSVM limitation). Fig. 16 shows the average PSNR for each test case together with the standard error (indicated by the error bar in the figure), averaged over at least 16 successful tests. Here, the video quality of E-MAC with 2 retransmissions is already better than 802.11e with 10 retransmission. Further increasing the number of retransmissions has little impact on the PSNR.

6.6. "Slot" Reuse and Throughput Gain

As mentioned before, one of the most important features of E-MAC is the time slot reuse. Slot reuse allows RT stations to take over unused transmission opportunities from other stations. By offering a flexible –"elastic" – scheduling for RT packets, it is possible to obtain a more efficient use of the available bandwidth resources, which results in an increased total throughput of the network. In order to measure the throughput gain obtained with this feature, we compare E-MAC with a *pseudo-TDMA* approach, which, similarly to E-MAC, provides slot reservation at the beginning of the frame, but does not provide any slot reuse.

We assume that up to five RT stations have to transmit RT packets generated by a ITU-T G.711 voice codec [20] with silence suppression. This codec generates a packet with 240-byte payload every 30 ms (i.e., 64 Kbit/s). However, if the user is silent, no packets are generated. One BE station is saturating the channel with a data rate of 2 Mbit/s. We use *Wireshark* to obtain voice traces of a VoIP RT communication with *GnomeMeeting* (now called *Ekiga*). We then generate dummy packets according to those voice traces on the RT stations in our testbed to compare the efficiency of E-MAC and pseudo-TDMA. The resulting throughput is shown in Fig. 17. As illustrated, E-MAC yields a considerably higher value than the pseudo-TDMA scheme for both total network throughput and total BE throughput, and the gap becomes larger as the number of RT stations increases.

Finally, we tested the E-MAC performance with video streaming traffic under heavy BE contention, comparing the results to the legacy IEEE 802.11g/e scenario. The result of this evaluation can be found in [2].

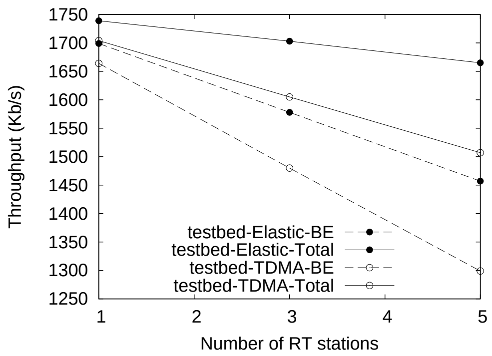


Figure 17. Throughput, using variable number of voice sources with silence suppression

7. Discussion and Future Work

Noisy channels, hidden nodes: The mechanism is not severely affected by a noisy channel as an acknowledgment frame must follow any successful transmission of a data frame. Not overhearing the RAM frame broadcast by the Maestro is not a problem either since any station can wait for the next RAM in order to synchronize its transmission with the others.

The system has been designed to operate in a single collision domain, i.e., with no hidden nodes. For a multi-hop network this assumption does not hold anymore. In the latter case, some real-time stations may not hear the transmission of the RAM or even the preceding station during the real-time period, causing the whole system to lose synchronization. Even a best-effort station that is not able to overhear a given transmission during a real-time slot may cause synchronization problems for real-time stations. The design of E-MAC for a proper operation in a multi-hop network is part of our future work.

Different data rates and packet sizes: The scenario in which RT stations have different packet lengths due to different data rates and/or packet sizes does not cause any problem to our system as the transmission sequence number, which is the most important parameter, is not changed because of this. In the worst situation in which several RT stations are transmitting large packets at very low data rates, the admission control rules conveyed by the Maestro through the RAM should reflect the stringent conditions and should defer new stations that are trying to join the system until the traffic conditions become better.

Another important issue is how to deal with different data rate requirements from RT stations. This problem is easily solved by allowing RT stations to transmit a burst of several packets during the RT period instead of only one as previously described. In this solution RT stations have an internal counter h_i set to the maximum number of packets to be transmitted during the RT period, which reflects their own bandwidth requirement. Any other packet in its buffer exceeding this limit should contend during the best-effort period and be promoted to the RT class after overhearing the next RAM as described previously.

Coexistence with other hotspots: A group of E-MAC stations may coexist with:

- other 802.11 (BE) stations operating on the same channel, using the same or different network IDs.
- other 802.11 (BE) and/or E-MAC stations operating on a neighboring channel.

In the first case, stations can overhear each other's transmissions and BE stations in both networks will be deferred using the duration field of RT stations, regardless of their network ID.

As for the second case, the 802.11 and/or E-MAC stations operating on a neighboring channel will interfere with the concerned ones. This results in increased channel errors and packet retransmissions, which should be taken into consideration in δ_{guard} for admission control, as mentioned in Section 3.

Energy saving: Energy consumption is an issue should take into consideration. In fact, we considered the Maestro station to stay awake (and not go into power saving mode) even if it has no data packets to transmit, in order to regularly send the RAM with which the E-MAC nodes are kept synchronized and informed about rules as well as the number of RT stations. An intuitive solution is to rotate the role of the Maestro among the E-MAC stations. Another alternative is to do the synchronization based on the (standard) beacons broadcasted by the AP, use fixed rules and parameters for admission control, while performing other "counting tasks" in a distributed manner. The challenges there are that the beacon period cannot be adapted to the RT stations needs. Furthermore it cannot be changed to compensate for the synchronization loss due to a BE transmission overlapping with the following frame period. We consider this alternative for future work.

8. Conclusion

In this paper we present a MAC protocol, called E-MAC, that offers strict QoS guarantees for real-time traffic (e.g., voice/video) in wireless networks. We implemented our system on a Linux testbed, making use of open-source network card drivers (MadWifi). The system is self-organizing, completely distributed, and requires no changes to existing legacy 802.11 stations.

We show how E-MAC provides strict QoS guarantees: a minimum-reserved throughput, and very low packet delays. All testbed measurements, mathematical model and the simulations showed consistently good results with E-MAC outperforming 802.11e, pseudo-TDMA and 802.11. Moreover, we deployed different implementation options and checked the real world behavior of E-MAC in the testbed. Our system supports not only VoIP, which is characterized by a fixed packet size, but also realtime video streaming with dynamic packet sizes. Our system is operational and ready to be used in existing 802.11 networks.

References

[1] http://www.videolan.org/vlc/.

[2] http://imad.aad.name/emac-videos/.

[3] G. B. Andrea Detti. SVEF: an Open-Source Experimental Evaluation. 2009.

[4] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE JSAC*, 18(3), 2000.

[5] G. Bianchi, A. Di Stefano, C. Giaconia, L. Scalia, G. Terrazzino, and I. Tinnirello. Experimental assessment of the backoff behavior of commercial ieee 802.11b network cards. May 2007.

[6] G. Bianchi, I. Tinnirello, and L. Scalia. Understanding 802.11e contention-based prioritization mechanisms and their coexistence with legacy 802.11 stations. *IEEE Network*, 19(4):28–34, 2005.

[7] S. G. C. Coutras and N. Shroff. Scheduling of real-time traffic in IEEE 802.11 wireless LANs. *Wireless Networks*, 2000.

[8] K. Chebrolu, B. Raman, and S. Sen. Long-distance 802.11b links: Performance measurements and experience. ACM, 2006.

[9] I. Dangerfield, D. Malone, and D. Leith. Experimental evaluation of 802.11e edca for enhanced voice over wlan performance. April 2006.

[10] D. J. Dechene and A. Shami. Experimental triple-play service delivery using commodity wireless lan hardware. IEEE, 2009.

[11] A. Di Stefano, A. Scaglione, G. Terrazzino, I. Tinnirello, V. Ammirata, L. Scalia, G. Bianchi, and C. Giaconia. On the fidelity of ieee 802.11 commercial cards. pages 10–17, July 2005.

[12] P. Djukic and P. Mohapatra. Soft-tdmac: Software tdma-based mac over commodity 802.11 hardware. IEEE, 2009.

[13] C. Doerr, M. Neufeld, J. Fifield, T. Weingart, D. Sicker, and D. Grunwald. Multimac - an adaptive mac framework for dynamic radio networking. pages 548–555, Nov. 2005.

[14] P. Engelstad and O. Osterbo. Non-saturation and Saturation Analysis of IEEE 802.11e EDCA with Starvation Prediction. In *Proceedings of MSWIM*, 2005.

[15] G.Anastasi and L.Lenzini. QoS provided by the IEEE 802.11 wireless LAN to advanced data applications: a simulation analysis. *Wireless Networks*, 2000.

[16] F. Guo and T. cker Chiueh. Software tdma for voip applications over ieee802.11 wireless lan. pages 2366–2370, May 2007.

[17] IEEE standard for information technology – LAN/MAN – specific requirements – part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1999.

[18] IEEE standard for information technology – specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Amendment 8: Medium access control (MAC) quality of service enhancements, 2005.

[19] IPW2100 – Intel PRO/Wireless 2100 Driver for Linux. `http://ipw2100.sourceforge.net/`.

[20] Pulse code modulation (PCM) of voice frequencies. ITU-T recommendation G.711, 1988.

[21] Y. Lin and V. Wong. Saturation Throughput of IEEE 802.11e EDCA Based on Mean Value Analysis. In *Proceedings of WCNC*, April 2006.

[22] `http://linuxwireless.org/`.

[23] L. Loyola, M.Ogawa, K.Nagata, and S.Aikawa. Proposal and Evaluation of a Mesh Wireless Local Area Network Architecture with Dual DCF-HCCA Channel Access Scheme in the Vicinity of Gateway Access Points. *IEICE Transactions on Communications*, (10), 2006.

[24] MadWifi – Multiband Atheros Driver for Wifi. `http://www.madwifi.org`.

[25] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald. Softmac - flexible wireless research platform. November 2005.

[26] R. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer. Wildnet: Design and implementation of high performance wifi based long distance networks. ACM, 2007.

[27] Host AP driver for Intersil Prism2/2.5/3 Chipsets. `http://hostap.epitest.fi/`.

[28] A. Rao and I. Stoica. An overlay mac layer for 802.11 networks. pages 135–148, New York, NY, USA, 2005. ACM.

[29] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. IETF RFC 3550, July 2003.

[30] A. Sharma, Belding, and E. M. Freemac: framework for multi-channel mac development on 802.11 hardware. pages 69–74, New York, NY, USA, 2008. ACM.

[31] A. Sharma, M. Tiwari, and H. Zheng. Madmac: Building a reconfiguration radio testbed using commodity 802.11 hardware. pages 78–83, Sept. 2006.

[32] A. Velayutham. An Enhanced Alternative to the IEEE 802.11e MAC Scheme . Iowa State University, 2006.

[33] Y. Y. Kwon and H.Latchman. A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs. In *Proceedings of INFOCOM*, 2003.

[34] H. Zen, D. Habibi, A. Rassau, and J. Wyatt. Adaptive segregation-based mac protocol for real-time multimedia traffic in wlans. pages 461 –466, 19-21 2007.