Università di Pisa
Dipartimento di Ingegneria dell'Informazione
Elettronica, Informatica, Telecomunicazioni

Tesi di Dottorato di Ricerca in
Ingegneria dell'Informazione
XIX ciclo

Traffic Control and Quality of Service in Wireless LANs

Candidato
Luca Tavanti

Tutori
Prof. Franco Russo
Prof. Stefano Giordano
Ing. Rosario G. Garroppo

Febbraio 2007

Acknowledgements

The author would like to thank Thales Italia S.p.A, which kindly sponsored his PhD scholarship, thus allowing the realisation of this work.

Contents

Introduction

The IEEE 802.11 standard has rapidly become the most popular technology for broadband wireless local area networks (WLANs). As a corollary, we have seen a steep rise in the efforts to increase system capacity to satisfy Internet users' hunger for bandwidth. This led to the ratification of the *a*, *b* and *g* amendments to the standard, which allow rates up to 54 Mbps, and to the pursue of enhancements to further increase the maximum data rate (Task Group *n*). More lately, the growth of multimedia services over the "wired" Internet has give strength to the idea of supporting real-time applications over wireless LANs too. However, it has also become evident that even the latest versions (*a*, *b*, and *g*), with the sole use of more efficient modulation schemes and/or other frequency bands, can hardly be employed to offer adequate support to services with strict Quality of Service (QoS) requirements, as they do not provide any means to guarantee the timely and reliable delivery of frames[1]. The IEEE 802.11 working group has therefore issued the new *e* amendment which defines some mechanisms for differentiating the traffic on the basis of the priority in accessing the radio channel. In this way, the IEEE intended to complement the 802.11 standard suite in order to let it satisfactorily support any kind of service.

[1] Actually, the PCF mode was originally conceived with this purpose; in practice, however, manufacturers never released commercial products supporting this mode.

Yet, this goal has been only partially accomplished, as 802.11 systems still present several drawbacks.

One of the most critical factors driving the efficiency of these networks is the ability to overcome the hurdles imposed by the wireless channel. The actually exploitable capacity of the links is highly variable in both time and space, thus leading to unpredictable frame delivery ratio and delays. In addition, the heterogeneity of the terminals and the possible use of multiple transmission rates are factors that may limit the overall system throughput. The same 802.11 Distributed Coordination Function (DCF, the basic and most employed operation mode) does not deal effectively with this problem. In fact, when employed in the above mentioned conditions, it often leads to the so called "performance anomaly". When this phenomenon occurs, the throughput of all the stations tends to align to that of the slowest station in the network. Moreover, the unique First-In First-Out (FIFO) queue commonly implemented at the Access Point (AP) creates an undesirable inter-dependence (in terms of delays) of traffic flows addressed to different stations. The overall efficiency loss is thus apparent.

A solution to these problems has been reckoned to reside in a smart scheduling algorithm to be deployed at the AP. In fact, several schedulers for wireless networks have already been proposed in literature. In most cases, however, these solutions rely on a model of the wireless channel. This has the drawback to be more or less distant from the actual channel behaviour, thus making the scheduler inefficient or unfeasible. Consequently, a more reliable solution would be centring scheduling decisions on a real measure of the state of the links.

Starting from these observations, we have designed and developed a simple scheduling algorithm to take into account the actual channel behaviour. The main innovation of our scheduler is the way it measures link quality. This is not appraised with usual metrics, such as signal-to-noise ratio, but is

quantified as the time needed to deliver a frame to the destination. Hence, the resource to share is not the total capacity of the channel, but the time the channel is in use. This is the actualisation of a different fairness idea, named *proportional* fairness. We show that this approach leads to noteworthy improvements, and in particular we show how it is possible to isolate the flows from each other, so that each flow can take advantage of its channel share irrespectively of the quality experimented by the others. This algorithm has been implemented and tested through simulations and, above all, on a prototype AP. To the best of our knowledge, this is the first working implementation of a scheduler realising proportional fairness.

The second part of this thesis deals with the interesting issue of transporting real-time services over 802.11e networks. The recent *e* amendment specifies the rules to realise traffic differentiation in wireless LANs. The basic philosophy is to give quicker access to medium to high priority traffic, i.e. traffic that is more sensitive to delay. While there is broad consensus about its fair capability to support real-time applications with a reasonable quality, it has also been shown that this capability is extremely limited. 802.11e provides less predictable performance than a reservation based method and suffers from network congestion. Scarce reliability of QoS guarantees, starvation of low priority traffic and unbalanced uplink/downlink bandwidths are the most serious drawbacks hampering its use. For these reasons, it can be argued that the support of QoS cannot be easily achieved if disjoint from the relevant issue of admission control (in short, a.c.). In detail, an algorithm must be run to determine the maximum number of users/services that can be admitted to the network while satisfying the respective QoS requirements.

Our activity in this field started with the determination of the *admission region* for voice and video in a 802.11e WLAN. The number of videoconference and VoIP sources that can be accepted in the 802.11e coverage

area has been evaluated, in presence of TCP traffic, considering the actual QoS requirements that can be assumed for these services. This preliminary study casts light on two aspects. On one hand transport differentiation is fairly efficiency as delay-sensitive traffic is scarcely influenced by low priority services. On the other hand, it has emerged the presence of a bottleneck at the AP queue. Hence the admission region turned out to be dependent almost exclusively on the streams towards the mobile stations.

Then we devised two admission control algorithms. In both cases the admission test is based on the time occupancy of the medium, but they differ in the way they compute it. As learned from previous research on the above mentioned scheduler, this metric turned out to be simple and very efficient. The first algorithm exploits an analytical model of the EDCA (Enhanced Distributed Channel Access) mode in non-saturation conditions. The use of this model, above all in the non-saturation part, allows to overcome the limits of previous works, based on saturation models. As shown in the study on the admission region, the non-saturation hypothesis closely matches the real state of the network when the interest is on time-sensitive applications. The second algorithm uses a parameter already defined, and very effective, for 802.11b systems. So, we have extended it in order to make it applicable to 802.11e networks. Differently from the first method, this scheme bases on measurements on the state of the network.

As a final remark, we would like to put some emphasis on the method we chose to evaluate our algorithms. In most cases the proposed models have been tested through simulations, and the majority of the performance tests reported in the thesis involves networks supporting voice services. In this context, the widely adopted measure of throughput, delay, and/or packet losses can only give a rough estimate of the goodness of the service. It is well known that subjective factors, like human perception of voice quality, should also be taken into account. For this reason, we have decided to take

advantage of the E-model, which is a specialised framework standardized by the ITU-T. The E-model translates the perceived speech quality into a single scalar value, which is computed not only from the measure of delay and packet losses, but also from the characteristics of the network and terminal equipment and the expectation of the average user. Therefore, for voice-oriented applications, this tool provides much more comprehensive and accurate performance measures than traditional metrics.

The thesis is organised as follows. At first, in Chapter 1, a brief overview of the IEEE 802.11 standard is given, with reference to both the basic and the enhanced (802.11e) access. A short review of the related literature is also reported, with particular attention to the topics subject of our work (i.e. performance anomaly, analytical models, admission control). Then, Chapter 2 presents the scheduler we designed to face the performance anomaly, together with simulation and experimental results. Admission control is the subject of Chapter 3, which describes the study on the admission region and the two admission control schemes. Finally, the conclusions can be found in Chapter 4.

Chapter 1

The IEEE 802.11: concepts and performance

In 1997, the Institute of Electrical and Electronics Engineers (IEEE) adopted the first wireless local area network WLAN standard, named IEEE 802.11. Initially provided with data rates of 1 and 2 Mbps, it did not receive an immediate market favour, mainly because of the difficulties in realising the interoperability between products from different vendors. Hence the IEEE created several task groups (TG) to supplement and improve these early specifications. In particular, task group *b* issued an amendment for backward compatible operations in the 2.4 GHz band with data rates up to 11 Mbps. The revised 802.11 standard [1], published in 1999 and often referred to as Wireless Fidelity (Wi-Fi), has then become an overnight success, turning the IEEE 802.11 into the *de facto* standard for wireless LANs. Further extensions of the standard followed, leading to even higher data rates: the *a* and *g* versions offer 54 Mbps in the 5 GHz and 2.4 GHz bands, and the still active task group *n* aims at reaching 108 Mbps. In parallel, other groups have been working on traffic differentiation (TGe), security (TGi), vehicle communications (TGp), mesh networking (TGs),

and so on.

As it often happens, the activities of the standardization body are being complemented by work from research centres and universities, whose typical outcome is the evaluation of the effectiveness of the mechanisms specified in the standards, as well as the proposal of further improvements. This has generated an ever increasing number of papers dealing with the 802.11 standard in all its flavours. These studies have pointed out several performance problems. Some weird behaviours and anomalies have also been reported.

At present, one of the most challenging tasks is the support of real time multimedia applications. It is by now plain that the sole use of more efficient modulation schemes and/or other frequency bands is not sufficient to offer adequate support to applications that have strict Quality of Service (QoS) requirements, such as Voice over IP (VoIP), video streaming or other delay (or bandwidth) sensitive applications. The same IEEE, aware of the unsuitability of the basic 802.11 standard, and given the increasing influence of these services, has recently ratified the *e* amendment to the standard [2]. 802.11e offers service differentiation to various classes of traffic, granting delay sensitive applications (e.g. voice and video) higher priority in accessing the channel. Though it has indeed improved the support of real-time services, yet no strict guarantee is given in terms of QoS parameters. A conspicuous number of works has already assessed the performance and proposed enhancements to this amendment.

In this Chapter a brief overview of the IEEE 802.11 standard is given. At first, the basic features and operations are described, for both the legacy version and the new *e* version. 802.11 essentially supports two modes of operation, distributed and centralised. Since the latter has received scarce attention by both researchers and, above all, manufacturers, we will focus mostly on the former. Then, a short scan of the literature dealing with

performance evaluation of both version we will be given. A particular emphasis will be put on the performance anomaly problem.

1.1 Basics of IEEE 802.11

The IEEE 802.11 standard defines two methods to access the wireless channel: the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). The former is based on CSMA/CA (Carrie Sense Multiple Access with Collision Avoidance) and let the terminals contend for the access to the medium. So it can only offer a best effort service. On the contrary the latter supplies polled access via arbitration by a Point Coordinator, which resides in the Access Point. PCF was meant to be used for applications requiring time-bounded frame delivery. In practice, however, PCF failed to deliver its promises, and has never been implemented by manufacturers. Therefore we will no longer deal with it. Describing the reasons of this failure is out of the scope of this thesis. The interested reader may refer to [1].

According to DCF rules, a station[1] wishing to transmit a frame shall sense the channel for a given period of time (called DIFS — Distributed Inter Frame Space). If the medium is sensed idle for the whole DIFS, the station is allowed to transmit. Otherwise the station shall wait until the medium becomes idle and then enter a collision avoidance phase. This phase consists in executing the exponential backoff algorithm. The station picks a random integer value from the interval [0, CW] to initialise a backoff counter. This interval is the so-called contention window. CW may vary between CW_{min}, which is the first assigned value, and CW_{max}. Then the host starts sensing the medium and, again after a DIFS, decrements the

[1] In this thesis, the terms "station" and "host" will be interchangeably used to refer to terminals equipped with an 802.11 card.

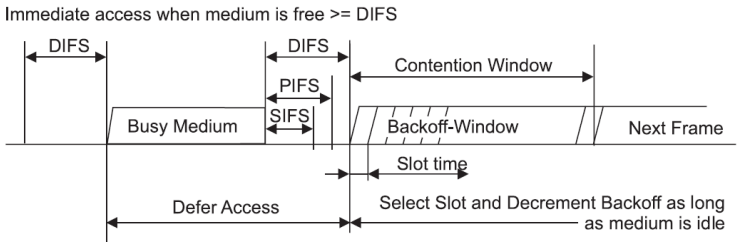


Figure 1.1: An example of 802.11 basic access

counter by one for each time unit (or SLOT) the medium is sensed idle. If the medium becomes busy the countdown must be suspended; it can be restarted only when the medium is idle again. When the counter reaches zero, the station is allowed to transmit. A graphic summary of this rules is reported in Figure 1.1.

When a frame (other than broadcast) is correctly received, the receiving host shall send an ACK frame after another fixed period of time called SIFS (Short IFS). As the name suggests, this is shorter than DIFS to allow the ACK to immediately follow the frame which it refers to. If the sending host does not receive an ACK within a specified timeout, it must assume the transmission has failed. As a consequence it must increase the retry counter for that frame. If this counter has reached the Maximum Retry Limit (MRL) threshold, the frame shall be discarded and a new transmission cycle may begin. Otherwise, the station must start a new backoff cycle with a quasi-doubled CW. The exact rule is $\mathrm{CW}_{new} = 2(\mathrm{CW}_{old} + 1) - 1$. CW is upper-bounded by CW_{max}. CW is reset to CW_{min} at the beginning of every transmission cycle, i.e. after either a success or a discarded frame due to the exceeded retry limit.

To overcome the problem of hidden terminals, the standard provides the optional RTS/CTS mechanism. The sending host shall at first transmit a short RTS (Request To Send) frame, following the same access rules

described before. When the destination hears the RTS it must respond with a CTS (Clear To Send) frame. The sender will then transmit the data frame and wait for the ACK as usual. All these frames are separated by SIFSs to prevent other stations (which shall wait for at least a DIFS) from interrupting the sequence. Moreover, each station hearing either an RTS or a CTS must set and start the so-called Network Allocation Vector (NAV). This timer is initialised with the remaining duration of the whole frame exchange, which is inserted in the control frames by the sender itself. NAV is then decremented down to zero. This mechanism, called virtual carrier sensing, lets every station know when the current transmission ends and therefore prevent them from colliding with possibly hidden stations. Since RTS are usually (much) shorter than data frames, if a collision occurs less time is wasted. However, due to the added overhead, the use of RTS/CTS is advantageous only when data frames are long (in terms of time occupancy of the medium).

Another optional, but widely implemented, feature is the possibility to use several physical data rates in the same network and to adapt these rates to channel conditions. In particular, Automatic Rate Fall-back (ARF) algorithms are employed to scale down the transmission rate to increase the probability of correct reception in case of multiple transmission failures. Amendments *b*, *a*, *g*, and the upcoming *n*, have introduced data rates up to 108 Mbps. However, to maintain backward compatibility with the first terminals, these speeds can only be used for data frames addressed to stations that explicitly support them. All other frames must be sent at one of the basic rates, i.e. 1 or 2 Mbps. We will see in Section 1.2.1 how these mechanisms, beyond raising the network capacity, are also the cause of one of the most limiting factors of 802.11 systems.

1.1.1 The *e* amendment to the standard

To introduce some level of Quality of Service (QoS), and to solve part of the coordination problems between PCF and DCF (see e.g. [3] for a brief review on this topic), the 802.11e standard defines a new function called HCF (Hybrid Coordination Function). HCF merges contention-based and controlled medium access into a single protocol. The contention method is called Enhanced Distributed Channel Access (EDCA) and provides relative QoS by differentiating the access priority to the radio channel. The other function is named HCF Controlled Channel Access (HCCA) and supports parameterised QoS through reservation of transmission time.

According to EDCA, traffic differentiation occurs exploiting four service classes, denoted as Access Categories (ACs). Every station has also four transmission queues. Thus, as opposed to DCF where all traffic shares a common queue, traffic is assigned to a specific queue on the basis of its QoS requirements. Each AC behaves like a virtual station, contending for the opportunity to transmit independently from the other (see Figure 1.2). The contention rules are the same of DCF, but diverse channel access parameters are used for each queue. If two (or more) ACs within a single station becomes ready to transmit at the same time, an internal collision occurs. The collision is resolved so that the frames with higher priority is actually sent on the channel, whereas the other AC enters into a new backoff phase, as if an external collision has really occurred (but without increasing the retry counter).

Once an AC has gained access to the medium, it is allowed to transmit more than one frame without contending again, provided that the total access time does not exceed a threshold (TXOPLimit). A SIFS is used between every frame in the burst to ensure that no other station interrupts the train. This option, called Transmission Opportunity (TXOP), has been

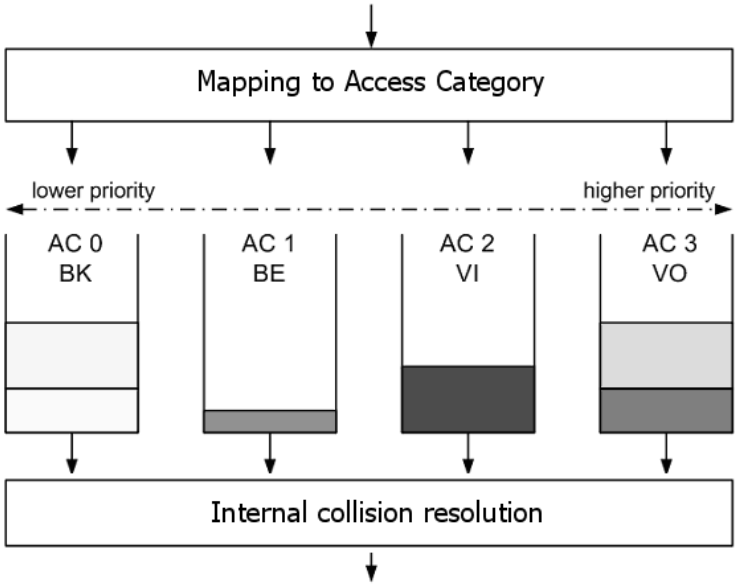


Figure 1.2: The reference implementation model of EDCA

introduced to reduce waste times and thus increase the bandwidth exploitation. Note that only the AC that gained the access to medium is entitled to transmit further frames from its queue. Hence, in the context of EDCA, station and AC are not interchangeable terms. In fact, during a TXOP, a station is not allowed to send frames belonging to ACs other than the one that won the TXOP, even though there is time left in the TXOP.

Traffic differentiation is realised through three parameters: the contention window (in terms of its bounds CW_{min} and CW_{max}), the interframe space and the TXOPLimit. The single DIFS is replaced by four specialised AIFS (Arbitration IFS). Each AIFS is equal to a SIFS plus a number AIFSN[AC] of time slots. Clearly, the higher the priority, the lower the AIFSN. Similarly, TXOPLimit is bigger for high priority ACs. Table 1.1 reports the values of the parameters for the four ACs.

Table 1.1: Default EDCA parameter set — TXOPLimit refers to DSSS physical layer

AC	Traffic	CW_{min}	CW_{max}	AIFSN	TXOPLimit
AC_BK	Background	31	1023	7	0
AC_BE	Best Effort	31	1023	3	0
AC_VI	Video	15	31	2	6.016 ms
AC_VO	Voice	7	15	2	3.264 ms

The presence of the ACs offers priority in accessing the radio channel, but it does not give any strict QoS guarantee. The actual level of achievable QoS depends on many factors, and among these the traffic load offered to the network is one of the most influencing. In particular, as the network approaches the saturation point, even the highest priority AC is unable to guarantee the required QoS. To overcome this problem, the same 802.11e standard suggests the use of admission control algorithm, whose specification is left to the manufacturers of the cards. Furthermore, to optimize the functioning of the whole system, the AP can dynamically adjust the contention window and the TXOPLimit for each AC. The new values are advertised in the beacon frames and must be immediately employed by the stations.

The other way of differentiating the service is offered by HCCA, which is designed to properly handle traffic streams with strict QoS requirements. HCCA is based on a polling scheme: the Hybrid Coordinator (HC) polls the stations according to their traffic requests. Since HCCA is not the target of this thesis, we will not deal further with it. We just mention that it seems destined to follow the unlucky faith of PCF, as in the first commercial 802.11e products HCCA is not implemented.

1.2 Performance evaluation of 802.11a/b/g

In the following, we present an outline of some of the most meaningful works in this area, summarising the achieved results. It is not our goal to give a comprehensive overview of the copious literature in this field. Rather we will focus on the topics that are most related to this thesis.

One of the first papers to evaluate the IEEE 802.11 standard through an analytical approach is [4], which provides a simple and accurate analytical model to compute the throughput of DCF. The key assumption is a constant collision probability, which is not dependent on the number of retransmissions. The behaviour of each station is modelled as a discrete-time Markov chain. By solving the chain, the author evaluates the asymptotic saturation throughput, which is shown to be dependent on the network size and on the contention window parameters. As the network size is not directly controllable, the only way to achieve optimal performance is to use adaptive techniques to tune the contention window on the basis of the estimated network size. However, when the RTS/CTS mechanism is employed, the performance is only marginally dependent on system parameters and throughput benefits even with fairly limited packet sizes.

Another interesting observation found in [4] is that 802.11, like other random access schemes, exhibits some form of instability. As the offered load increases, the throughput at first grows up to a maximum value, then shows a significant decrease. This behaviour translates into the practical impossibility to operate the scheme at its maximum throughput for a long period of time, and thus in the meaningless of the maximum throughput as a performance figure for the access scheme. Comparison with simulation results shows that Bianchi's model is pretty accurate. Remarkably, this work has been the starting point of most of the successive analytical models. Bianchi himself further refined and corrected it in [5].

The high variability of throughput and delay in 802.11 networks has been revealed by many authors. Ref. [6] is one of the first and still one of the most complete simulation papers. The effects of payload size, channel conditions, and system thresholds (for RTS/CTS and fragmentation) are considered. One of the most evident outcomes is that throughput is heavily affected by all these factors. This result is confirmed in [7], which shows that even in a simple unsaturated network, the standard deviation of throughput is around 42%, while the average and standard deviation of access delay are 1 and 1.2 seconds, respectively. Given these values, the DCF is clearly unsuitable to support QoS applications. Note that this result is easily understandable: many of the parameters regulating the access to the medium introduce random delays while the contention window itself depends on the history of previous attempts. It might also worth noting that newcomers, being their contention window at the minimum value, have higher probability to access the medium than already "collided" stations.

PCF too is shown to present some drawbacks that severely limit its support to time-bounded services [6; 8]. In a scenario with mixed data and voice traffic, even assuming rather loose delay bounds, a fair amount of voice frames must be discarded due to an exceeded delivery time. The unpredictable beacon delays and the unknown polling instant and transmission time of the polled stations are the most prominent factors of performance degradation. Furthermore, it has been pointed out that all stations in the polling list have the same priority and are polled with same rate, whether they have or not data to transmit. So, PCF is scarcely flexible with respect to different traffic requirements [9].

The authors of [10] and [11] point out that the main cause of the reduction of throughput with respect to the nominal data rate is constituted by the physical preamble and header, which are always transmitted at the lowest data rate (1 Mbps). Therefore it is particularly relevant when the

transmission of data occurs at higher data rates (e.g. 11 Mbps). Even with large packets sizes, the bandwidth utilization is very low. The values extracted from an analytical formula giving the throughput achievable by a single station generating UDP traffic are compared with the results obtained in an experimental testbed. The results confirm the formula, although in some cases the effective measured throughput is surprisingly slightly higher than the theoretical one. Furthermore, an investigation on packet loss rate as a function of distance at different transmission rates reveals that, especially when using the highest data rates, there is a significant difference in the transmission range of control and data frames. As a consequence, stations reserve the channel for a radius (much) larger than they can actually reach with data [10].

Some performance differences might be expected for the *a* and *g* amendment to the standard. Both employ the same access method of the plain 802.11 but new physical layer specifications (i.e. Coded Orthogonal Frequency Division Multiplexing, COFDM); 802.11a also operates in a different frequency band. A comparison between these two version is provided by [12] for an indoor scenario. The authors translate a ray-launching propagation model into achievable data rates and hence throughput performance. The result is that 802.11g achieves superior coverage (around 10 percent) thanks to lower signal attenuation at 2.4 GHz, but much lower data rates caused by MAC inefficiency when maintaining backward compatibility with 802.11b.

Summarising, DCF is well suited to support data traffic under light load and with a limited number of stations. That is mainly caused by two factors: first, collisions and random backoff not only constrain the throughput to a limited quota of the nominal bandwidth, but also decrease network capacity as the number of stations increases. Moreover, given the wide fluctuations of throughput and delay, all traffic different from

asynchronous data transfer will be severely impaired by DCF. On the other hand, PCF may be adopted for the delivery of time bounded services only when the requirements, in terms of bandwidth, delay and jitter, are not very stringent. As a matter of fact, the presence of a conspicuous number of terminals with data traffic and different access needs and capacities may seriously abate the PCF effectiveness.

1.2.1 The performance anomaly

A potentially deleterious behaviour has been revealed by Heusse et al. in [13]. The authors observe that when some mobile hosts use a bit rate lower than the others, the performance of all hosts is considerably degraded. This situation is common in wireless local area networks in which a host far from the Access Point is subject to important signal attenuation and interference. To cope with this problem, the host usually down-scales its bit rate to some lower value to exploit more robust modulation schemes. But the same phenomenon may also occur when stations are equipped with cards supporting different data rates (e.g. 802.11b and 802.11g).

The authors derive simple expressions for the useful throughput and validate them by means of simulation. They conclude that the basic CSMA/CA channel access method is at the root of this anomaly: it guarantees an equal long term channel access probability to all hosts, thus penalising fast hosts and advantaging slow ones. For example, a host transmitting at 1 Mbps reduces the throughput of all other hosts transmitting at 11 Mbps to a value below 1 Mbps. The network presents a clear "performance anomaly": the throughput of all hosts transmitting at the higher rate decays even below the level of the lowest rate. This anomaly holds whatever is the proportion of slow hosts. The question becomes important for hot spots that cover

areas with a great number of hosts, when the probability that some mobile hosts suffer poor channel conditions and/or use a lower bit rate is high.

As a final remark, it is worth noting that the performance anomaly, is a direct consequence of the philosophy backing the 802.11 standard. The CSMA/CA medium access strategy coupled with the fairness objective of DCF (*max-min* throughput fairness) is exactly the cause of this drawback. Hence the anomaly affects the whole 802.11 family, and, in particular, it also affects the recently ratified *e* amendment. So, real-time services too, even when transported over 802.11e networks, are subject to be heavily spoiled by this phenomenon.

1.3 Performance evaluation of 802.11e

Performance evaluation of 802.11e EDCA has been thoroughly carried out in recent literature[1]. An analytical model to evaluate saturation throughput, channel access delay and frame dropping probability for EDCA was proposed by Xiao in in [14]. Though the paper refers to an early draft of the standard, the model actually served as a starting point for many successive works. The author extends Bianchi's model to the four ACs of the *e* amendment, accounting for the different AC access parameters. The solution of the discrete-time Markov chain is then compared to simulation results for an 802.11a physical layer. The paper shows that saturation delay is very sensitive to the minimum CW, while the AIFSs provide faster/slower access to channel, but do not reduce collisions. Therefore using a backoff-based metric, which has the function of reducing collisions and providing priorities, is a better choice, in terms of total throughput and delay, than

[1]Most of the works actually refer to previous drafts of the standard, thus dealing with the precursor of EDCA, called EDCF (Enhanced Distributed Coordination Function).

differentiating the inter-frame space. However, differentiating the inter-frame space offers an easy way to favour the classes with a short AIFS. As for traffic with sensitive delay requirements, a smaller retry limit is appropriate, whereas non-real-time traffic may need a larger retry limit to enhance reliable transmissions.

Among the many extension and additions to Xiao's work, Engelstad and Østerbø's is particularly interesting as it also accounts for non-saturation conditions [15]. The model is used to derive throughput, delay and frame dropping probabilities in the whole range from a lightly loaded, non-saturated channel to a heavily congested, saturated medium. It analyses the differentiation based on all the adjustable parameters (i.e. window-sizes, retransmission limits, TXOP lengths, and AIFS values). Through the presented model, the authors also provide an approximate expression to determine the starvation point of the different ACs. In particular, by measuring the channel load and by knowing the AIFSN assigned to each AC, the access point is able to tell when the starvation conditions are present for any of the ACs, independent of whether packets of these ACs are attempted for transmission. A very detailed description of this model is reported in Section 3.2.2.

Performance assessment of 802.11e is performed via simulations in [16]. Three different types of traffic are considered (voice, video, and data), with each station generating only a single type of traffic. The authors also consider the use of multiple frame transmissions during a single TXOP. From the observed delay and error performance (very low voice and video frame losses are recorded), the authors conclude that EDCA can support real-time applications with voice and video traffic with a reasonable quality of service in certain environments. Furthermore exploiting the TXOP is found to increase the overall system throughput and achieve more acceptable streaming quality in terms of frame losses and delays. Finally the

authors remark that EDCA could be optimized by adapting the parameters at run-time, depending on network load and supported applications, and, for acceptable QoS provisioning, there should be an admission control process in place. These same results are confirmed by many authors, e.g. He and Shen [17], who also highlighted that, under heavy load conditions, low priority traffic goes into starvation. Moreover, in a centralized scenario, the downlink has far worse performance than the uplink. This is because all the down link traffic, which is supposed to be N times higher than the uplink, share the channel with all the stations, receiving only a small fraction of the bandwidth (also see Section 2.3.2.3). In this case too, the authors suggest employing some form of access control or scheduling.

A detailed study on the effects of the different AIFSs and on the coexistence of EDCA with legacy DCF-based stations is given in [18]. Rather than focusing on high-level performance figures (e.g. throughput and delay), the authors look at the details of EDCA operations in terms of low-level performance metrics (e.g., the probability of accessing specific channel slots). This investigation reveals that AIFS differentiation provides superior and more robust operation than contention window differentiation. It does not trade off service differentiation with aggregate throughput impairment, it is natively adaptive to network congestion, and even a single slot difference may result in a substantial difference in terms of performance. As for the coexistence between the two versions of the standard, the authors show that the different mechanisms for backoff counter decrement used in EDCA allow gaining, in practice, one extra slot to be used for AIFS differentiation. Setting the EDCA AIFS equal to the DCF DIFS (this is the minimum possible setting for the AIFS value), EDCA traffic experiences substantially higher access priority. Hence AIFS differentiation is effectively deployable in an hybrid EDCA/DCF scenario.

In summary, from the work so far, it appears that the distributed access method (EDCA) provides relative QoS differentiation among traffic classes but it does not provide any QoS "guarantee". In other words, a traffic contract for a connection is only an objective that the wireless network will only attempt to honour as often as possible. EDCA is relatively simple but the performance it provides is obviously less predictable than a reservation-based method and suffers from network congestion. Moreover it is intrinsically unfair, as low priority traffic can easily go into starvation. Therefore, to move from service differentiation to provision of QoS objectives, it is necessary either to switch to a centralized form of channel access control, namely HCCA, or to enhance the EDCA operation with additional admission control mechanisms. This latter issue is the object of current research work, and, in fact, of this thesis.

Chapter 2

The Deficit Transmission Time scheduler

The success of the 802.11 standard has encouraged the IEEE to spend much effort in improving both the raw throughput and the support of more appealing services such as voice and video. Nevertheless, one of the most critical factors driving the efficiency of such systems still remains the ability of the terminals to overcome the hurdles imposed by the wireless channel. As explained in Section 1.2.1, one of the most hampering phenomena is the so-called "performance anomaly". This behaviour, coupled with the simple "First-In First-Out" (FIFO) strategy employed at the Access Point (AP), causes a severe degradation of the system performance.

An optimal solution to overcome the performance anomaly is reckoned to reside in a distributed scheduling algorithm, based on a coordination among all the stations including the AP. Since this is not a trivial task, we can at first approach the problem in a centralized way, focusing only on the scheduling discipline at the AP. This will obviously lead to a sub-optimal solution, but also to a noticeable simplification of the problem. The fog on the correctness of such a simplification can be cleared with the assumption

of limited contention in case of traffic asymmetry. This is truer and truer as the unbalancing between downlink and uplink grows (in favour of the downlink direction), as it happens for example for data traffic generated by web browsing, email, and so on, which are, at present, the vast majority of the applications run over today's WLANs.

Several schedulers have already been proposed in literature (see e.g. [19]). Most of them rely on a model of the wireless channel: the links between the base station and the user devices are independent of each other and are subject to bursty errors. Markov models are often used to imitate the quality of the link (a comparison among the different models is presented in [20]). However, these models could be sometimes distant from the actual channel behaviour, and a system based on it could become inefficient. A more reliable solution would be centring scheduling decisions on a real measure of the channel.

Starting from these observations, we propose a scheduling algorithm that accounts for the actual state of the channel. The quality of the links is quantified as the amount of time spent for the transmissions of the frames. Using this approach, we can adopt the time the channel is in use as the resource to share between the stations (in place of the total capacity of the channel used by plain 802.11). The criterion for this sharing can also be changed. In an infrastructured WLAN, proportional fairness in bandwidth (or equivalently max-min fairness in air-time usage) allows a reasonable trade-off between efficiency and fairness, and leads to the very desirable property of flow isolation. If the transmissions are either in the uplink or downlink direction only, proportional fairness is equivalent to air-time usage fairness [21].

The next Section reports a brief overview of the fairness concept and its application to wireless LANs. Afterwards we describe the architecture and the algorithm of the proposed scheduler, together with some experimental

trials and some simulations in a voice over IP context. Then, in Section 2.3, we extend the scheduler to a distributed version, in order to approach the optimum solution.

2.1 A different vision of fairness

In order to properly design a scheduler, it is necessary to define an objective to be maximized. In wired networks, the design of traffic control algorithms has been carried out optimizing two parameters: fairness among different flows and efficiency in link utilization. In wireless networks, and particularly in 802.11 WLANs, these two parameters are somewhat conflicting. Recent experimental analyses (e.g. [22]) have shown that in multirate wireless networks throughput fairness leads to bandwidth underutilization. This is further confirmed by [13], in which the authors prove analytically that the performance of an 802.11 network is determined by the stations using the lowest data rate.

In this scenario, a fundamental choice is whether we should strive to maximize throughput fairness (i.e. achieve "max-min fairness"), maximize the total throughput (achieve the best "efficiency"), or strike a balance among the two. Typically, achieving one of these goals is directly related to the maximization of a particular "utility" metric. A widely accepted general expression for such a metric is $\sum_j U(x_j)$, where x_j is the rate (throughput) of flow j and $U(\cdot)$ is a concave function called the "utility function". The characteristics of the utility function $U(\cdot)$ affect the properties of the utility metric, and consequently the particular fairness objective that is pursued. The most used class of utility functions is the one proposed in [23], which is described by the following expression:

$$U(x,\alpha) = \begin{cases} \frac{x^{1-\alpha}}{(1-\alpha)}, & \text{if } \alpha \neq 1 \\ \log(x), & \text{if } \alpha = 1 \end{cases} \tag{2.1}$$

In this equation, α is a parameter that can be modified to tune the trade-off between efficiency and fairness. Some utility functions are plotted in Figure 2.1 for typical values of α. In particular, when $\alpha = 0$ we should maximize $U(x) = x$, so the utility function leads to the extreme goal of throughput maximization, at the complete expenses of fairness. A scheduler that pursues this goal would allocate the medium to the station with the highest data rate, whereas low data-rate stations would starve. In contrast, when $\alpha \to \infty$, the utility function leads to extreme fairness, or max-min fairness in bandwidth. The 802.11 MAC implicitly adopts this function, achieving a long term fairness in terms of channel access probability among all the competing stations. However, as previously explained, this is may not be efficient, as the throughput of all stations tends to be aligned to that of the slowest terminal.

In most cases, including the one we are studying, the two above mentioned behaviours are not suitable, and a different choice for α must be found. One of the results of the study presented in [24] is that a reasonable trade-off between efficiency and fairness can be obtained by setting $\alpha = 1$, which corresponds to the concept of proportional fairness. In mathematical terms, this translates into maximizing $\sum_j \log(x_j)$, or equivalently $\prod_j x_j$.

An interesting property of this criterion was proven in [21]. The authors demonstrate that in an infrastructured WLAN, given a fixed number of stations, the throughput of any of them is independent of the data rates used by the others if proportional fairness in bandwidth is achieved (flow isolation). In practice, proportional fairness can be realized imposing that the air-time usage shares of every station (accounting for both uplink and

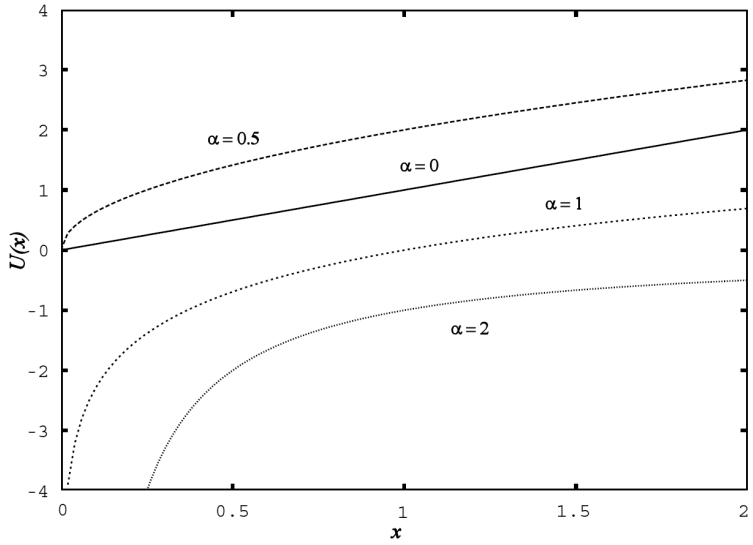


Figure 2.1: Utility functions for different values of α

downlink transmissions) are equal. Proportional fairness in bandwidth is thus equivalent to max-min fairness in air-time usage.

2.2 The Deficit Transmission Time scheduler

The considerations expressed in the previous Section led us to design a scheduling algorithm whose goal is to achieve proportional fairness. The scheduler is going to be implemented at the AP and will operate according to a centralized policy, delivering fair air-time usage only to the flows addressed to the associated wireless stations. As discussed at the beginning of the Chapter, this behaviour is backed by the assumption that the volume ratio of the offered traffic is biased towards the downlink direction.

2.2.1 Description of DTT

The architecture of the proposed scheduler is illustrated in Figure 2.2. The whole framework is inserted above the MAC layer, which is in no way modified. A classifier splits outgoing traffic into several queues according to some predefined rule, which currently is the destination MAC address. Yet, this can be easily extended to support other rules, e.g. different user priorities, as in 802.1Q and 802.11e. A "bucket" is associated to each queue to account for the time the queued frames have spent on air during the previous transmissions. Air time is thus the "water" (or tokens) used to fill/drain the buckets.

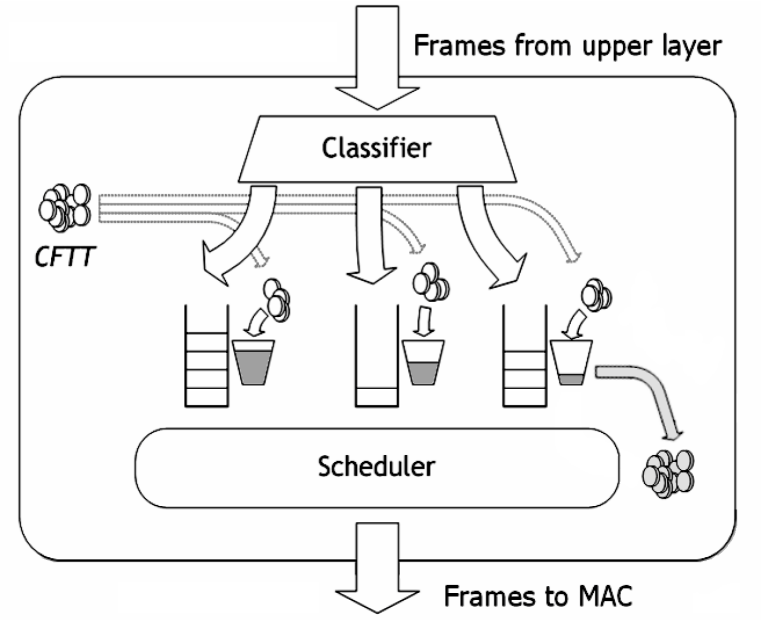


Figure 2.2: Architecture of the DTT scheduler

At the end of every frame transmission cycle, the scheduler computes the Cumulative Frame Transmission Time (CFTT). The CFTT computation starts when the frame has reached the head of the transmission queue

at MAC level and comprises the whole time spent to deliver that frame, including all retransmission attempts, backoff and idle periods. The CFTT is also produced when the retry limit is reached (i.e. in case of transmission failure). The CFTT is then used to drain the bucket associated to the destination of the transmitted frame. Next, this same value is equally divided by the number of non-empty queues and the result loads the related buckets. The bucket whose frame has just been sent, if non-empty, is included in this count. This is needed to grant all the queues the same transmission possibility. All the buckets connected to queues that have been empty for a given inactivity timeout are cleared (set to zero). This is necessary to avoid that these queues, after having been idle for a long period of time, keep a credit/debit that would reduce short-term fairness once they have some new frames to transmit (e.g. they could have enough water in the bucket to monopolize the access to medium for a while).

Once these tasks have been completed, the scheduler picks the next frame to be transmitted from the queue whose associated bucket is the fullest. If more buckets are at the same level, the scheduler chooses randomly among them. This frame is then passed to the MAC layer, which provides for the physical delivery. Note that all frames are stored at scheduler level, and only one frame at a time is sent to MAC. This avoids the MAC buffer to hold any other frame but the one under delivery and allows the scheduler to make a precise computation of the CFTT and to have a tight control on the access to medium.

Let us illustrate the behaviour of the scheduler with an example (refer again to Figure 2.2). Let us consider a network with three users. The scheduler will therefore create a queue and a bucket for each associated station: left, centre, right. Let us assume that the MAC layer has just completed a transmission of a frame for the station connected to the queue on the right. Then CFTT tokens are drained from the rightmost bucket,

and, since all queues are non empty, the CFTT is divided by three and each bucket is added CFTT/3 tokens. The one on the left is now the fullest bucket, hence the scheduler will pick a frame from the left queue.

2.2.2 An insight into DTT features

To fully understand the advantages of the proposed DTT scheduler, it is necessary to develop some considerations about its main features.

The water that fills the buckets is not related to transmission times with complex formulae, but with a simple and direct one-to-one relationship. Thus it is exactly a transmission time, or a fraction of that, and the CFTT, in opposition to most channel models, is a deterministic measure (not an estimate, nor a prediction) of the link state. More retransmissions, possibly at lower bit rates if automatic rate fall-back (ARF) algorithms are in use, are carried out in the attempt to deliver the frame to stations whose link quality is poor. An example is reported in Figure 2.3. It refers to a successful frame transmission composed of two unsuccessful and one successful events (the third). Note that, although the DIFS and backoff periods are not strictly transmission times (as far as the radio is concerned, they are idle periods), they have been included in the evaluation since they do limit the maximum throughput.

The direct consequence of the CFTT computation and distribution method is that stations that are difficult to reach, or using low bit rates, get their buckets emptied by more water, thus having to wait longer before being chosen for the next transmission. On the contrary, easily reachable, or high data rate, terminals get their buckets lightly drained, and so they will be likely to wait for shorter intervals. Under these rules, it becomes clear that the fullest bucket is also the one whose associated queue has occupied the medium for less time, and hence that should be served next to

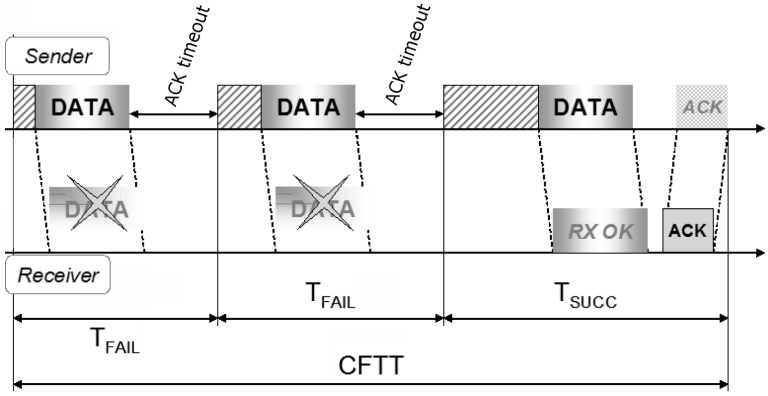


Figure 2.3: Example of CFTT computation

achieve the long term fairness in air time usage. The name of the scheduler, Deficit Transmission Time (DTT), aims at reminding just this concept.

Some other advantages of DTT are the following. Since the scheduling metric is a simple time measure, it does not need any calibration. Then, the scheme is conservative, in the sense that the water in the buckets (that might also be negative) does not diverge, quite the reverse it always tends to zero. Finally, by introducing some weights when distributing the water, traffic and stations can be easily differentiated on the basis of various parameters, such as privileged users, IP Traffic Classes (if such knowledge is available) or 802.1Q VLAN Priority tags.

Also note that only one frame at a time is sent to MAC and only after the previous frame has been transmitted (or dropped). This avoids the MAC buffer to hold any other frame but the one under delivery and therefore prevents the card from sending packets when it is not its turn. In other words, this allows the scheduler to have the tight control on the access to medium that is necessary to let it work properly.

2.2.3 Prototype implementation

The field of existence of our scheduler was not limited to theoretical design and simulation tests, as it often happens for this kind of proposals, but it was extended to a prototype of a DTT-based Access Point. This allowed us to achieve two important goals. First, we had an experimental verification and measure of the effectiveness of DTT. Second and most important, we realised the first working implementation of an algorithm that puts proportional fairness into practice.

2.2.3.1 Description of the prototype AP

To embed the DTT scheduler in a customisable AP, we have developed a software framework. This is realised as a set of Linux kernel modules, tightly integrated with the Host AP driver for devices based on Intersil's Prism2.5 chipset [25]. The most relevant components of the framework are shown in Figure 2.4. This architecture is integrated in the standard protocol stack of Linux systems to allow an easy and straightforward implementation without the need of custom-made MAC controllers. It also allows a simple software upgrade, needing no expensive hardware replacement. Still, we foresee that in the future the framework could be integrated in programmable MAC controllers; as a side effect, this would additionally reduce the complexity of the scheduling architecture.

In simple Linux-based APs, the device driver encapsulates the packets addressed to wireless stations into frames and passes them directly to the device, where they are queued in a hardware buffer waiting for their transmission turn. In our framework, the scheduler is inserted as a kernel module at device driver level (the Scheduler Module, which comprises all the objects of Figure 2.3). It intercepts the frames that the Device Driver

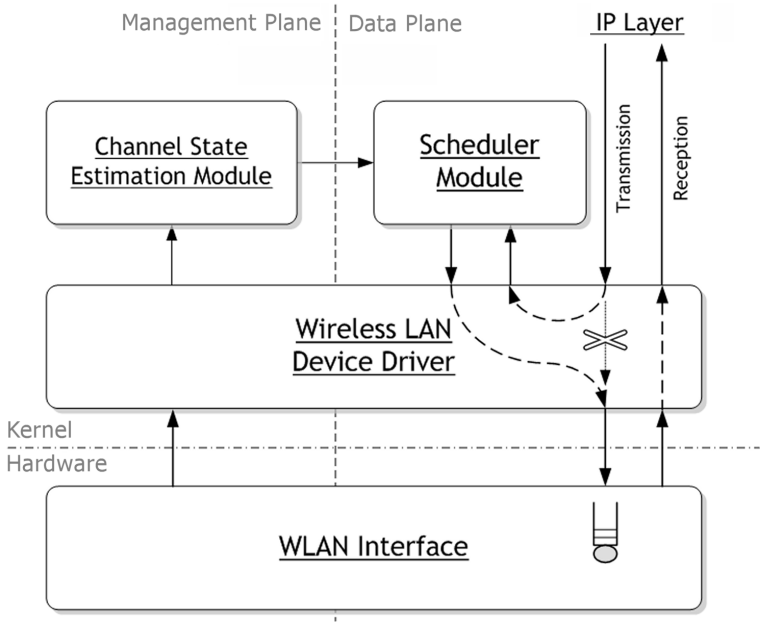


Figure 2.4: Overall architecture of the framework

is sending down to MAC and stores them internally. It dynamically creates and manages as many queues and buckets as the number of registered stations. Then, after performing the operations described in Section 2.2, it re-inserts a frame at a time into the transmission chain, letting the Device Driver deliver it to the MAC interface, which dispatches it over the medium according to the standard IEEE 802.11 rules. The Channel State Estimation Module (CSEM) is hooked to the Device Driver and is in charge of computing the CFTT and communicating it to the Scheduler Module. The Device Driver has been modified to notify the CSEM of completed frame transmissions and reached retry limits. The reception path is not touched.

As stated in Section 2.2, to let the scheduler work perfectly, all frames should be stored in the Scheduler Module, keeping the hardware queue empty, and fetching a single frame from the host memory only when the

previous frame has been definitely dequeued. However, due to current hardware limitations, it is not convenient to have only one frame in the hardware buffer. The time to transfer a frame from the host memory to the device is much higher than the average idle time between two consecutive transmissions (DIFS plus backoff). This will introduce an unnecessary delay that severely reduces the maximum achievable throughput. Hence we have implemented a mechanism that manages to keep (at most) two frames in the hardware queue. One of them is the frame currently served, the other waits in the queue. This strategy obviously brings a little deviation from the ideal, which however we found to be negligible (as shown in the following). Further details can be found in [26].

The CFTT measured by CSEM is exact in the hypothesis that the AP is the sole active transmitter in the network. If another station performs a frame transmission between two transmission attempts (related to the same frame) from the AP, the CFTT will incorporate that duration too. In principle, this time should be subtracted from the CFTT. Practically, these events can be considered to occur at random intervals and consequently influence the frames of all the queues in the same way. Moreover, given the traffic unbalancing (the greater part is in the downlink direction), these events are rather limited. Therefore, to keep CSEM simple, the current implementation does not perform such an operation. We will show that this simplification does not actually affect the performance in a noticeable way.

2.2.3.2 The experimental testbed

The experimental trials are performed over a testbed of one AP and a variable number of associated stations (see Figure 2.5). The scheduler has been installed on the AccessCube, which acts as the prototype AP. This is a compact hardware platform dedicated to Wireless LAN mesh routing. It is

based on a 400 MHz MIPS processor running a compact Linux distribution (for technical specifications see [27]). A commercial AP, the HP Procurve 420, was used as the baseline for the FIFO discipline. The stations are simple laptop PCs. All terminals are equipped with 802.11b cards, except for the commercial AP, which is b/g capable; we have therefore configured it to support only the 802.11b standard. All cards run their vendor specific ARF algorithm. The RTS/CTS mechanism is always disabled.

The AP is connected through a wired link to a server that generates UDP and TCP traffic addressed to the mobile stations. UDP packets are created using the MGEN traffic generator [28]; TCP traffic is obtained via an FTP file transfer session. Traffic is mostly in the downlink direction, from the AP to the stations. The only exceptions are the TCP control packets sent back from the stations to the server. To create various and varying channel conditions, the stations change their position, moving closer and farther from the AP, until they exit its radio coverage range, and thus are disassociated. The timeout value to force disassociation of poorly connected stations has been lowered, in the Host AP driver, from 300 to 30 seconds. This is an optimization that aims at quickly releasing the bandwidth of stations that have clearly become unreachable. It does not affect the scheduler performance, given that all its operations are carried out in a much finer time scale (one second or less). The experimental trials were run in a typical office environment.

We compared our scheduler with the simple FIFO discipline and with another solution to the performance anomaly proposed by Portoles et al. in [22]. This solution too accounts for real link conditions, but instead of scheduling frames it just tries to control the rate of the downlink flows by setting a limit that specifies, for each destination, the maximum number of packets enqueued at the AP. We will refer to this traffic shaping scheme with the acronym "PZC" from the authors' names.

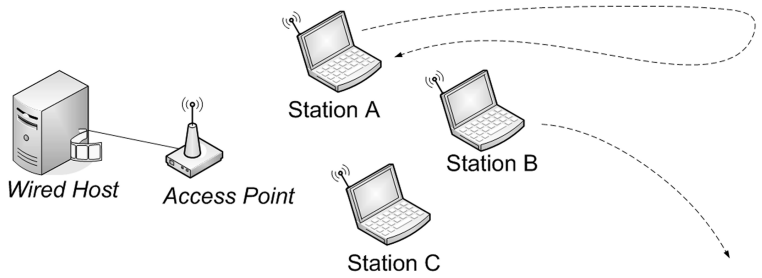


Figure 2.5: Topology of the experimental testbed

2.2.3.3 Analysis of the results

In a first try, the server generates two 5 Mbps UDP flows, with an IP packet length of 1500 bytes, addressed to two wireless stations. The system is clearly saturated, as the offered traffic is higher than what the 802.11b network can serve in ideal conditions (about 6.2 Mbps). At the beginning of the try both stations enjoy a very good link to the AP. After some time, station A moves away from the AP, and the quality of the link starts degrading until it is disassociated. After a while, the station gradually returns to its starting position, thus re-associating and improving its channel conditions. During this period, the other station (B) is constantly kept in optimal radio visibility with the AP.

Throughput values for both stations are reported in Figures 2.6, 2.7 and 2.8. When both stations are in a good position, each one receives about 3.1 Mbps, which is half of the available throughput. So, the two flows evenly share the available bandwidth, independently of the AP scheduling policy. When station A starts moving, the differences among the different schedulers become noticeable. Measurements with the commercial AP (see Figure 2.6) clearly reveal the anomaly of the 802.11: station B, which still enjoys a good link, is dragged into bandwidth shortage when the link of station A degrades.

The enhancements introduced by the other two schemes are likewise evident. As for the PZC scheme (see Figure 2.7), it reacts allocating more and more bandwidth to the closer station, and reducing the number of queued frames addressed to station A. However, it cannot avoid a non negligible transient period. Since all the flows share the same queue, and since PZC, following to previous successful transmissions, raised the number of enqueued frames to the maximum, the AP must get through a considerable backlog of frames before handling the new situation. Additionally, as soon as a frame addressed to station A is correctly transmitted, further frames are allowed to be enqueued. Some improvement is therefore noticeable only when link quality of station A is severely reduced (after about 45 seconds), as very few (one or two) enqueued frames belong to its flow, causing just a limited impairment to the transmissions towards station B. Summarising, the maximum throughput that B can experience depends on the maximum and minimum number of frames that PZC allows to be enqueued between two transmissions to A. The higher this value, the higher the throughput, but also the higher the reaction time.

Finally, with regard to the DTT scheduler (see Figure 2.8), station B keeps receiving its data flows at roughly 3.1 Mbps almost irrespective of the other station's position. The oscillations are due to the attempts by the AP to deliver the frames to the far terminal and to small reaction delay due to the presence of two frames in the hardware queue. Given the poor link quality, most frames addressed to A reach the retransmission limit, thus occupying the channel for the longer time possible (this includes ARF policies). As soon as station A is disassociated, station B acquires the full control of the channel. The 6 Mbps peak, present in some of the following graphs as well, is due to the packets backlogged in the AP transmission queue. In conclusion, the main response of this try is that, although the

efficiency of PZC may sporadically exceeds that of DTT, only DTT is able to completely isolate the flows and hence achieve proportional fairness.

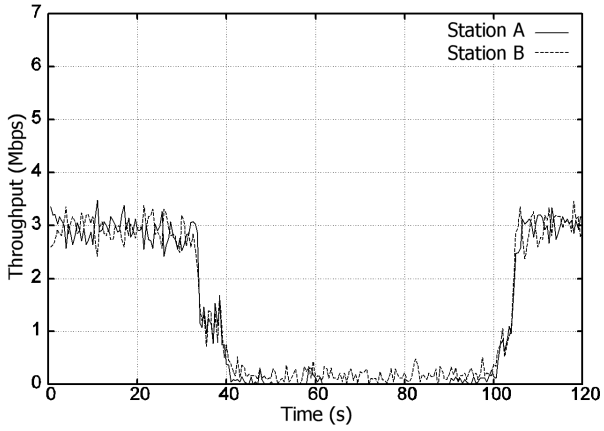


Figure 2.6: Throughput for two flows for the standard FIFO AP

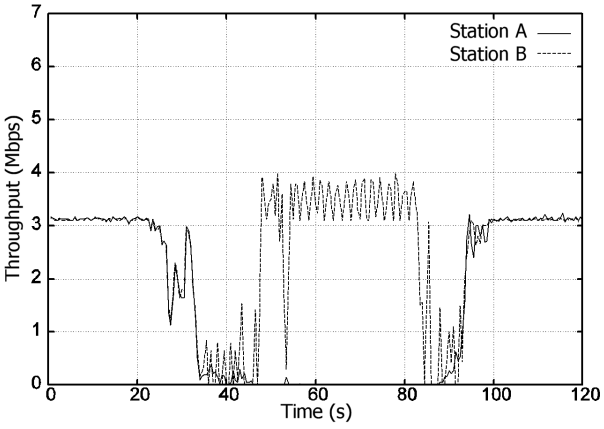


Figure 2.7: Throughput for two flows for the PZC-based AP

To increase the confidence in the achieved performance, some more tests were performed. The set of possible patterns of number of stations, movements, timings and traffic loads to choose from is extremely vast. As ref-

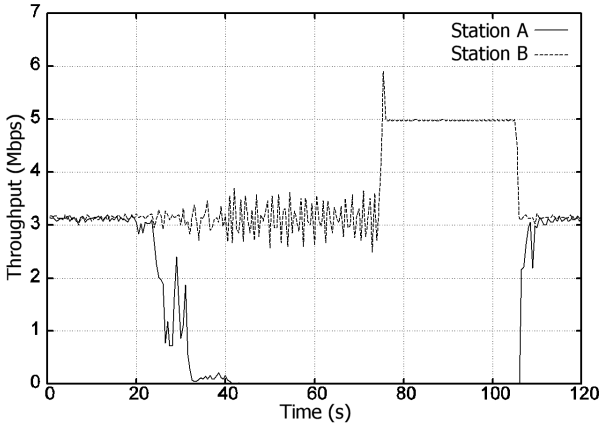


Figure 2.8: Throughput for two flows for our DTT-based AP

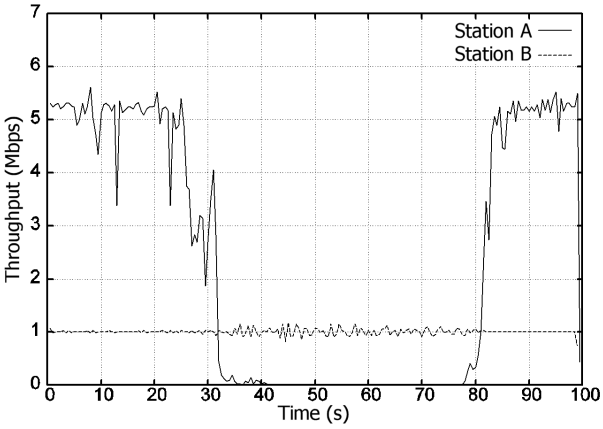
erence tests, we report an example of traffic asymmetry, a very simple scalability test, and a test with mixed UDP and TCP traffic. Through these tests we will show that the principle of operation of the DTT scheduler, and the corresponding implementation in the Linux kernel, making no assumption on the number and position of the associated stations, nor on the amount and kind of traffic, is general enough to be applied to a wide variety of scenarios.

In the same context of the previous experiment, we raised the flow addressed to station A to 6 Mbps, whereas the other was lowered to just 1 Mbps. We have alternately moved both stations. The results are reported in Figure 2.9 (we omit both FIFO and PZC, since we are only interested in a thorough evaluation of DTT). When both stations are in a good position, station A gets 5.2 Mbps, that is the sum of its fair share of the available capacity (3.1 Mbps) plus the 2.1 Mbps that are not used by station B. We say that DTT allows some "spare bandwidth borrowing": if some station has no frames addressed to it, its unused air-time is distributed to the stations with non-empty queues, according to the work-conserving principle

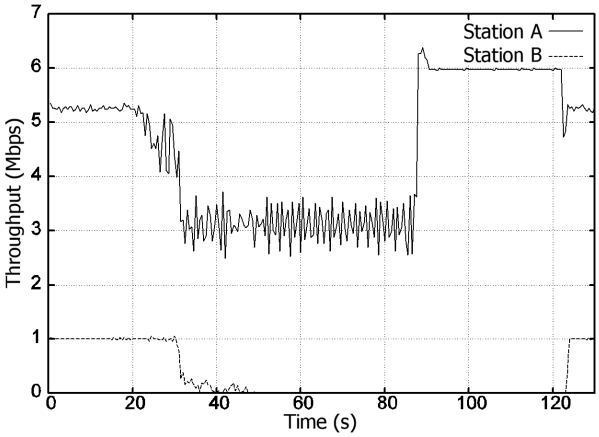
presented in Section 2.2. The notches in the first part of the plot are due to environmental interferences. When station A moves away from the AP its throughput decreases, but the flow towards station B is unaffected, as a consequence of the flow isolation property of the DTT scheduler (see Figure 2.9(a)). On the contrary, when B moves away (see Figure 2.9(b)), less and less spare air-time (i.e. bandwidth) is left for transmissions to station A. The air-time previously borrowed to A is taken back in the effort to sustain the 1 Mbps flow addressed to station B. The data rate perceived by station A drops to 3.1 Mbps, which however is nothing less than its fair air-time share. When B is finally disassociated (at around 86 s), station A can enjoy the whole channel.

The next example was used to test DTT in a scenario with more complex features than the previous cases. There are three stations (A, B, and C), at first all close to the AP. Then station A starts moving until it reaches a position in which its link becomes weaker, but is never cut (it may now represent a user standing in an unfavourable place). Later on, station B departs from the AP, and keeps on moving until it is disassociated. Towards the end of the test, B re-enters the AP coverage area. Station C always has a good connection to the AP. The behaviour of DTT is reported in Figure 2.10.

As expected, when station A starts moving, the scheduler manages to make this event unnoticeable to the other stations, which perceives no significant changes in the received throughput. A similar behaviour has been observed also when B moves. At last, when B is disassociated (at 125 s), both A and C can get more channel capacity. Station A only improves slightly, being still subject to poor channel conditions, but C is able to exploit this increase at its best, reaching 3.1 Mbps. Finally, when B re-enters the network, it can have its air-time share back, which is equally



(a) Station A moves away



(b) Station B moves away

Figure 2.9: Throughput for two asymmetrical flows under the DTT scheduler

subtracted from those of A and C. In conclusion, apart from some short-term oscillations, the three flows are still isolated.

Up to now, all the measurements have been carried out with downlink UDP traffic only. Now we introduce in the system some TCP traffic. For

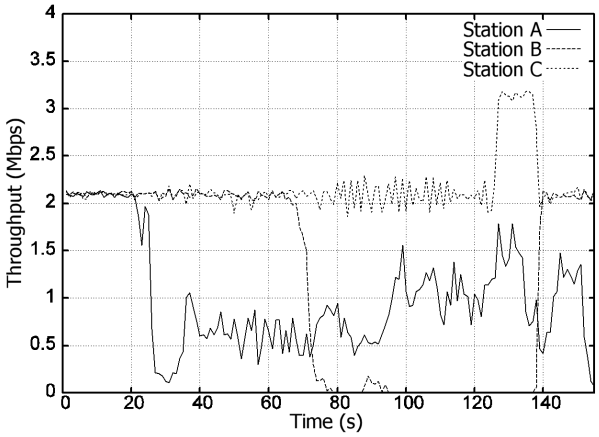


Figure 2.10: Throughput for three stations with the DTT scheduler

the sake of simplicity, we present two experiments in which one station is the destination of a downlink UDP flow with a data rate of 5 Mbps while the other is the endpoint of a downlink TCP session (see Figures 2.11 and 2.12). We recall that the station receiving TCP downlink traffic is requested to send (TCP) acknowledgement packets, thus subtracting a small fraction of bandwidth to the other flows. The performance of DTT is compared to PZC. The commercial AP did not yield any result, as the TCP session could hardly begin when injecting the 5 Mbps flow. When the unique AP queue gets saturated, TCP starts the congestion control procedures, thus reducing its throughput. UDP however keeps its packet rate constant and quickly monopolizes the queue, eventually causing TCP to starve.

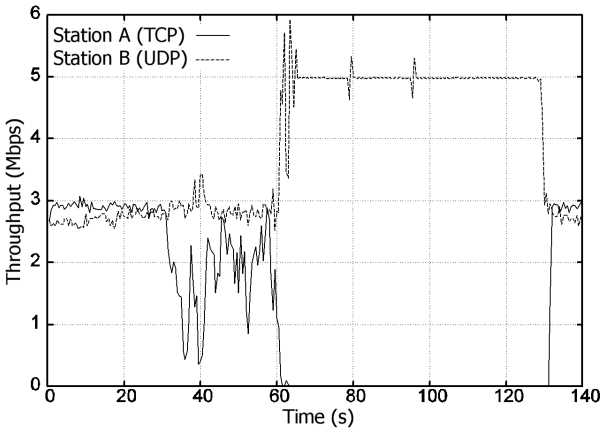
In a first experiment the TCP station is moved away from the AP. Note that, when the PZC scheme is employed (see Figure 2.11(b)), the TCP flow is unfavoured since the beginning, and never gets close to its maximum theoretical data rate (3 Mbps), as it happens when the DTT scheduler is running. Then, when the TCP station starts moving away, DTT can grant it some degree of fairness, whereas the PZC immediately

shows better regard for the closer station. Also note that, due to the congestion control mechanism of TCP, some idle periods of transmission from the queue related to the TCP station are used by the other queue, causing the small spikes visible (at around 40 s) in Figure 2.11(a). When the TCP station is disassociated, both DTT and PZC let the whole bandwidth be captured by the UDP flow. When the TCP station re-associates, it acquires back its original share of air-time (and throughput).

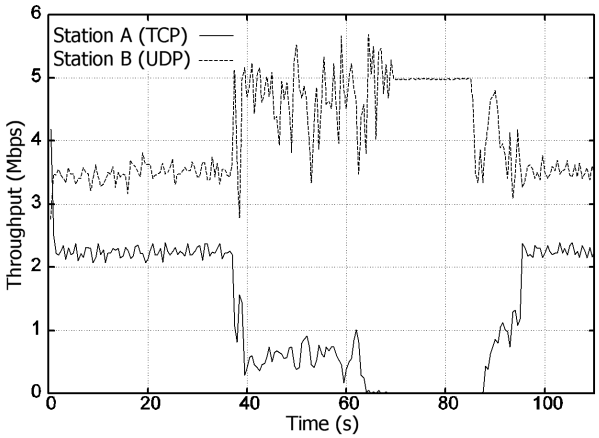
In a second experiment the roles are inverted and the UDP sink station moves away. When DTT is running (see Figure 2.12), the TCP session is almost completely unaffected by the movement of the other station, and continues transferring bits at a data rate very close to the limit. Conversely, PZC shows once again that it is not able to refrain the two flows from influencing each other. This is a further proof of the good flow isolation properties of DTT.

2.2.4 DTT for VoIP applications

In this Section we further analyse the behaviour of DTT focusing on bidirectional Voice over IP (VoIP) traffic. In this context, the simple throughput, delay, and/or packet losses are not able to offer a thorough indication of the goodness of the service. Subjective factors, like human perception of voice quality, should also be taken into account. Hence we have decided to carry out a series of simulative tests within the framework defined by the E-model. This approach has already been proved to be practical by Coupechoux et al. [29], who studied the VoIP capacity of an 802.11b network operating in the DCF mode. In their work they showed that the capacity of the network is highly dependent on the position of the terminals, thus proving the effectiveness of the E-model in revealing the performance anomaly. In our study we adopt a similar approach to demonstrate that the scheduler



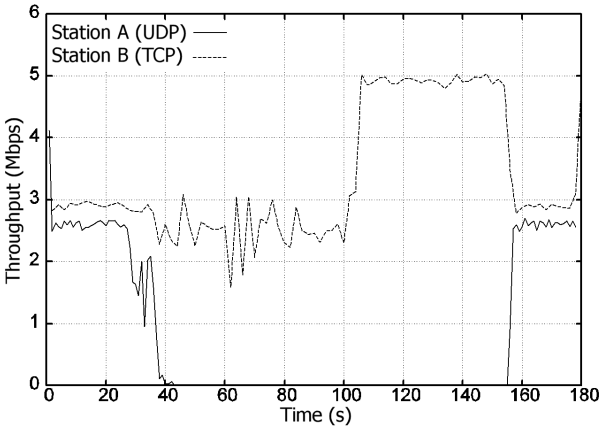
(a) DTT



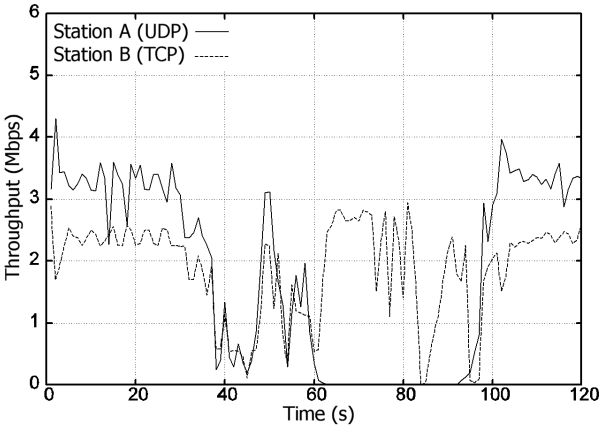
(b) PZC

Figure 2.11: Throughput for mixed UDP and TCP traffic — TCP station moves away

we propose is successful in mitigating the already mentioned issues. The performance of DTT is compared to the basic FIFO discipline employed in commercial APs. All terminals always work with plain 802.11 cards. The maximum number of queued frames is the same for both the FIFO and the



(a) DTT



(b) PZC

Figure 2.12: Throughput for mixed UDP and TCP traffic — UDP station moves away

DTT schedulers. For the latter, it sums up the frames in all queues.

2.2.4.1 The E-model

The E-model [30] is an ITU-T standardized computational method for the assessment of the quality of voice connections as perceived by an average user. It allows the designer to calculate the expected speech quality given the transmission characteristics of the network and terminal equipment. The E-model takes into account many parameters, such as the effects of room noise, quantizing distortion, delay, and impairments due to codec and packet loss. The primary output of the model is the scalar rating factor R, that is calculated as:

$$R = R_0 - I_S - I_D - I_{Eeff} + A \tag{2.2}$$

In this equation R_0 represents the basic signal-to-noise ratio, including e.g. circuit noise and room noise, I_S is a combination of all impairments occurring simultaneously with the voice signal (e.g. quantizing distortion), I_D accounts for the deterioration caused by delay of voice signals, I_{Eeff} represents impairments caused by the equipment and packet losses, and the advantage factor A allows for compensation of the other factors when the user is likely to accept some degradation of the speech quality due to the adopted technology. The terms R_0, I_S, I_D and I_{Eeff} are further subdivided into more specific factors, for a total of about twenty atomic parameters. Describing them all is outside the scope of this thesis (the interested reader may refer to [30]); we will just mention the ones that are directly related to our study.

One of the most important element is the total delay T_a undergone by each voice packet since its creation. This time can be split into the packetisation time T_{pack}, the voice encoding/decoding time T_{DSP}, the network delay T_{nw} and the dejittering time T_{jit}. Network delay T_{nw} can be further divided in two parts, one depending on the wireless LAN (T_{WLAN}), and the

other representing the time to traverse the wired portion of the connection (T_{fixed}). Therefore:

$$T_a = T_{pack} + T_{DSP} + T_{fixed} + T_{WLAN} + T_{jit} \tag{2.3}$$

A second critical factor is the packet loss ratio P_{pl}. In our system, losses mainly occur in three situations: packets dropped at the AP due to buffer overflow (L_{of}), frames dropped after the retransmission limit has been reached (L_{ch}), and packets arriving at destination with a delay that cannot be compensated by the dejittering buffer (L_{jit}), thus resulting useless for the smooth reconstruction of the speech. This last term is inversely related to the temporal size of the dejittering buffer: the greater T_{jit} the smaller L_{jit}, but also the greater the total delay T_a. Packet losses have an impact on the I_{Eeff} factor, and can be mitigated by the robustness of the codec (B_{pl} factor). For a more in depth discussion about the dejittering issues, see again [29]. The rating factor R ranges between 0 and 100, being 100 the better possible connection quality. In most cases, a value of R higher than 70 represents an acceptable level of user satisfaction. Therefore this value will be our threshold for the analysis of network capacity.

2.2.4.2 The simulation environment

In this Section we describe the tool we made use of and the simulation scenario. The tool we have chosen to employ is the OMNeT++ simulator, version 3.0b1 [31]. Since the core library comes with no modules beyond the bare minimum, we have integrated it with the Mobility Framework (version 1.0a3) developed at the Technical University of Berlin [32], and with an accurate 802.11b MAC layer that we have built on our own. In particular, the 802.11 part was developed strictly adhering to the procedures defined in the standard, making no assumptions on channel model, collisions, etc.

Given that both the simulator and its parts are not yet deeply established in literature, we carried out some validation tests, comparing the results with known models. Specifically, our baselines were an analytic formula for the maximum achievable throughput with a single transmitting station [33] and Bianchi's performance study [4].

As for the first model, the maximum observed difference between the theoretical value and the simulation is in the order of 0.1%, which can be considered negligible. A very similar behaviour has been observed with regard to Bianchi's model. Figure 2.13 presents the results for a network of 10 stations working in saturation conditions. As it can be seen, the simulator matches pretty closely the theoretical values. Therefore we can conclude that the accuracy level of the employed tool is enough to guarantee reliable trials.

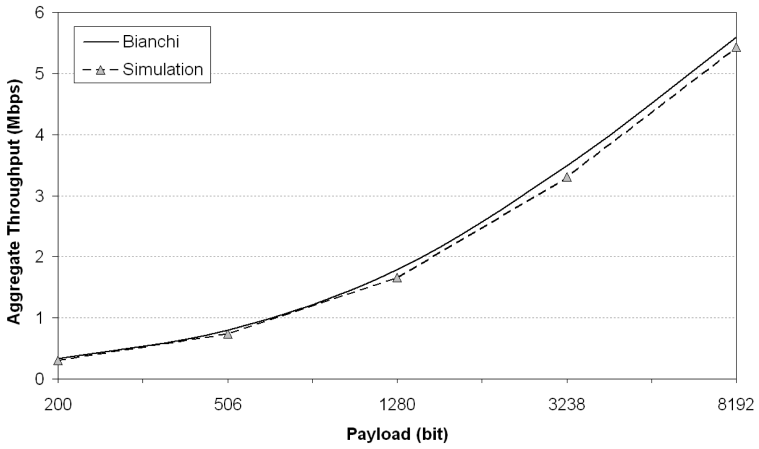


Figure 2.13: Throughput vs. payload size for a network of 10 stations transmitting at 11 Mbps

We simulated two kinds of scenario (see Figure 2.14). In the first (let us call it "A"), all stations are at the same distance from the AP and experience good channel quality, i.e. all frames are received correctly (apart

from collisions). All stations transmit at the highest rate, that is 11 Mbps. In the second scenario (say "B") one or more stations are far from the AP and therefore their links are very poor, causing frequent failures and the need of some retransmissions in order to deliver each frame. The far station(s) are placed at two different distances, so that two levels of poor quality are simulated. It is therefore convenient to distinguish the terminals into three classes: class I, comprising the stations with good link quality; class II, consisting of the stations at half the way; and class III, that refers to the farthest stations.

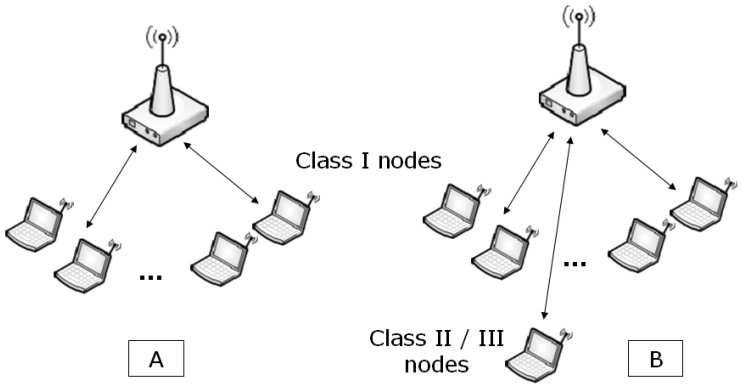


Figure 2.14: Simulation scenarios

In our simulator, following to the many existing solutions, we also implemented a simple ARF mechanism that lowers the physical bit rate after failed transmissions. A summary of the features of the simulated scenarios is reported in Table 2.1. Note that the number of retries only counts those due to the poor channel quality, neglecting collisions (which depend on the number of transmitting stations).

Note that scenario A is the ideal case, with all stations working at their best; on the contrary, scenario B-2 illustrates a very critical situation: while class III stations have no possibility to sustain any voice service (around

Table 2.1: Features of the simulation scenarios

Frame delivery failure rate and mean number of retries		
class I stations	0%	0
class II stations	0.15%	1.9
class III stations	29.1%	3.2
Class of stations per scenario		
Scenario A	I only	
Scenario B-1	I and II	
Scenario B-2	I and III	

29% of the frames is never received), they nevertheless spoil the chances of the other stations. We therefore expect a significant reduction in the system capacity, event that in fact occurs. Scenario B-1 represents a compromise, as class II stations do encumber the network, but they can still carry voice traffic.

In every scenario all the stations are involved in a bidirectional voice call in which the other end is represented by a remote terminal connected to the AP along a wired network. Voice frames are produced by a GSM-EFR encoder and encapsulated into an IP packet, which is in turn transported by the RTP protocol. Each voice source is modelled according to the ITU-T recommendation P.59 for artificial conversational speech: the source alternates on and off periods, whose length is described by an exponential distribution with mean 1 and 1.35 seconds respectively. During the on periods the source transmits at the GSM-EFR nominal rate (12.2 kbps), during the off periods it is silent. Table 2.2 resumes all the values.

As for the parameters of E-model, the choice of the codec and the network topology determined the values of some of the involved factors, while we assigned the remaining (in fact, the greatest part) the default values. A summary of the most meaningful factors, with their assigned

Table 2.2: Simulation parameters

Codec GSM-EFR	
Size of voice frames	244 bit
Frame interval	20 ms
RTP/UDP/IP header	320 bit
MAC IEEE 802.11b	
RTS/CTS	disabled
Retransmission limit	4
Max no. packets in all queues	150

Table 2.3: E-model parameters

Packetization time	T_{pack}	20 ms
Voice encoding/decoding time	T_{DSP}	10 ms
Wired network delay	T_{fixed}	50 ms
Dejittering time	T_{jit}	40 ms
Robustness to packet losses	B_{pl}	10
Equipment impairment factor	I_E	5
Advantage factor	A	0
Rating factor threshold	70	

value, is reported in Table 2.3. These are the typical figures for modern equipment (see e.g. [29][34]). Note that, since our attention is focused on the wireless access, we have given T_{fixed} a constant value. The results of the simulations gave the values of T_{WLAN} and P_{pl} for each run.

2.2.4.3 Simulation results

For each simulation scenario we tried to evaluate the maximum number of stations allowed in the network with all the users experiencing a satisfactory

speech quality, i.e. having $R \geq 70$. All voice calls started at the beginning of the simulation and lasted until the end. The simulation time was set to 210 seconds. For each scenario we averaged the results obtained in five runs with different seeds for the random number generator and collected the statistics for the stations with the worst quality in each class.

Figure 2.15 shows the R-factor for the first scenario with a different number N of user stations. In this context the insertion of DTT does not change the behaviour of the network, being the values very close to those for the plain FIFO strategy. Up to 24 stations can be supported with this configuration with either the standard FIFO discipline or the DTT scheduler. In accordance with [29] and [34], we also noted that the R-factor is much more sensitive to packet losses than to increased delays. Having a look at Table 2.4, where T_{WLAN} and L_{jit} are reported (L_{of} and L_{ch} are both zero), we can find the confirmation of the previous statement. In the case of 24 stations, L_{jit} registered for the FIFO policy is greater than for DTT, while T_{WLAN} is smaller. The resulting R promotes DTT, which gains 1.4 over FIFO. A similar position holds throughout all scenarios.

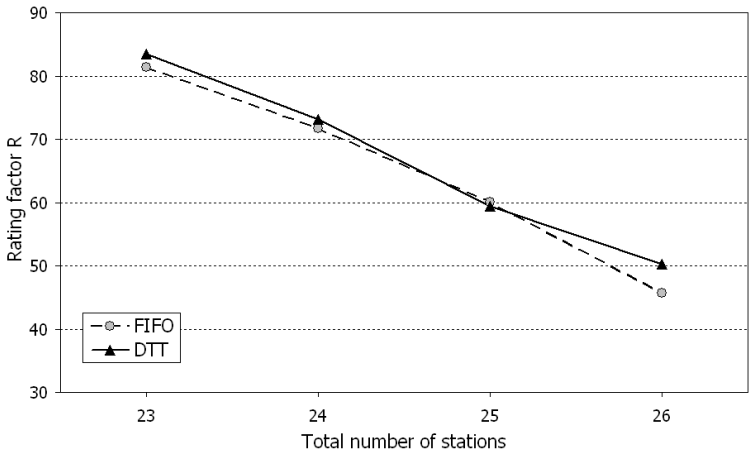


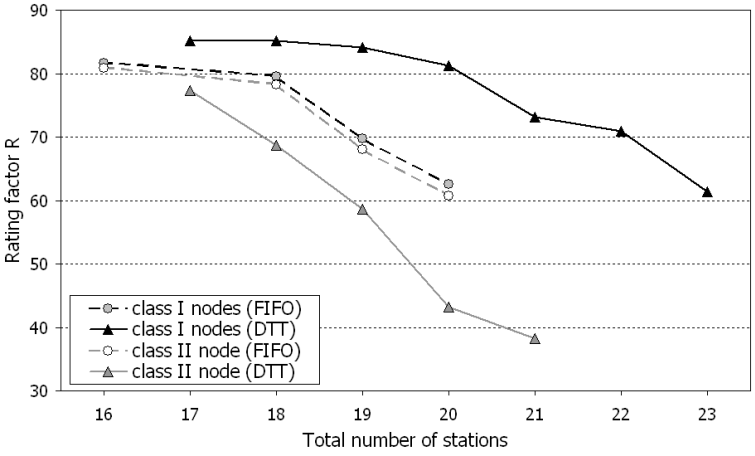
Figure 2.15: Rating factor vs. number of stations for scenario A

Table 2.4: Results for scenario A

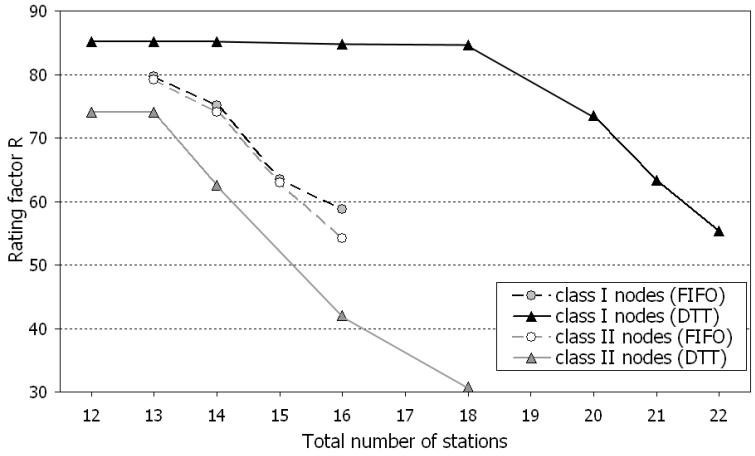
N	Scheduler	T_{WLAN} [ms]	L_{jit} [%]	R
23	FIFO	2.38	0.44	81.4
	DTT	2.13	0.21	83.4
24	FIFO	3.80	1.76	71.7
	DTT	3.93	1.55	73.1
25	FIFO	7.10	3.87	60.1
	DTT	8.81	3.99	59.4

As soon as a station with a not-so-good link quality joins the network, the two analysed schedulers start showing significant divergences. Figure 2.16 illustrates the outcome for scenario B-1 with one and two class II stations. The number of stations reported along the abscissa includes the terminal(s) in bad position too. The plain FIFO-based AP does not distinguish among the stations, therefore all users experience almost the same quality of the worst station (a clear realisation of the performance anomaly). Consequently we can see that no more than 18 users can be supported when there is just one class II station (Figure 2.16(a)). The slight difference in the R factor comes from the longer frame transfer delays and the small amount of lost frames. On the contrary, using DTT gives a sharp separation of the two classes. DTT is able to preserve a very good speech quality for near stations by penalizing the far stations. Therefore we can see how the system can support up to 22 users before they all experience a critical connection decline. If we do not count the single class II station that is impaired by the AP, the DTT provides a net gain of three users over the basic 802.11 discipline (the class II user cannot be counted). The gap between the two strategies gets larger and larger as the number of far stations increases. As an example, Figure 2.16(b) report the result for two class II stations. Under the FIFO governance, only 14 users are allowed in

the network, with a drop of 10 units with respect to the ideal case. On the other hand, DTT reduces the loss to 6 stations, 4 due to network saturation and 2 due to the serving policy.



(a) One class II station



(b) Two class II stations

Figure 2.16: Rating factor vs. number of stations for scenario B-1 — black lines refer to class I stations, grey lines to class II stations

Finally, we studied the performance with scenario B-2, when all stations but one are in a good position (class I), and the last station is subject to a very faulty link (a class III station). Note that the class III terminal, due to the high frame loss percentage, can never keep up a voice connection with an acceptable level of quality. Therefore we can limit the capacity analysis to class I stations only. Figure 2.17 plots the rating factor for class I stations only. In this case the adoption of the DTT scheduler has a major impact on system performance. The capacity is increased from 12 (FIFO policy) to 20 class I stations, i.e. the capacity is almost doubled. Moreover, when the FIFO-based AP is already in a critical condition, DTT can still offer a very high speech quality. This happens e.g. with 14 class I stations: DTT still registered the highest possible R (85.2) whereas the FIFO policy is already well below the threshold (47.0).

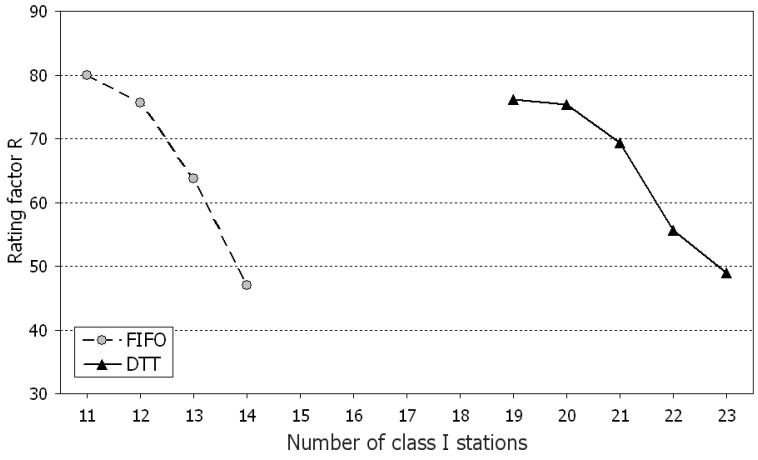


Figure 2.17: Rating factor vs. number of users for scenario B-2 — both lines refer to class I stations only

This scenario allows us to remark again that the main reason of capacity degradation is not network saturation, but the increasing value of jitter,

Table 2.5: Results for scenario B2

Scheduler	N	Class	Stations	T_{WLAN}[ms]	L_{jit}[%]	L_{of}[%]
FIFO	13	I	12	4.07	1.19	0
		III	1	10.9	1.37	
FIFO	14	I	13	6.08	3.12	0
		III	1	12.9	2.93	
FIFO	15	I	14	13.5	7.26	0.01
		III	1	21.6	8.14	
DTT	21	I	20	7.61	1.22	0
		III	1	510	33.9	
DTT	22	I	21	8.85	2.12	0
		III	1	576	33.3	

which makes more and more packets useless for speech reconstruction. Table 2.5, in addition to Table 2.4, also shows L_{of}. This parameter gives a measure of congestion through the number of frames dropped at the AP due to buffer overflow. It can be seen that L_{of} is zero, or close to zero, in all cases. At the same time, L_{jit} more than doubles for every added user under the FIFO scheme, whereas it increases more smoothly when the DTT is working.

2.3 The Distributed DTT scheduler

As described above, DTT is a centralized algorithm that offers an optimal — in the sense of proportional fairness [21] — scheduling to downlink flows. DTT has indeed proved to be very effective in contrasting the performance anomaly. Yet, due to its centralized nature, it has no control over the uplink flows. As a consequence, the same kind of misbehaviour still affects part of the network. We have therefore extended the idea and techniques of DTT

to a distributed version, called Distributed DTT, or $(DT)^2$. The aim of this scheduler is to complement DTT in order to protect both traffic directions — hence, the whole 802.11 network — from the performance anomaly.

2.3.1 Description of $(DT)^2$

The general architecture of $(DT)^2$ is sketched in Figure 2.18, while a flow-chart of the operations is reported in Figure 2.19. The scheduler is inserted in the device driver of the client stations, between MAC and network layers, and is transparent to both, so that compatibility with existing hardware and software is ensured. Please note that, while DTT will be deployed at the AP, $(DT)^2$ is meant to work on client stations.

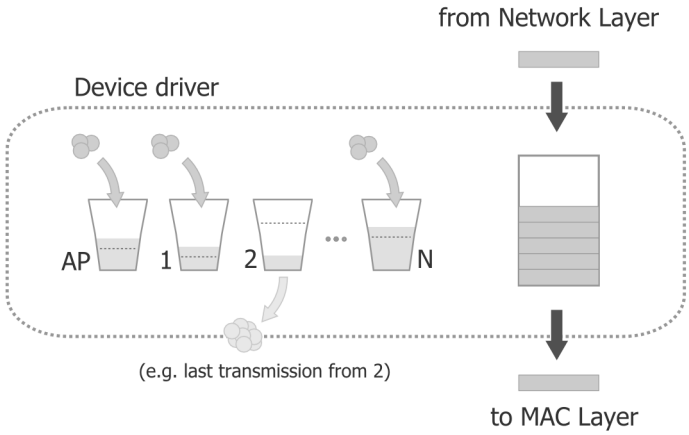


Figure 2.18: Elements and architecture of $(DT)^2$

Since the philosophy behind $(DT)^2$ comes from the already mentioned DTT scheduler, many of the operations are similar and will not be described again. Rather, in this Section, we will point out the main innovations and differences with its "ancestor".

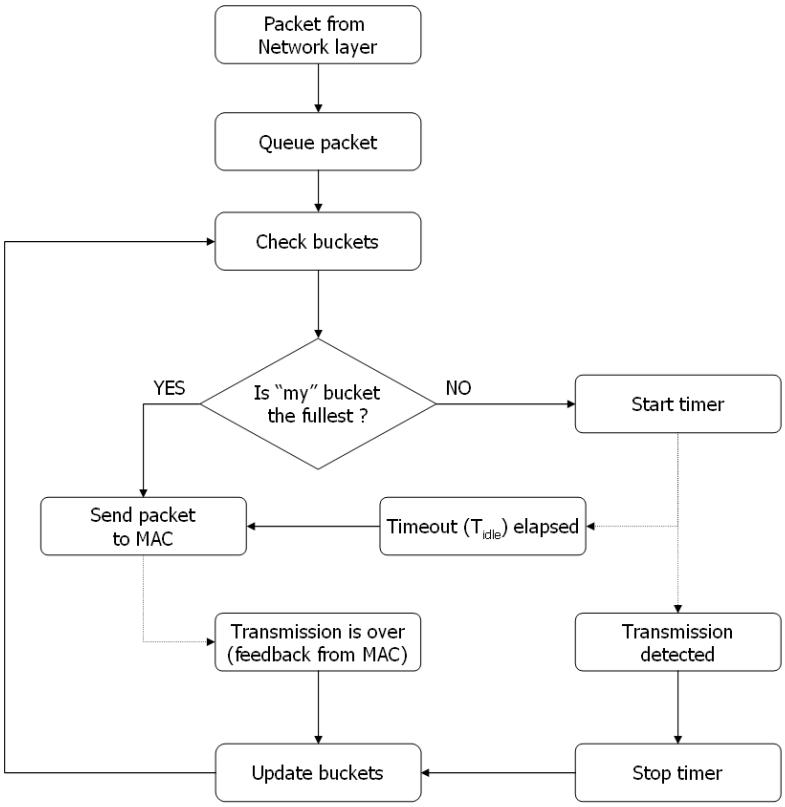


Figure 2.19: Flowchart of the operations of $(DT)^2$

Like DTT, each station equipped with $(DT)^2$ shall maintain a snapshot of the state of the network in terms of air-time usage of the channel. The scheduler allocates a set of buckets, one for every station in the network, including itself and the AP. Differently from DTT, all the outgoing data frames are stored in a single queue, which is still allocated in the device driver. As the goal is to give each station the same amount of air-time usage, regardless of the destination of the single transmitted frames, there is no need for multiple queues. Moreover, in practical cases, the whole client traffic will be addressed to one station only, the AP. Each bucket

holds some water that is in a direct one-to-one relationship to the channel occupancy time. The rules to fill and drain the buckets are very similar to DTT (see Section 2.2.1). At the end of every heard transmission, the scheduler computes the CFTT and drains it from the bucket associated to the originator of the heard frame[1]. Next, this value is divided by the number of stations and the result is used to fill the other buckets. All the buckets associated to a station that was neither a source nor a destination of heard frames in the last activity timeout ($T_{inactive}$) all cleared.

When a station has frames to transmit in the driver queue, the scheduler checks the buckets to find the fullest one. If that is associated to the scheduler own station, then it picks a frame from the queue and passes it to the MAC layer. At the end of the transmission (and every possible retransmission), the buckets are again updated as described before. If the fullest bucket refers to another station, the scheduler waits until the state of the network changes, so that an opportunity is given to the station associated to that bucket to transmit its frame. To avoid a possible stuck of the system, which may occur if the other stations have nothing to send, the scheduler starts a timer. When a timeout (T_{idle}) is elapsed, if no transmission has been heard, the scheduler passes the frame to MAC anyway. To prevent a simultaneous attempt to transmit by all stations, with a consequently high collision probability, the schedulers should avoid using the same T_{idle}. So we have provided for some randomness (10%) around the set value.

The described algorithm works fine under some assumptions, that we have implicitly made. First, we assume that the number of stations in the network is known. This can be achieved in several ways. For instance,

[1]With the term "originator" we mean the station that started a frame exchange sequence, i.e. the source of RTS and DATA frames and the destination of CTS and ACK frames.

it becomes straightforward if the stations can all hear each other, either directly, or through the use of the RTS/CTS mechanism. Otherwise, the AP may inform the stations by inserting this value into the periodic beacons (but this may require a modification to the underlying MAC layer), or into a special frame that is understand by the schedulers in the stations. Alternatively, an estimation algorithm can be used (e.g. [35][36]). Second, the scheduler works perfectly if the states of the buckets in all stations are synchronized. If all stations agree on the station that should access the channel next, this will result in a fast and easy transmission (having no contention). On the contrary, if the bucket states are not aligned, it may occur that more than one station, or no station at all, will try to transmit. In the first case, the contention will nevertheless occur between a limited number of stations, depending on how much the states are out of synchronization. In the second case, the timeouts set by each scheduler will solve the temporary idleness of the network. We will show in Section 2.3.2 that even in a pretty unlucky case the scheduler still works fine.

Finally, we believe that T_{idle} is worth a remark. At a superficial glance, this timeout could appear like a replica of the 802.11 backoff procedure, but this is not true. The presence of T_{idle} should be evaluated in the context of the whole scheduler. $(DT)^2$, by implementing a smarter medium access strategy, trades this little waste of time (and maybe throughput), with an improvement in the network capacity. Especially with regard to delay sensitive applications, sheer throughput is not the sole and main performance parameter — as explained Section 2.2.4 — and loosing a bit of throughput should not be regarded as a shortcoming at all.

$(DT)^2$ retains all the advantageous features of DTT (see Section 2.2.2). Moreover, an interesting, and easy to obtain, side effect of $(DT)^2$ is the uplink/downlink balancing. Similarly to DTT, $(DT)^2$ may allow some stations to take more or less control of the channel by assigning a different

weight to the different buckets. In particular, if the water held in the bucket associated to the AP is counted N times (being N the number of client stations), the AP could access the medium N times more frequently than each station and hence achieve a throughput N times higher. This strategy is particularly useful when the traffic in the uplink and downlink directions is roughly the same, as it happens for example for voice services. Indeed it has been proven that the uplink/downlink unbalancing (also known as bottleneck effect) is one of the main causes of the limited support of voice and video services over WLANs [37][38].

2.3.2 $(DT)^2$ for VoIP applications

To verify the effectiveness of $(DT)^2$ we carried out a series of simulations. The tool is the same used for DTT, i.e. the OMNeT++ simulator (see Section 2.2.4.2). As the scheduler is mainly targeted at real-time services, our tests were based on a scenario where the users run VoIP applications. The performance of $(DT)^2$ is compared to the default FIFO discipline on the basis of the R-factor (see Section 2.2.4.1).

Voice traffic is modelled according to the ITU-T recommendation P.59. The source alternates on and off periods, whose length is described by an exponential distribution with mean 1 and 1.35 seconds respectively. The voice codec is G.729, which produces 160 bit of payload every 20 ms, i.e. a net throughput of 8 kbps. The RTP/UDP/IP headers (320 bit) complete the packet that is delivered to the MAC layer for the physical transmission. Table 2.6 reports the main parameters of the service, together with the most meaningful factors of the E-model related to this codec.

Table 2.6: Parameters for the G.729 codec

Parameters of a G.729-based service	
Payload of voice frames	160 bit
RTP/UDP/IP headers	320 bit
Frame interval	20 ms
Duty cycle	0.426
Average throughput	10.22 kbps
Some E-model factors	
Packetization time (T_{pack})	20 ms
Voice encoding/decoding time (T_{DSP})	10 ms
Wired network delay (T_{fixed})	50 ms
Dejittering time (T_{jit})	40 ms
Total delay at WLAN entry	120 ms
Robustness to packet losses (B_{pl})	18
Equipment impairment factor (I_E)	10
Advantage factor (A)	5
Maximum achievable R	85.26

2.3.2.1 Topology and settings

The simulation scenario (see Figure 2.20) is somewhat different from the one used for DTT. The AP is placed in the centre of a 100 $m \times 100$ m field and is surrounded by a variable number of client stations. Each client is involved in a voice connection with a peer on the wired network. The AP is the gateway between the wireless and the wired worlds. All terminals are equipped with 802.11b cards. We ran four series of tests, varying the channel features and the position and settings of the stations (see Table 2.7).

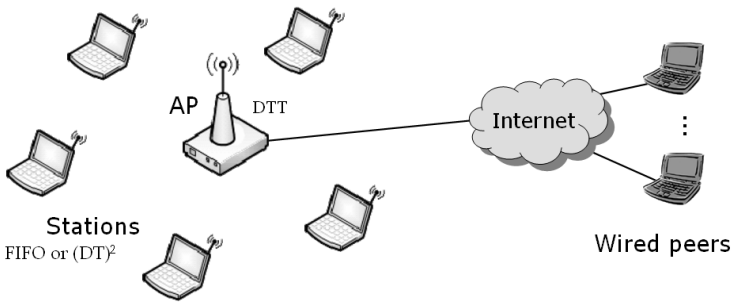


Figure 2.20: Topology of the simulated network

Scenario I. In the first series we tested $(DT)^2$ in a rather didactic case. We placed two stations at the very edge of the field and the remaining ones very close to the AP (within a 10 m-side square). This is a typical way to put in evidence the performance anomaly, since the two far stations will suffer the worst channel conditions and the close stations enjoy an almost perfect link to the AP. The channel attenuates the signals and introduces some errors. With these settings, the station the farthest from the AP needs to retransmit a frame with probability 0.79 (the same is true for the AP, being the channel symmetrical), whereas the stations close to the AP will

always receive a frame correctly. Moreover, mimicking most commercial cards, we implemented an ARF algorithm.

Scenario II. The stations are placed randomly across the whole field, thus experimenting much more variable channel qualities. The propagation rules and MAC settings are the same as before. This scenario is meant to be more realistic.

Scenario III. Up to now, since it may easily happen that some stations cannot hear each other, both the estimate of N and the bucket states may present some discrepancies across the whole set of client stations. $(DT)^2$ then works pretty far from the optimal conditions. Therefore, in the third series, we enabled the RTS/CTS mechanism to broaden the sensing range and reduce the number of hidden terminals and hence enhance the knowledge of the network. The goal is to evaluate how the improvement on the estimate of N and on the alignment on the bucket states increases the performance of $(DT)^2$.

Scenario IV. Finally, we ran some tests in an ideal scenario. The channel does not introduce attenuation nor errors on the propagated signals, hence the only reception errors are due to collisions. All stations transmit at the maximum bit rate, i.e. 11 Mbps, with no RTS/CTS handshaking and no ARF. Since all stations can hear each other, the number N of stations in the network is perfectly known and the states of the buckets are perfectly synchronized. This test was used to show the best performance achievable by $(DT)^2$, including a better balancing between the uplink and downlink directions.

For every scenario the FIFO and $(DT)^2$ schedulers were alternated on the whole set of client stations, while the AP was always equipped with the DTT scheduler. This allows a basic improvement of the performance of downlink flows without affecting the uplink traffic (the MAC layer at the AP is unchanged and totally conform to the 802.11 standard). As for

Table 2.7: Settings for the simulation scenarios.

	I	*II*	*III*	*IV*
Attenuation factor	2.85	2.85	2.85	0
Frame error probability	0/0.79	0÷0.79	0÷0.79	0
RTS/CTS	off	off	on	off
Automatic rate fallback	on	on	on	off
Retry limit	4	4	4	4

$(DT)^2$, all tests were run with $T_{inactive} = 1$ s and $T_{idle} = 450$ s. All the points reported in the figures come from the average over all stations of ten simulation runs with different seeds.

2.3.2.2 Performance analysis

Figure 2.21 reports the R-factor for the first scenario, when the performance anomaly is more emphasized. The curves distinguish between uplink and downlink flows, between station position (far from and close to the AP), and between the FIFO and $(DT)^2$ schedulers. The gain of $(DT)^2$ over the basic FIFO discipline is apparent. If we refer to all stations, no more than 5 voice calls can be set up with a satisfactory quality (R greater than 70) when the stations use the FIFO strategy. On the other hand, when $(DT)^2$ is at work, 18 calls con coexist in the network. This result is indeed remarkable, as the gain is more than double.

A more in depth analysis of the curves shows that the bad performance of FIFO is mainly driven by the downlink traffic directed to the two far stations. Even with very few users, its performance is barely above the threshold we have set for the R factor. This should not be surprising, as the two stations experience bad channel conditions and it is known that the downlink of 802.11 systems is more penalised than the uplink. The

point is that the uplink traffic too does not reach the highest R values, yet it consumes a lot of network resources. As a consequence, also the other flows suffer a performance degradation. In particular, downlink flows to the stations near the AP cannot even reach a score of 80.

The insertion of $(DT)^2$ brings considerable improvements. While the uplink traffic from the far stations is roughly at the same level as before, all the other flows register an increased R. Both far and near stations can enjoy a very good call quality (R close to 80) up to 16 users. At this point the quality starts degrading for those users at the edges of the field, but it remains almost constant for the stations close to the AP. These perceive a sharp degradation only when the number of users becomes higher than 18/19.

It might be practical, in some cases, to focus only on the users close to the AP, assuming for example that users too far from it will try to move closer or give up the connection. In this case (see empty symbols in Figure 2.21) the number of admissible stations is 14 with FIFO and 21 with $(DT)^2$. The difference seems to be smaller. However, we should also consider the quality. $(DT)^2$ allows the stations to reach almost the maximum achievable R, whereas if the users employ the FIFO discipline they cannot expect more than 78, which is still a tangible difference.

The results for the more realistic scenario II, with the stations randomly scattered across the whole field, are reported in Figure 2.22. Both schedulers perform better than before, with a capacity increase of one station each (if compared to the near users of Figure 2.21). This is the effect of a less extreme positioning of the stations, which reduces both channel losses and the hidden terminal problem. In particular, all uplink flows can now reach the maximum achievable R. Going to numbers, $(DT)^2$ increases network capacity by 47% with respect to FIFO. Moreover, the average quality experienced by the users when $(DT)^2$ is working is the best it can be

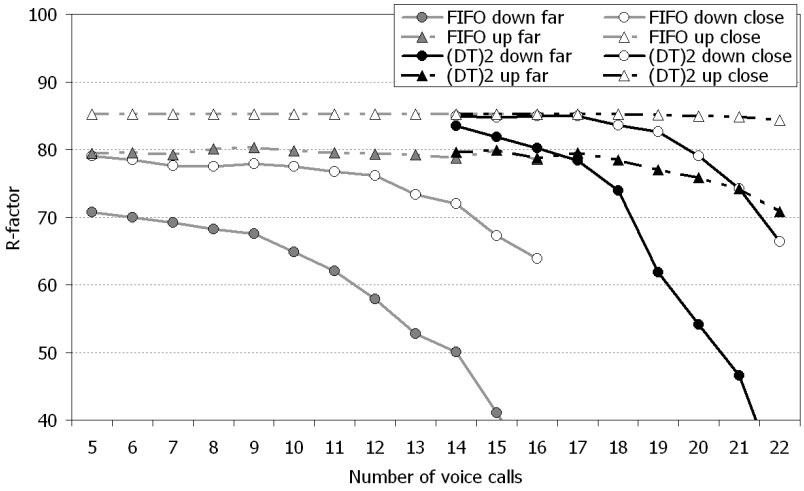


Figure 2.21: R-factor vs. number of voice calls for scenario I — black lines refer to $(DT)^2$, grey lines to FIFO

achieved, whereas FIFO cannot offer an R factor higher than 78.5. This proves that, even when the network conditions are not particularly tough, users employing our solution can still expect a valuable improvement in the perceived voice quality.

An interesting remark should be done about the quality of the uplink flows. When FIFO is running, the R factor registered in the uplink direction is constantly 85.25. This means that all the degradation is put on the downlink traffic. On the contrary, when the stations are equipped with $(DT)^2$, the uplink traffic too suffers from network congestion. This is an effect of the strategy pursued by our scheduler, that tries to increase network capacity by isolating the flows from each other and by distributing channel impairments to both directions. A more in depth explanation on this point is given in Section 2.3.2.3.

In the third series of tests (scenario III) we have enabled the RTS/CTS mechanism. This additional overhead drastically reduces the capacity of

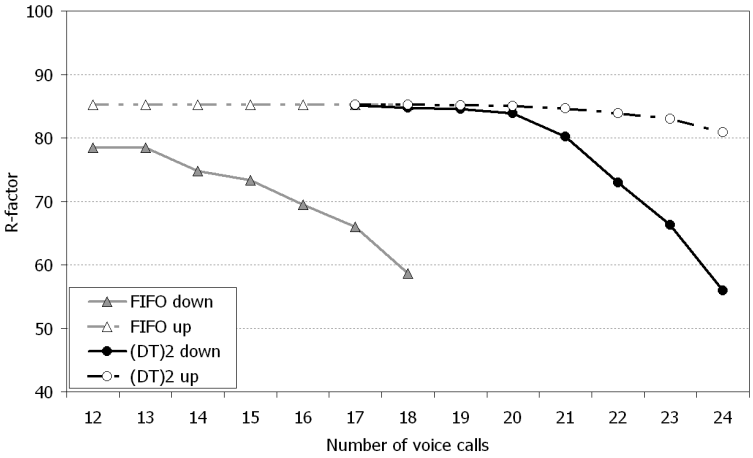


Figure 2.22: R-factor vs. number of voice calls for scenario II

the network. No more than seven client stations can coexist in the network when they run the basic FIFO scheduler, and no more than 10 when $(DT)^2$ is at work (see Figure 2.23). The improvement brought by our solution is therefore in the order of 43%. In this case it is even more apparent how $(DT)^2$ tries to charge the losses to both uplink and downlink flows. However, no matter what scheduler is at work, the capacity is more than halved when the RTS/CTS mechanism is employed. Hence we argue that, even when the hidden terminal problem can be a serious issue, enabling the RTS/CTS handshaking is not a worthy solution, especially when the traffic is mainly produced by voice (or similar) applications.

2.3.2.3 Facing the bottleneck effect

As previously outlined, an interesting and useful side effect of our scheduler is the possibility to tune it to achieve a better balance between the uplink and downlink flows. It is well known, and it has also emerged from the previous tests, that the IEEE 802.11 standard favours uplink traffic (see all

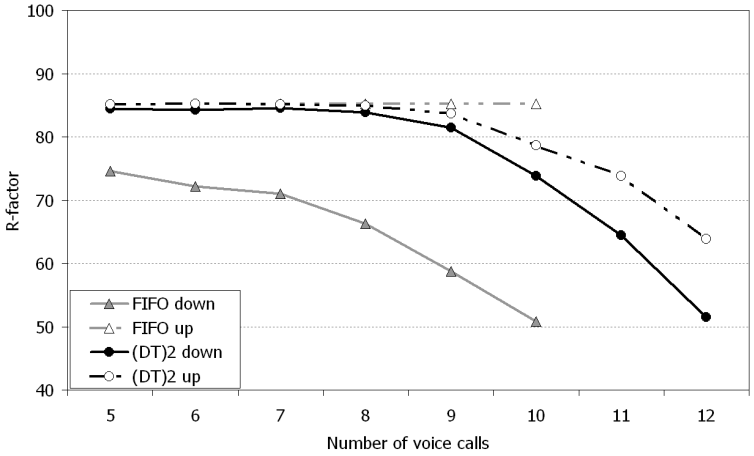


Figure 2.23: R-factor vs. number of voice calls for scenario III

the curves referring to FIFO: the uplink is almost constantly at its top). Particularly for voice services, the AP has N times more data to deliver and the same channel access probability than each single station. The unfairness is therefore clear.

The architecture of $(DT)^2$ offers an easy way to lessen this unfairness. Beyond isolating all the uplink flows through the achievement of proportional fairness, the scheduler can be easily instructed to assign weights to the different buckets. By "multiplying" the water in the bucket associated to the AP, this bucket will have more chances to result the fullest, all the stations will expect a transmission from the AP more often, and consequently it is possible to increase the AP channel access rate. We expect that the downlink flows will benefit from this strategy, and at the same time we also believe that uplink flows will not undergo a severe quality degradation. Since we do not work in saturation conditions (in fact, real-time services need the network to be pretty far from the saturation point, as it can be easily inferred from the presented results), giving the AP more chances to access the medium does not mean subtracting bandwidth to the

other stations.

To prove this statement and to prove that $(DT)^2$ is effective in balancing the flows, we report in Figure 2.24 the outcome of a test run in ideal conditions (scenario IV). The R factors for the up and down directions for $(DT)^2$ have the same trend, and both uplink and downlink flows can reach their best performance. Moreover, from the similarity with Figure 2.22, we can also deduce that the non-perfect knowledge of the network that $(DT)^2$ has in real environments does not undermine its efficiency, since the loss from ideal conditions is limited to one station only.

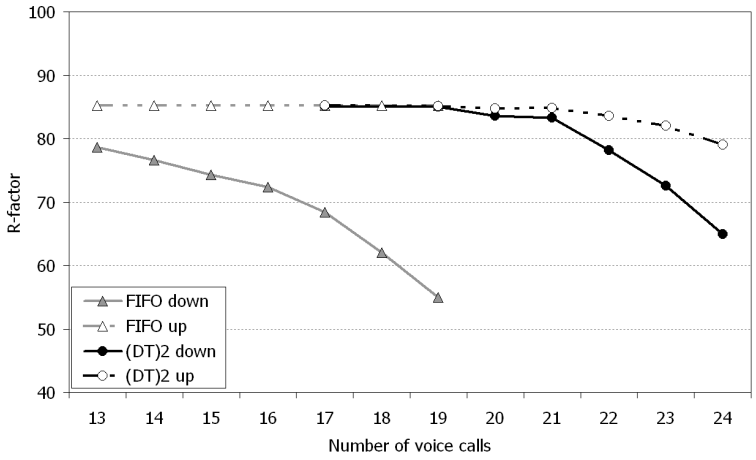


Figure 2.24: R-factor vs. number of voice calls for scenario IV

2.4 Conclusions

In this Chapter we described and analysed DTT, a channel-aware wireless scheduler for IEEE 802.11 networks. Its main goal is overcoming the performance anomaly, which is a phenomenon strongly tied to the max-min throughput fairness implemented by the 802.11 MAC algorithm. Starting

from this awareness, we designed a scheduler based on a different principle: proportional fairness. This property is actualised imposing that the air-time usage shares of every traffic flow in the network are equal. If that occurs, the throughput of any flow becomes independent from the others (flow isolation). An indirect but reliable measure of the link quality, the Cumulative Frame Transmission Time (CFTT), is the tool to perform such scheduling.

We proved the effectiveness of DTT in several operational conditions. We have realized a working implementation of the scheduler on a Linux-based AP, and performed some measurements in a scenario dominated by downlink data traffic. Figures 2.6-2.12 prove that DTT can separate the flows toward each station. While the "far" user see its bandwidth severely reduced, the closer stations can still exploit their good links. The improvement is noteworthy, and is particularly beneficial to TCP traffic, which can be kept at high throughput levels even though other bandwidth-hungry flows (e.g. UDP) are loading the network.

Then we assessed the performance of DTT for transporting real-time services such as voice over IP. This has been done via simulation within the framework of the standardised E-model. From the trials, it has clearly emerged that the insertion of DTT brings a significant improvement in terms of network capacity. Remarkably, in all "B" scenarios the closest stations are able to sustain a voice connection almost until the network saturates, which occurs almost at the same stage as in the ideal "A" case (see Figures 2.15-2.17).

The enhancements brought by DTT can be further increased with the insertion of $(DT)^2$ on the mobile terminals. This gain is both in network capacity and in the quality perceived by the users.

DTT, and its sibling $(DT)^2$, have therefore multiple advantages. Most importantly, it is the first scheduler to actualise proportional fairness in

IEEE 802.11 networks. To the best of our knowledge, no other prototype exists that implements this important and very useful property. Then, it runs a simple algorithm, needing no complex channel modelling or heavy computations. It quickly reacts to variable link conditions and data rates. It can accommodate different scheduling policies, on the basis of a different distribution of the "water" in the internal "buckets". As an example, this method has been used to overcome the uplink/downlink unbalancing (with $(DT)^2$). It is transparent to both user applications and network interface cards, thus being compatible with all the deployed hardware. This is particularly appealing, as it could be hosted on all 802.11 based devices with a simple software/firmware upgrade. Furthermore, it could also be coupled with the emerging 802.11e systems. In fact, since the performance anomaly affects all 802.11 flavours in the same way, DTT could be a valid solution also for this increasingly important version.

Chapter 3

Admission Control in IEEE 802.11e Networks

The recently approved 802.11e amendment introduced traffic differentiation in order to meet the demands of the growing number of users of real-time services. The distributed access method (EDCA) does improve the support of these services but, as we have seen in Chapter 1, it does not provide any strict QoS guarantee. Rather it just offers delay-sensitive frames a privileged way to access the transmission medium. So, EDCA behaviour is not completely predictable, and to move from simple service differentiation to real provision of QoS objectives we should switch to the centralized control offered by HCCA.

Unfortunately, HCCA has received scarce attention from the industrial world and seems destined to follow the unlucky faith of its predecessor PCF, given that the first commercial 802.11e products only implement EDCA. Therefore, it becomes crucial to enhance EDCA operations with additional mechanisms that allow it to provide a better form of QoS. These mechanisms are know as admission control (in short, a.c.). The same IEEE 802.11e standard suggests a distributed a.c. algorithm in which the Ac-

cess Point can govern traffic load by periodically announcing the available bandwidth for each class. This algorithm, however, is rather complex and of difficult implementation.

In the scope of admission control, it assumes a paramount relevance the determination of the "admission region", i.e. the maximum number of users for each traffic class that can be admitted to the service while satisfying the respective QoS requirements. An a.c. algorithm should be able to determine this region exploiting all its knowledge of the network (e.g. number of users/services, traffic load, traffic requirements). Then, it should prevent new connections to be established if their presence would lower the level of QoS experienced by pre-existing services. Given this features, it is straightforward to place the a.c. algorithm at the Access Point, since it is the terminal that has global network awareness.

In this Chapter, we present at first the results of a simulation study aimed at defining the admission region for videoconference and VoIP sources in an 802.11e wireless LAN. The number of sources that can be accepted has been evaluated considering the actual QoS requirements that can be assumed for these services. The presence of TCP traffic was also considered. Then we describe two admission control algorithms that we devised in this context. Both algorithms are based on the concept of time occupancy of the medium, which has been proved to be an effective tool for managing 802.11-based networks (see Chapter 2).

By the way, the most interesting previous work on this topic is the one by Chen et al. [39]. The authors propose two call a.c. schemes that relies on the average delay estimates and the channel busyness ratio, defined as the portion of the time that the channel is busy in an observation period. An analytical model is built to derive the average delay estimate for traffic of different priorities in an unsaturated network. When deciding on the acceptance of a new real time flow, the a.c. algorithm considers its effect

on the channel utilization and the delay of existing flows. The proposed G/M/1 and G/G/1 models deliver a rough upper bound for the average delay, which becomes looser as the number of flows increases. As a consequence, the proposed a.c. schemes can only suggest a pessimistic limit on the number of admissible users for small-size networks. Yet, they prove the effectiveness of using a time based metric for regulating the access.

3.1 The Admission Region

In this Section we present a simulation study to evaluate the admission region for two relevant QoS-aware services: VoIP and videoconference. Note that the admission region is a function of the QoS requirements of the services the network is expected to support.

3.1.1 Criteria for estimating the Admission Region

The Admission Region is defined in a cartesian plane, where abscissa and ordinate represent respectively the number of VoIP and videoconference sources that can be admitted in the system while guaranteeing their QoS requirements. For the definition of the QoS parameters associated to the considered services, we refer to ITU-T recommendations Y.1540 [40] and Y.1541 [41]. The former defines the QoS parameters and how to measure them, the latter introduces the concept of Class of Service (CoS) and for each CoS indicates the maximum values that the QoS parameters should not exceed. The considered parameters are the following:

- IPTD (IP Packet Transfer Delay): the time to transfer a packet from the network interface of a measurement point (e.g the transmitter) to that of the companion measurement point (e.g. the receiver).

- Mean IPTD: the arithmetic average of IPTD for a population of interest.
- IPDV (IP Packet Delay Variation): the difference between IPTD and a fixed reference IPTD value, which can be assumed equal to the Mean IPTD.
- IPLR (IP Packet Loss Ratio): the ratio of the total lost IP packets to the total transmitted IP packets.
- IPER (IP Packet Error Ratio): the ratio of the total errored IP packets to the total of successful and errored IP packets.

These parameters are estimated with regard to the single packet flow. Table 3.1 summarizes the maximum values suggested by recommendation Y.1541 (NS indicates that the value for the parameter is Not Specified). In Classes 0 and 1 the upper bounds are defined for all the parameters. This indicates that these CoSs have been thought for real-time delay-sensitive services, e.g. Voice over IP and highly interactive videoconference. Classes 2 and 3 differ from Classes 0 and 1 only in terms of IPDV, which is not specified. Hence, they can be adopted for data transfers requiring only IPTD constraints, such as signaling services. Finally, Class 4 is defined for services with no strict delay constraints (e.g. data transfer and videostreaming), while Class 5 is for best effort services.

For the estimation of the admission region, we also assumed that packets exceeding Mean IPTD and IPDV are dropped by the application. Then, for each flow, we computed a Virtual IPLR (vIPLR) as the ratio of the total number of discarded packets to the transmitted packets. The first term is the sum of the packets lost at MAC layer (due to buffer overflow or to to reaching the maximum number of retransmissions) and discarded by the application as defined above. Finally, we assume that a new user service

Table 3.1: Upper bounds for different QoS parameters

Class	0	1	2	3	4	5
Mean IPTD [ms]	100	400	100	400	1000	NS
IPDV [ms]	50	50	NS	NS	NS	NS
IPLR	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	NS
IPER	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	NS

cannot be accepted if its activation implies that the estimated vIPLR of at least one of the active CoS exceeds the IPLR upper bound.

Finally we established the associations between Y.1541 CoSs, 802.11e ACs and the services considered in our study. In detail, we reckoned VoIP services to require the QoS constraints of Class 0, and videoconference those of Class 1. Then we mapped Class 0 to AC 0 and Class 1 to AC 1, so that VoIP traffic would be transported with AC 0, and videoconference with AC 1. TCP data traffic was associated directly to AC 2.

3.1.2 Simulation framework

The simulation study was carried out using Network Simulator 2 (ns-2) version v2.26 [42] patched with an extension developed by the Technical University of Berlin that models the EDCA mechanism. In particular, this patch has added the four different AC parameters defined in the EDCA, i.e. AIFSN, CW_{min}, CW_{max} and $TXOPLimit$, and the possibility to transmit a sequence of frames for a time up to TXOPLimit after winning the contention. The model also allows mapping the traffic flows to the proper AC by means of a dedicated field in the IP header of ns-2.

We performed a few tests to validate the simulation tool. Simulation and theoretical results were compared in a simple scenario. We focused on the ACs having the TXOPLimit equal to zero, i.e. AC 2 and AC 3.

In these conditions, we evaluated the maximum throughput on a simple point-to-point connection, hence without collisions. The obtained results showed a good accordance among theoretical and simulation values, with differences within 1% in the worst case (see [37] for more details).

The simulation scenario is illustrated in Figure 3.1. There is a variable number of wireless stations running a VoIP or videoconference session with a companion station on the wired network. In addition, a single couple of stations simulates an ftp data transfer. All mobile terminals support the 802.11e EDCA mechanism. The distance between the AP and the stations is very small, in order to mimic ideal channel conditions. Furthermore, the wired link was characterized with very high bandwidth and very small latency to place in the wireless part the only bottleneck of the system.

Bidirectional voice and video traffic used in the simulation was generated through a simple real testbed. Voice calls were based on the G.723.1 codec with VAD (Voice Activity Detection), which produces a 24-byte payload every 30 ms (for a total information rate of 6.3 Kbps) when voice activity is detected. The videoconference service employed the H.263 codec. For this we have taken two different traffic data sets, referring to audio and video packets transmitted by one of the two involved users. From the recorded acquired data we produced the trace files holding IP size and inter-arrival time of the packets belonging to a particular traffic flow.

Several sets of simulations have been carried out using different numbers of VoIP and Videoconference sessions simultaneously active. Each set was run ten times with diverse seeds for the random number generator. From these data we extracted the mean values of all QoS parameters and also the 95% Confidence Interval (95%-CI) for the vIPLR parameter.

The admission region is the set of scenarios where the constraints on the upper value of IPLR are satisfied (i.e. a value lower than 10^{-3}, found

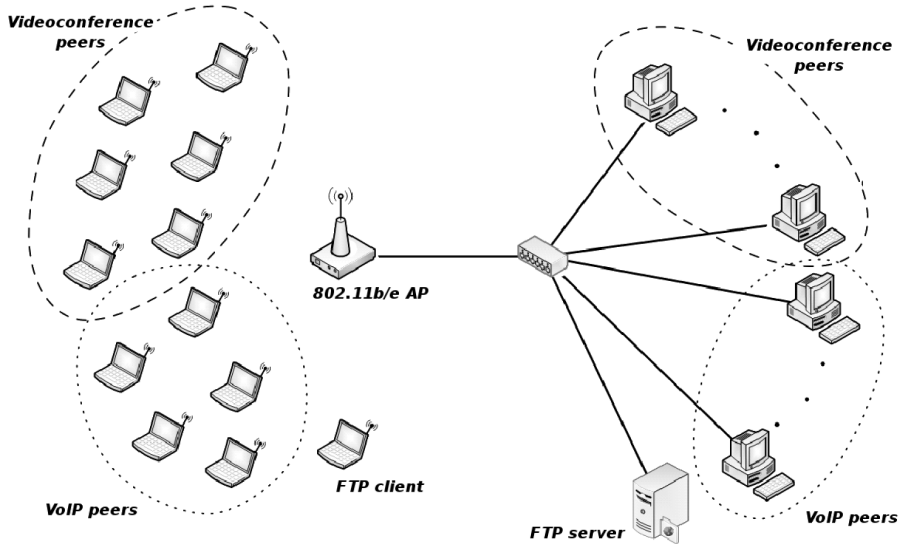


Figure 3.1: The simulation scenario

for the 95%-CI of vIPLR). The analysis regards both directions of traffic flows, i.e. from the mobile station to the wired peers and vice versa.

3.1.3 Analysis of the results

We start the discussion from the simple case where only VoIP sources are active. We recall that the TCP connection is always present. The obtained performance parameters are reported in Table 3.2, and refer 21 and 22 VoIP sources (columns named "0-21" and "0-22"). For each flow we observed about 20000 packets. From the figures, we can deduce that the system is unable to satisfy the QoS requirements for 22 VoIP sources as the constraint on vIPLR is not satisfied in the downlink stream (the value exceeding the limit is reported in bold). Hence, a first point delimiting the admission region is zero videoconference and 21 voice sources. It should be noted, however, that the uplink vIPLR is always under the 10^{-3} upper bound, even

Table 3.2: Simulation results for scenarios 0-21 and 0-22

	0-21		0-22	
	Uplink	Downlink	Uplink	Downlink
Mean IPTD (ms)	1.72	2.87	1.82	3.06
Lost packets	1.55	5.95	2.66	9.09
Discarded packets for IPTD	0.08	2.23	0.21	5.46
Discarded packets for IPDV	0.6	4.31	0.86	8.72
vIPLR (10^{-4})	1.11	6.29	1.85	**11.7**
95%-CI (10^{-4})	0.789	2.06	1.14	4.03

in the "0-22" scenario. This is due to the bottleneck role played by the AP: considering all the four ACs, the AP has the same probability to acquire the right to transmit a frame as whatever client station, while it is expected to transmit a traffic that is about N times higher (with N being the number of mobile stations, and in the hypothesis of having symmetrical session). For completeness, Figure 3.2 shows the complementary probability of IPTD for upstream and downstream traffic in the two cited scenarios. The different behaviour in the two directions is apparent.

In a second set of simulations we considered only videoconference sessions. Table 3.3 reports the results for five videoconference users (scenario 5-0) which show that the considered system is unable to satisfy the required QoS figures. In this case the number of transmitted packets for each direction is about 59000. The maximum number of active sources is imposed by the IPDV and vIPLR parameters in the downlink direction. In addition, the large packet sizes produced by the video codec lead to higher performance differences between uplink and downlink. In particular vIPLR for the downlink flow is three order of magnitude higher than the other. It is also worth noting that the main contribution to vIPLR is given by the packets discarded for overcoming the IPDV upper bound.

After evaluating the boundaries of the admission region for homogeneous QoS sources (either VoIP or videoconference), we took into account all the other scenarios with a mix of the two services. We avoid reporting the outcome for all the specific scenarios and we jump directly to the aggregated result, which is in Figure 3.3. The grey area indicates the admission region, in terms of VoIP (x-axis) and videoconference (y-axis) sources. It can be observed the low number of simultaneous videoconferences that can be admitted. Many more VoIP services can be supported, but if we relate this number to the actual amount of produced traffic (each VoIP source generates about 11 Kbps per direction), we can esily calculate that the sum is well below the maximum theoretical bandwidth of the network. The number of QoS-aware services that can be simultaneously supported by the 802.11e technology is therefore surprisingly low.

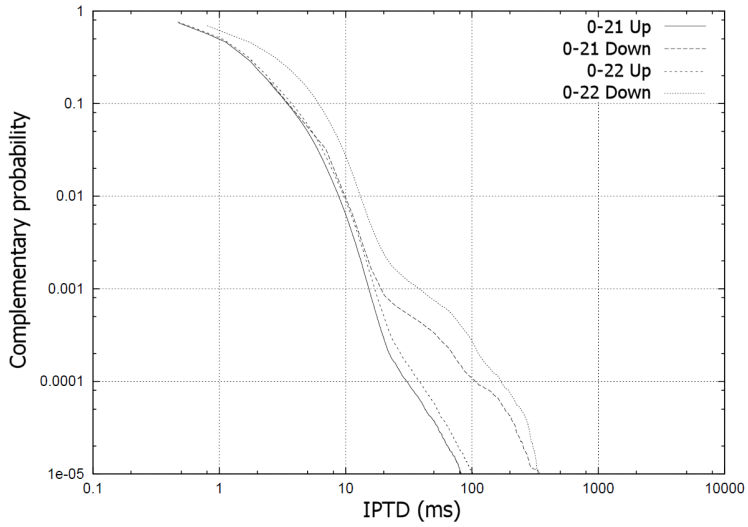


Figure 3.2: Complementary probability of audio IPTD for scenarios 0-21 and 0-22

Table 3.3: Simulation results for scenario 5-0

	Uplink	Downlink
Mean IPTD (ms)	3.59	9.05
Lost packets	0.12	3.78
Discarded packets for IPTD	0	0
Discarded packets for IPDV	0.06	**206.38**
vIPLR (10^{-4})	0.03	**35.5**
95%-CI (10^{-4})	0.18	27.2

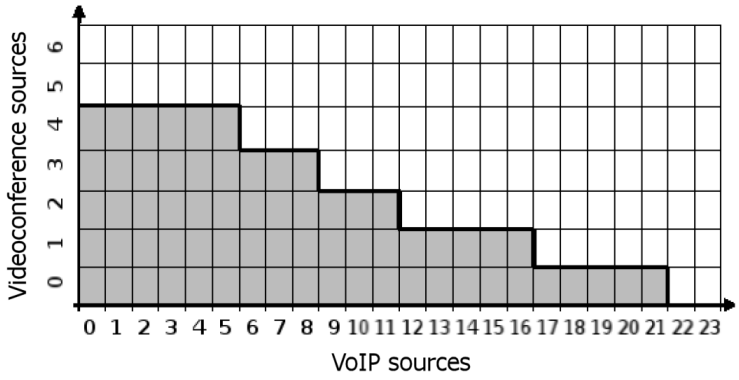


Figure 3.3: The Admission Region

3.2 Model-Based Admission Control

The a.c. scheme we describe in this Section operates on some input parameters retrieved by an underlying analytical model of EDCA. Instead of developing our own model, we build on the one described in [15]. An interesting feature of this model is that it also accounts for the non-saturation condition, thus being closer to a real scenario. Most previous works are instead based on saturation models, and this has been pointed out to be a major drawback (see e.g. [43] for a review of the related literature). The analytical model is used to extract the probability of unsuccessful trans-

mission, which is then exploited by our algorithm to estimate whether accepting a new user will cause an unbearable degradation of the quality of the ongoing communications. This assessment is performed on the exclusive basis of the temporal occupancy of the medium. We test whether, in a generic reference period, there would be time for all users to transmit their frames. We show that this simple test, which does not involve any complex computation on the traditional QoS parameters (e.g. bandwidth, delay, delay jitter, packet loss), is sufficiently accurate to guarantee a satisfactory network performance.

3.2.1 The admission control algorithm

Let us indicate with T_{ref} an arbitrary reference period. The basic principle of the proposed algorithm is to verify whether in a T_{ref} the medium occupancy time T_{occ} of the offered traffic (including collisions, retransmissions, etc.) keeps below T_{ref} itself. Upon the admission request of a new flow, the algorithm evaluates T_{occ} in the hypothesis of acceptance of the incoming flow. The flow is actually admitted to the service only if $T_{occ} \leq T_{ref}$.

T_{occ} is thus the key element of our algorithm. Its calculation bases on the following considerations. During a T_{ref}, the j^{th} flow offers frames at rate λ_j, for a total number of offered frames $\lambda_j \cdot T_{ref}$. The transmission of each of these frames occupies the channel for a time whose mean $E[T_j]$ can be analytically evaluated. $E[T_j]$ will also include all the overhead related to backoff, retries, etc. Thus the a.c. scheme can compute T_{occ} as:

$$T_{occ} = \sum_{j=1}^{M} \lambda_j T_{ref} E[T_j], \tag{3.1}$$

with M being the total number of flows[1].

[1]Let us sort the flows so that values of j from 1 to $M-1$ index existing flows, and $j = M$ indexes the incoming flow.

Estimation of sufficiently realistic and accurate $E[T_j]$ is the essential part of the a.c. scheme. In particular, we are interested in measuring $E[T_j]$ from the channel point of view. Not always does this correspond to the time as seen by each single station, and consequently computing T_{occ} as the sum of all the individual station-measured contributions will not be correct. Specifically, it will result in a gross overestimate of T_{occ}. This happens because there is no difference, from the channel point of view, whether the medium is occupied by one or more than one station, since the channel sees this event just as "the medium is busy".

For example, if we consider a collision, two (or more) stations do actually transmit their frame at the same time, hence occupying the channel for virtually half (or one third, fourth, etc.) of the time each. Similarly, when two or more stations are contending for the channel, the time spent in backing off is a shared virtual "occupancy" of the medium, as the backoff counter is decremented for all the contending stations.

To account for this point, the number of stations that cause those events should be included in the computation of $E[T_j]$. We actually do that by means of two parameters, α and β. We indicate with α the mean number of colliding stations in a generic time slot, given that a collision occurred, and with β the average number of stations competing to access the medium to transmit a queued frame. Both α and β can be estimated using the EDCA model developed in [15].

It is now possible to define a model, in Figure 3.4, to evaluate $E[T_j]$. Let us introduce the index i, which designates the Access Category (AC) to which the j^{th} flow is mapped, and let p_i be the probability of unsuccessful transmission for a frame belonging to AC_i. As depicted in Figure 3.4, $E[T_j]$ depends on p_i and the values $E[T_j(k)]$, which are the mean medium occupancy time of a frame transmission (of flow j and AC i) that requires

exactly k retransmissions[1].

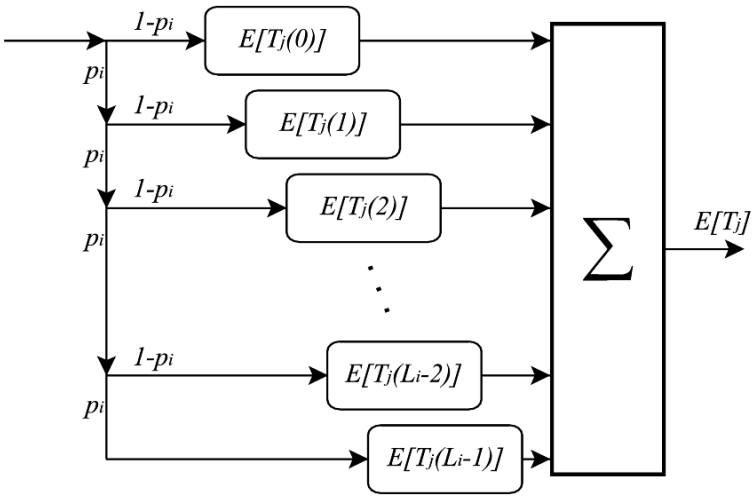


Figure 3.4: Model for the evaluation of $E[T_j]$.

The terms $E[T_j(k)]$ can be evaluated making reference to the single transmissions. We recall that a frame transmission attempt is composed of a sequence of periods. There is at first an inter-frame space (IFS), then the proper frame transmission (including PHY and MAC headers), possibly preceded by a backoff interval (e.g. due to contention). In case of correct reception, there is a SIFS followed by an ACK; in case of collision[2], the medium can be assumed to be idle immediately after the end of the longest frame. In both cases, a backoff procedure concludes the transmission cycle.

In the following, $B_{i,k}$ is the average backoff time of AC_i at the k^{th} backoff stage. In particular, $B_{i,k}$ can be expressed as $T_{slot}W_{i,k}/2$, being T_{slot} the basic IEEE 802.11 time slot and $W_{i,k}$ the contention window (for AC_i at the k^{th} backoff stage). $E[T_j(k)]$ is obtained from the mean medium occupancy

[1]To keep the notation simple, we omit the index i in $E[T_j]$ and $E[T_j(k)]$; yet their dependence on i is implicit in the mapping of the flow j to AC_i.

[2]We assume that collisions are the sole source of transmission errors.

time in case of a successful transmission $E[T_{succ,j}(k)]$ and in case of collision $E[T_{coll,j}(k)]$. Their expressions are:

$$E[T_{succ,j}(k)] = AIFS[i] + T_{PHY} + T_{MAC} + T_{DATA,j} + SIFS + T_{ACK} + \frac{B_{i,0}}{\beta}$$

$$E[T_{coll,j}(k)] = AIFS[i] + T_{PHY} + T_{MAC} + T^*_{DATA} + T_{ACK\_Timeout} + \frac{B_{i,k+1}}{\beta}$$

Here, $AIFS[i]$ is the Arbitration IFS of AC_i, T_{PHY} and T_{MAC} are the durations of the physical and MAC headers, $T_{DATA,j}$ is the time to transmit the payload (MSDU) of flow j, T_{ACK} is the time to transmit the ACK, and $T_{ACK\_Timeout}$ is the timeout for the reception of the ACK. In particular, $T_{DATA,j}$ can be expressed as D_j/R, where D_j is the MSDU size and R the transmission rate, assumed to be constant in the absence of rate adaptation algorithms. In case of collision, we should consider the time related to the longest collided data frame (T^*_{DATA}). As already outlined, β accounts for the number of stations that are doing backoff.

$E[T_j(k)]$ can then be evaluated for $k = 0 \dots L_i - 1$ (being L_i the retry limit for AC i) according to the following expressions:

$$E[T_j(0)] = E[T_{succ,j}(0)]$$

$$E[T_j(1)] = \frac{E[T_{coll,j}(0)]}{\alpha} + E[T_{succ,j}(1)]$$

$$E[T_j(2)] = \frac{E[T_{coll,j}(0)] + E[T_{coll,j}(1)]}{\alpha} + E[T_{succ,j}(2)]$$

$$\dots$$

$$E[T_j(L_i - 1)] =$$

$$= \frac{\sum_{k=0}^{L_i-2} E[T_{coll,j}(k)]}{\alpha} + p_i \frac{E[T_{coll,j}(L_i - 1)]}{\alpha} + (1 - p_i) E[T_{succ,j}(L_i - 1)]$$

Given these and the scheme in Figure 3.4, it is finally easy to obtain $E[T_j]$:

$$E[T_j] = \frac{p_i B_{i,0}}{\beta} + \sum_{k=0}^{L_i-1} (1 - p_i)^{1-\delta(k-L_i+1)} p_i^k E[T_j(k)], \tag{3.2}$$

where the function $\delta(\cdot)$ is 1 where its argument is zero and 0 otherwise. The term $B_{i,0}/\beta$ accounts for the extra backoff that must be performed when a frame arrives at an idle station and the medium is busy. As explained in [15], the probability of this event is again p_i.

3.2.2 The reference model

The probability p_i, which is at the basis of most of the previous formulae, can be computed using any analytical model of the 802.11e EDCA. Among the many existing, we have chosen the model proposed by Engelstad and Østerbø in [15]. This work basically improves [14], which in turn extended [4] to the 802.11e. One major feature is the accounting for a non-saturated channel, which makes the model much closer to reality and much more useful for application to QoS-sensitive services. As explained in Section 3.1, a saturated network based on a distributed access control (DCF and EDCA), though it may reach the highest throughput, is not able to transport real-time data with any satisfactory level of QoS. So the optimal work-point is well below the saturation state, hence the utility of a model for a non-saturated system.

Engelstad-Østerbø's model introduces the utilization factor ρ_i, which is tied to the probability that there is a frame in the transmission queue of AC_i at the time of a completed transmission. ρ_i can thus be used to account for the non-saturated network. The authors build a Markov chain in which the states are identified by the AC, the retransmission attempt, the backoff stage and the state of the transmission queue (either empty or not). Without entering into the many details of the solution of the chain (see [15] and also [44]), we just report the final results.

At each station, the probability τ_i of a transmission attempt in a generic

time slot for the i^{th} AC is given by:

$$\frac{1}{\tau_i} = \frac{1-2p_i^*}{2(1-p_i^*)} + \frac{W_{i,0}(1-p_i)(1-(2p_i)^{m_i})}{2(1-p_i^*)(1-2p_i)(1-p_i^{L_i+1})} + \frac{1-p_i}{1-p_i^{L_i+1}}\frac{1-\rho_i}{q_i}(1+\frac{(W_{i,0}-1)p_iq_i}{2(1-p_i^*)}) + W_{i,0}\frac{(2p_i)^{m_i}(1-p_i^{L_i-m_i+1})}{2(1-p_i^*)(1-p_i^{L_i+1})} \tag{3.3}$$

It can be noted that the transmission probability depends on several parameters. Beyond the already defined ρ_i and p_i, p_i^* is the probability that the backoff counter is not decremented (i.e. the channel is sensed busy), q_i is the probability that at least one frame arrives in the transmission queue during the following time slot under the condition that the queue is empty, q_i^* is the probability that a frame arrives while the backoff is frozen, $W_{i,0}$ is the initial contention window, and m_i is the backoff stage at which the contention window has reached its maximum. The probabilities can be evaluated with the following expressions:

$$p_b = 1 - \prod_{i=0}^{C-1}(1-\tau_i)^{n_i} \tag{3.4}$$

$$p_s = \sum_{i=0}^{C-1} n_i(1-p_i)\tau_i \tag{3.5}$$

$$p_i = 1 - \frac{1-p_b}{\prod_{c=0}^{i}(1-\tau_c)} \tag{3.6}$$

$$p_i^* = min(1, p_i + \frac{A_i p_b}{1-\tau_i}) \tag{3.7}$$

$$q_i = 1 - (p_s e^{-\lambda_i T_s} + (1-p_b)e^{-\lambda_i T_e} + (p_b - p_s)e^{-\lambda_i T_c}) \tag{3.8}$$

$$\rho_i = \lambda_i \overline{s}_i \tag{3.9}$$

In the formulas: n_i is the number of stations contending for the channel for each AC; C is the total number of ACs (usually four); $A_i = AIFSN[i] - AIFSN[C]$[1]; T_e, T_s and T_c denote respectively the real duration of an empty slot, of a slot containing a successfully transmitted frame and of a slot containing two or more colliding frames; p_b represents the probability that the channel is busy; p_s is the probability that a time slot contains a successful transmission. Eq. (3.6) refers to the case of virtual collision handling; however, if we assume that no more than one flow is generated at each station, a simpler form may be found. Furthermore, we can assume $q_i^* = q_i$ (according to [15] this is a good approximation), where q_i has been computed in the hypothesis of Poissonian traffic. Finally, the queue utilization factor ρ_i depends on the mean frame service time $\overline{s}_i$ once it has reached the front of the transmission queue. The derivation of $\overline{s}_i$ is rather complex, and therefore it has been skipped; the interested reader can refer to [44].

3.2.3 Application to a VoIP scenario

We consider a network with a variable number of associated stations, each sustaining a bidirectional VoIP call with a corresponding peer on the wired network. An 802.11e Access Point is the gateway between the wired and the wireless worlds. Voice is the only type of traffic in the network, and all stations use the same constant bit rate (CBR) codec.

This is undoubtedly a quite trivial scenario, but it is a good starting point to verify the proposed scheme, as the application is straightforward and allows a number of simplifications. All the flows have identical features (same frame rate, payload size, etc.) and belong to the same AC, hence

[1] Let's order the ACs so that AC_0 has the lowest priority and AC_C the highest. $AIFSN[C]$ is thus the smallest AIFSN.

index i can be removed. The direct consequence of this is that $p^* = p$, since A_i becomes zero. Then, there is only one flow per station, so that the internal collisions are eliminated and the series in (3.4), (3.5), and (3.6) can be reduced to only one term. So (3.4)–(3.7) are now (n_i is replaced by the total number of stations N):

$$p_b = 1 - (1 - \tau)^N \tag{3.10}$$

$$p_s = N\tau(1 - p) \tag{3.11}$$

$$p = p^* = 1 - (1 - \tau)^{N-1} \tag{3.12}$$

Eq. (3.8) can be computed exploiting the CBR nature of the traffic. In detail, the terms $e^{-\lambda t}$, which denote the probability that the inter-arrival time is greater than t for a Poisson process, can be replaced by their equivalents for a process with periodic arrivals. We can reasonably assume that the average T_s, T_e and T_c are all smaller than T_{frame} (otherwise there will not be room even for a single flow) and that these events occur with the same probability across the whole frame arrival period (the probability distribution is therefore uniform). Hence:

$$q = q^* = 1 - \frac{p_s T_s + (1 - p_b)T_e + (p_b - p_s)T_c}{T_{frame}} \tag{3.13}$$

Finally:

$$\rho = \lambda \overline{s} \tag{3.14}$$

It is worth noting that $\overline{s}$ can be computed using (3.2) where α and β have been removed (from the definition, $\overline{s}$ measures the service time at each single station, hence it does not depend on medium sharing parameters).

Together with (3.2) and (3.3), (3.10)-(3.14) form a system of equations that can be solved numerically, once the values for the voice traffic AC have been substituted. The value of $E[T]$ is then fed to (3.1) to allow for the verification of the admissibility of the new call.

The last step to complete the computation of T_{occ} is finding an estimate of α and β. α can be computed using the probability that k stations transmit given that a collision occurs (i.e. $k \geq 2$). Indicating with P_k the probability that in a generic time slot there are exactly k transmitting stations (and $N-k$ silent stations), we get:

$$\alpha = \frac{\sum_{k=2}^{N} kP_k}{\sum_{k=2}^{N} P_k} = \frac{\sum_{k=2}^{N} k\binom{N}{k}\tau^k(1-\tau)^{N-k}}{1-(1-\tau)^N - p_s} \tag{3.15}$$

Obviously, α is a function of the transmission probability τ. As for β, the average number of stations competing to transmit can be derived directly from the utilization factor ρ:

$$\beta = \rho N. \tag{3.16}$$

Once the simplifications are made, we can compare the indication given by our a.c. scheme to the outcome of some simulations. In this context, it is convenient to choose $T_{ref} = T_{frame}$, so that the criterion for the a.c. algorithm simply becomes checking that each flow has the time to transmit one frame in each frame generation period of the codec.

We consider a network with a variable number of associated stations, each sustaining a bidirectional VoIP call with a corresponding peer on the wired network. The scenario is very similar to the one in Figure 3.1, where VoIP peers only are present. All the flows are mapped to the highest priority AC, namely AC_VO, whichs pecifies the following parameters: $W_0 = 7$, $m = 1$, $L = 4$. The number of active calls has been increased from one

to system saturation, where no QoS guarantee is possible. Note that admitting a call requires that two flows (one in the uplink and one in the downlink) are admitted.

Two codecs have been used in two different series of tests. The first is the ITU-T G.723.1, with codec framing time T_{frame} of 30 ms and a payload D of 192 bit, leading to a net bit rate of 6.3 Kbps and an IP throughput of 17.2 Kbps. The other codec is the ITU-T G.729: it has $T_{frame} = 20$ ms and $D = 160$ bit, thus offering a net bit rate of 8 Kbps and an IP throughput of about 24 Kbps. In both cases no silence suppression is used, hence all sources generate data at constant bit rate (CBR).

To verify whether the a.c. scheme provides an accurate prediction of the maximum number of allowable calls with a satisfactory speech quality, we have set up our simulations in order to evaluate the R-factor for each call [30], following the same approach as in Section 2.2.4.1.

We performed the simulations with OPNET Modeler [45]. In each simulation run we varied the number of calls in the network and measured the R-factor of the worst call. All the results have been averaged over ten runs. Tables 3.4 and 3.5 reports the outcome for the G.729 and G.723.1 codecs respectively. n is the number of calls; R is the R-factor, computed from the simulation data following the rules given in [30]; $E[T]$ and T_{occ} (both in ms) were derived using the presented formulae.

Table 3.4: Simulation and a.c. results for G.729 ($T_{frame} = 20ms$)

n	R	$E[T]$	T_{occ}
11	83.2	0.849	18.68
12	82.9	0.838	20.11
13	12.0	0.829	21.55

Table 3.5: Simulation and a.c. results for G.723.1 ($T_{frame} = 30ms$)

n	R	$E[T]$	T_{occ}
16	78.2	0.860	27.50
17	74.9	0.851	28.91
18	23.7	0.842	30.32

The R-factor in Table 3.4 suggests that up to twelve G.729-based calls can be admitted with a satisfactory level of service. The model instead would have rejected the twelfth call, as T_{occ} results greater than T_{frame} (20 ms). However, the error is very small (less than 1%), and can be ascribed to the approximations necessary to make the model manageable. An even better result has been registered with G.723.1 (see Table 3.5). The a.c. scheme agrees with the results of the simulation, accepting the 17^{th} call, for which the R-factor is still above the target level, and rejecting the 18^{th}.

3.3 Measure-Based Admission Control

In the present Section we describe a second time-based admission control scheme that we have devised and tested. The scheme builds on an existing work by Garg and Kappes [46], who defined a parameter, called Network Utilization Characteristic (NUC), to measure the temporal occupancy of the channel of a legacy 802.11b network. Using a simple Markov chain, we have extended the application of this parameter to IEEE 802.11e systems and we have then built our a.c. scheme on the extended NUC. The effectiveness of the resulting scheme has been tested through simulations in presence of real-time voice and video traffic.

3.3.1 Overview of the NUC

The Network Utilization Characteristic, or NUC, is used to assess the evolution of the network in terms of temporal occupancy of the wireless channel. It is defined as the fraction of time per time unit needed to transmit a flow over the network [46]. Hence, NUC is measured on a per-flow basis. The NUC of each flow can range between 0 and 1, and summing up the NUCs of all flows yields the fraction of time the network is busy (NUC_{total}). Among all the flows, the so-called "auxiliary flows" must also be included. These account for activities, such as management frames, erroneous transmissions and collisions, that cannot be accredited to any particular flow and represent wasted capacity. With regard to admission control, a new flow can be accommodated without sacrificing other flows if its NUC is going to be smaller than the remaining capacity, which is the difference between one and NUC_{total}.

A method to compute the NUC for the basic IEEE 802.11 standard is also given in [46]. In particular, NUC is measured at the Access Point (AP) of an infrastructured 802.11b network, with the RTS/CTS mechanism disabled (the extension is trivial). The parameters needed for the computation of the NUC of a flow are the number of frames sent per second and the average transmission time of a frame sequence (both referred to that same flow). The latter consists of a series of periods: the time to transmit the payload, including MAC and physical layer overheads, plus the time to receive the acknowledgement. In addition, the backoff time must also be considered (see Figure 3.5, top). However, as our perspective is the usage of the wireless medium, we are not interested in the whole backoff window chosen for the last transmitted frame, but just in the number of idle slots seen by the channel, i.e. only those immediately preceding the transmission. In other words, we are implicitly assigning the slots of suspended

backoff periods, that elapse concurrently for all the stations backing off, only to the station that actually captures the channel. This number, multiplied by the slot time, yields the desired value. So, NUC can be computed with this expression:

$$n \cdot (\text{DIFS} + s \cdot \text{SLOT} + 2 \cdot H_{PHY} + b/R_{avg} + \text{SIFS} + \text{ACK}/R_{ACK} \quad (3.17)$$

In the formula, n is the number of sent frames, s is the average number of backoff slots waited right before transmission, b is the average MAC payload size in bit, R_{avg} is the average transmission bit rate, R_{ACK} is the bit rate used for the ACK frame, and H_{PHY} is the duration of the physical header. DIFS, SIFS, and SLOT are the time elements defined by the standard; ACK is the length of the acknowledgement frame (without physical header). The resulting value is expressed in μs. We can obtain the NUC dividing by the time unit (e.g. 1 second).

Though n, b, R_{avg}, and R_{ACK} can be easily obtained (e.g. observing the transmitted frames), determining s on a per-frame basis is not possible for anyone but the station transmitting the frame. However, since we are interested in the average value, and due to the fair nature of DCF, s can be assumed to be the same for all stations. So, the average number of backoff slots can be measured at the AP and the same value used for all stations. In [47], the authors highlight that s depends, in a non-linear way, on the number of active stations in the network. They also argue that, for an 802.11b network supporting VoIP services, a good approximation is $s = 8.5$ slots.

3.3.2 Extending the NUC to 802.11e

We have seen in Section 1.1.1 that the 802.11e amendment introduces several new features. Some of these (e.g. AIFS, TXOP) influence the computation of the NUC. Eq. (3.17) must therefore be adjusted to meet these

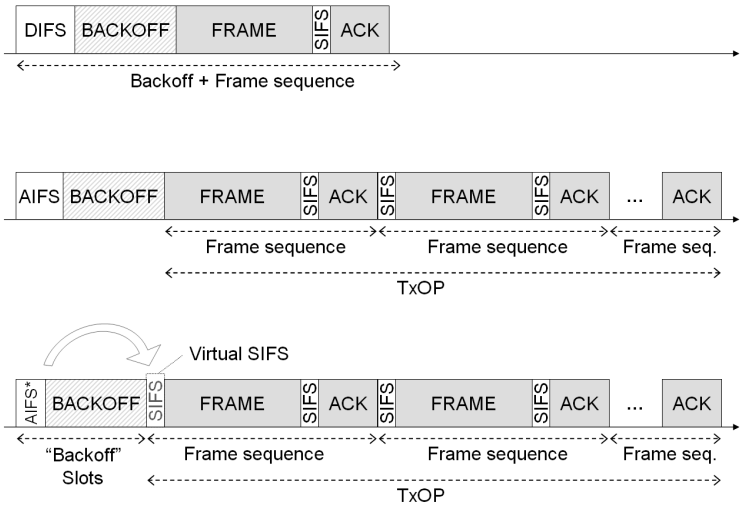


Figure 3.5: Frame sequences in IEEE 802.11 (top), 802.11e (centre) and our virtual rearrangement of 802.11e (bottom).

changes. It should account for the single initial AIFS and backoff slots, the multiple frames and related ACKs, and the SIFS separating each frame sequence from the following (see Figure 3.5, centre).

To make things simpler, we can rearrange the frame sequence as depicted in Figure 3.5, bottom. We assume that each frame sequence is composed by a SIFS, the data frame, another SIFS and the ACK. The first SIFS actually does not conform to the standard, but we can borrow it from the AIFS preceding the whole TXOP. So, the AIFS is virtually shortened. From its definition, what remains is AIFSN $\cdot$ SLOT. Let us call this quantity AIFS*. This rearrangement, which obviously is only virtual, allows us to simplify many computations. In such a way, (3.17) can be re-written as follows:

$$n/x \cdot s \cdot \mathrm{SLOT} + n \cdot (2 \cdot \mathrm{SIFS} + 2 \cdot H_{PHY} + b/R_{avg} + \mathrm{ACK}/R_{ACK} \quad (3.18)$$

We split the formula in two parts. The first accounts for the slots preceding each TXOP. x is the average number of frames per TXOP; s, beyond counting the number of backoff slots immediately preceding the transmission of the TXOP, also includes the AIFS*. The second part simply measures the time to transmit all data frames in a TXOP, including the virtual SIFS.

In (3.18), as in (3.17), all the parameters can be measured (or known in advance) except for s. However, due to the different modes of operation of the two versions of 802.11 (base and "e"), we cannot assume that the approximation suggested in [47] still holds. Hence we must find an estimate for this value. Simulations and analytical models are two possible methods. Yet, the former does not offer a general formula, and the latter may become excessively complex (see e.g. [15]) and/or neglect the possibility of multiple transmissions in the TXOP [48]. So, we have resolved to use a discrete-time Markov chain, which may be a rather simple tool and offer a general formula at the same time.

The resulting Markov chain is reported in Figure 3.6. The chain represents the evolution of frame transmissions once the station has obtained the right to transmit. Each state represents a frame of the current TXOP. m is the maximum allowed number of frames in a TXOP, and, in saturation conditions, it is also the length of the chain. A state transition occurs after the transmission of each frame sequence, which may be either successful (with probability $1 - p$) or not (p). A transmission fails if a correct ACK is not received. We assume that after each failure a new TXOP is started (when the station gains access to the channel).

Each transition is indicated with an arrow and associated to a couple of values (e.g. p/A). The first is the transition probability (as explained before) and the second is number of slots to be waited, as specified by the 802.11 standard. No slot, apart from the SIFS, shall pass between

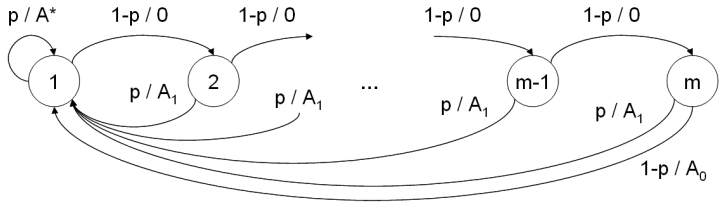


Figure 3.6: The Markov chain for the computation of the average number of backoff slots

two frames of the same TXOP. In all other cases the number of slots is determined by CW. Note that, following to our definition of the frame sequence (see again Figure 3.5), these slots are not pure backoff slots, but they also comprise the AIFS*. From a formal point of view this can be done by introducing the terms A_0, A_1 and A^* in place of the different CWs plus AIFS*. So, A_0 indicates that CW is reset to its minimum value CW_{min}. This happens after each successful frame transmission. A_1 marks the first increased value, i.e. $2 \cdot (\mathrm{CW}_{min} + 1) - 1$. Finally, A^* does not assume a constant value, but denotes all the increases of CW, due to retransmissions, up to CW_{max}. This point will become clearer when expanding A_0, A_1 and A^* for a practical case (see next Section).

The Markov chain can be solved for the steady-state probability vector $\underline{P}$ using the balance and the normalization equations (the chain is clearly

ergodic):

$$\begin{aligned}
(1) &: (1-p)P_1 = (1-p)P_m + p(P_2 + P_3 + \cdots + P_m) \\
(2) &: P_2 = (1-p)P_1 \\
(3) &: P_3 = (1-p)P_2 \\
&\cdots \\
(m-1) &: P_{m-1} = (1-p)P_{m-2} \\
norm &: P_1 + P_2 + \cdots + P_m = 1
\end{aligned}$$

Solving this system, we obtain:

$$P_i = \frac{(1-p)^i}{\sum_{j=0}^{m-1}(1-p)^j} \tag{3.19}$$

Now, we can find s as a function of the terms A_0, A_1 and A^*:

$$s = 0 \cdot p\{s=0\} + A_0 \cdot p\{s=A_0\} + A_1 \cdot p\{s=A_1\} + A^* \cdot p\{s=A^*\} \tag{3.20}$$

The terms $p\{s = \dots\}$ are the probabilities that s assumes the value in the brackets. They can be calculated from the Markov chain (we skip $p\{s=0\}$, being useless in (3.20)):

$$\begin{aligned}
p\{s=A_0\} &= (1-p)P_m \\
p\{s=A_1\} &= p(P_2 + P_3 + \cdots + P_m) \\
p\{s=A^*\} &= p \cdot P_1
\end{aligned}$$

Once substituted in (3.20), we can find s as a function of p. The probability of unsuccessful transmission p may be obtained in several ways (e.g. analytically, through simulations). A very simple method is the following.

Let k be the number of frames that the MAC layer has to transmit in the time unit, n the number of frames actually sent over the air, and L

the retry limit for a given AC. At first, following to unsuccessful transmissions, a fraction p of the k frames will be subject to retransmissions. Of these, another p^{th} part will be retransmitted, and so on, until L is reached. Therefore:

$$n = k + pk + p(pk) + \cdots + p^{L-1}k = \sum_{j=0}^{L-1} p^j k \tag{3.21}$$

This (L-1)th grade equation gives p as a function of n and k, whose numerical values can be easily drawn from simulations or from real network measurements. The last parameter to determine is m, the maximum number of frames that can be fitted in a TXOP. Given the length of the TXOP and the MAC payload to be carried (b), m is just the integer part of the division:

$$m = \left\lfloor \frac{\text{TXOP} + \text{SIFS}}{2 \cdot H_{PHY} + 2 \cdot \text{SIFS} + b/R_{avg} + \text{ACK}/R_{ACK}} \right\rfloor \tag{3.22}$$

where R_{avg} coincides with the maximum data rate. The SIFSs we added at both the numerator and the denominator are used to simplify the writing of the formula, in accordance to our rearrangement of the frame sequences.

3.3.3 Application to voice and video

To verify the goodness of the presented method, we applied it to a scenario that may represent a practical case. In particular we focused on an IEEE 802.11e network with voice and video traffic. Once the kind of traffic is specified, we can associate it to the proper ACs and hence obtain the numerical values of m, A_0, A_1, and A^* by replacing the values of AIFS, CW and TXOP in the preceding formulae. Real-time audio and video flows are mapped respectively to AC_VO and AC_VI. The values of the 802.11e parameters are reported in Table 3.6. For the categories AC_VO and AC_VI the contention window has only two possible values, CW_{min} and CW_{max}.

Table 3.6: Default EDCA parameters

Access Category	AIFS	TXOP	CWmin	CWmax
AC_VO	50 ms	6016 ms	7	15
AC_VI	50 ms	3264 ms	15	31

This implies $A^* = A_1$, since no further increase is possible beyond A_1. As a consequence, (3.20) becomes:

$$s = A_0 \cdot (1-p) \cdot P_m + A_1 \cdot p \cdot (P_1 + P_2 + \cdots + P_m) \tag{3.23}$$

Then, given the values of the 802.11e parameters, A_0 and A_1 assume the values reported in Table 3.7 (the unit is a SLOT, i.e. 20 μs). In the computation, two time slots (the remainder of AIFS - SIFS) have been added to the mean value of the backoff counter for the specified CW.

To determine m, we also need to know the payload of the frames, which in turn depends on the codec. In our simulations we adopted G.729 for the audio streams and H.261 for the video data. The first produces 160 bit of payload every 20 ms. For the video we used a trace of data generated by a video conference based on the H.261 codec. The output is actually a variable bit rate stream, with average payload of 7344 bit. Both the audio and video payloads must be corrected by adding the RTP/UDP/IP and MAC headers (respectively, 320 and 272 bit). So, assuming $R_{avg} = 11$Mbps and $R_{ACK} = 1$Mbps, from (3.22) we get $m = 5$ for audio and $m = 4$ for video. Finally, in case a station transmits using the lowest bit rate (1 Mbps), we get $m = 2$ for both audio and video. Hence we should consider three Markov chains of 2, 4 and 5 states each.

3.3.3.1 Simulation results

The described scenario has been simulated with Opnet Modeler [45]. We run an extensive set of simulations, embracing voice-only and mixed voice-

Table 3.7: Model parameters applied to voice and video

Traffic	m	A_0	A_1
Voice at 11 Mbps	5	5.5	9.5
Video at 11 Mbps	4	9.5	17.5
Voice at 1 Mbps	2	5.5	9.5
Video at 1 Mbps	2	9.5	17.5

video traffic patterns. A variable number of stations runs bidirectional voice and video calls. We assumed that each station transmits and receives only one type of traffic (either voice or video). Each station is connected to a peer in the wired domain, where the connection is ended. An Access Point is the gateway to it. The stations can experience two possible channel states: good and bad. This is achieved by varying their distance from the AP and their transmission power. In case of poor link quality, the station lowers its physical bit-rate to increase the reliability of the communication. A scheme of the network is reported in Figure 3.7.

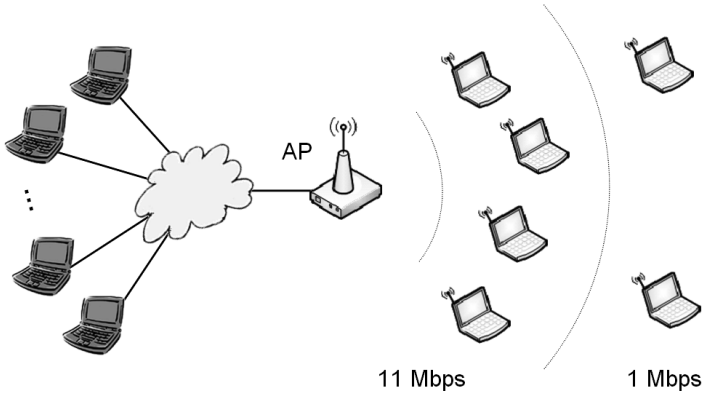


Figure 3.7: The network topology for the simulations

The admissibility of a call is based on the rating factor R, whose features

and use have already been explained in Section 2.2.4.1. Unfortunately, the R-factor is suitable only for voice calls. Therefore we had to refer to other parameters to asses the quality of video calls. We resolved to measure the more classic packet loss and end-to-end delay. For these we have set two thresholds that, once exceeded, denote that the quality of the video stream is no longer acceptable. These thresholds are 350 ms for the delay and 2% for the packet loss.

I. Voice calls only, all links are good. Table 3.8 reports the results for the simplest scenario, in which there is only voice traffic and all stations work in a noiseless channel at 11 Mbps. The first column refers to the number of voice stations in the network (N). NUC STA is the average NUC for the flows originating at the stations, while NUC AP refers to the sum of the flows transmitted by the Access Point. All NUC values are in per cent. The two R-factors are the worst among all stations in the uplink and downlink directions.

Both NUC_{total} and R indicates twelve as the capacity limit for the network. Some interesting remarks are the following. While the network is not heavily loaded, the NUC of the AP is roughly N times the average NUC of each station, but when the network starts congesting, the AP suffers more than the stations, thus confirming the well known problem of the bottleneck role of the AP. The same phenomenon can be observed through the R-factor. R for the uplink flows is smaller than the downlink for up to 13 stations. This is the effect of the different levels of contention for the AP and the stations. The AP has much more frames to send, hence it will try to access the medium more often. However, it has to compete with stations that transmit less frequently, and it can take advantage of this capturing the medium more easily. This situation holds until the network starts to be congested, when the traffic becomes too heavy and the bottleneck effect dominates thus making the downlink R-factor to collapse (also see

Table 3.8: Results for voice only, no slow links

N	NUC STA	NUC AP	NUC_{total}	R up	R down
11	3.13	34	68.4	81.5	82
12	3.47	39	80.6	80	81.5
13	5.17	46.5	113.7	62.5	79.5
14	7.24	44.8	146.1	44	0

Section 2.3). Back to the NUC, we can see that, if the network is not congested, the NUC of each station is scarcely dependent from the number of stations in the network. But, as the number of stations reaches the saturation, the NUC starts growing much faster. This growth is easily explainable with the increased number of retransmissions.

II. Voice calls only, some links are poor. We moved some of the stations far from the AP and set their transmission bit rate at 1 Mbps. This allowed to simulate the performance anomaly of 802.11 (see [13] and/or Chapter 1 for details) and to verify the efficacy of the NUC in a less ideal case. The results for one and four stations experiencing low quality links are reported in Tables3.9 and 3.10 (where NUC STA "ldr" and "hdr" refer to the stations transmitting at low data rate, i.e. 1 Mbps, and high data rate, i.e. 11 Mbps).

With one station in a bad position, the maximum number of allowable stations is still 12, so it is unchanged with regard to the ideal case. However a general increase in the NUC can be noted. The addition of the 13^{th} station provokes a steep rise in the NUC and drop in the R-factor. When the number of stations with poor link quality increases to four, the impact of the slow transmissions becomes stronger. The capacity is reduced by two stations. In both cases, NUC and R agree on the number of allowable voice calls.

Table 3.9: Results for voice only, one slow link

N	NUC STA hdr	NUC STA ldr	NUC AP	NUC_{total}	R up	R down
11	3.59	6.58	36	78.5	80.5	81.5
12	3.69	6.65	38.7	85.9	80	81.5
13	7.16	14.8	41.9	142.7	43	0

Table 3.10: Results for voice only, four slow links

N	NUC STA hdr	NUC STA ldr	NUC AP	NUC_{total}	R up	R down
9	3.61	7.29	29.4	76.6	80.5	82
10	4.01	8.71	34.4	93.3	76	81
11	6.61	13.7	34.3	135.8	47.5	0

III. Voice and video, all links are good. The presence of video stations unavoidably lowers the maximum capacity of the network. The outcome of the simulations with one video station and N voice stations in a noiseless channel is summarized in Table 3.11. We have divided the NUC of the audio and video flows, at both the stations and the AP. The R-factor is comprehensive of both uplink and downlink (the worst of the two). NUC suggests that at most 9 voice stations and one video station can coexist while offering an acceptable level of service. The analysis of the R-factor gives a slightly more pessimistic answer, because it is less than 70, so that the quality of this call cannot be considered satisfactory. The difference between NUC and R however is very small. Both parameters are very close to the admissibility threshold, thus confirming that the network works at a critical point.

As for the video, the delay is always under 50 ms, therefore it does not impede a satisfactory communication. Losses are well below 0.01 with

Table 3.11: Results for voice and video, no slow links

N	NUC STA		NUC AP		NUC_{total}	R
	Voice	Video	Voice	Video		
7	3.71	11.33	22.7	11	71	79
8	4.35	11.88	27.2	11.5	85.4	74
9	4.72	12.47	31.3	11.8	98	69.5
10	6.09	14.08	35.8	12.1	122.9	53

Table 3.12: Results for voice and video, one voice link is slow

N	NUC STA			NUC AP		NUC_{total}	R
	Vo.hdr	Vo.ldr	Video	Voice	Video		
7	3.84	8.12	11.53	23	11	76.8	78
8	4.55	8.49	11.96	27.3	11.5	91.1	72.5
9	5.62	11.52	13.54	32.2	12.1	114.3	57.5

up to eight voice stations, reach one per cent at nine stations, and then increase dramatically (roughly 14%), thus making the video connection unacceptably corrupt. Hence, these data confirm the indications from the NUC and R-factor.

IV. Voice and video, one voice link is poor. In this case, differently from what described in II, the presence of a single voice station working in sub-optimal conditions is enough to reduce the capacity of the network. As it can be seen in Table 3.12, one less voice station can be admitted with respect to the ideal case. Being all the configurations far from the critical work-point of the network, NUC and R now agree on the numbers. The data on video traffic is a further proof: packet loss is low (less than 1%) up to eight stations and becomes unbearable (6% and more) afterwards. Delay is always acceptable.

V. Voice calls only, one video link is poor. In the last test,

Table 3.13: Results for voice and video, the video link is slow

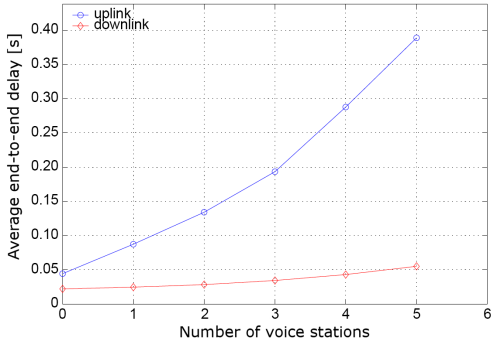
N	NUC STA		NUC AP		NUC_{total}	R
	Voice	Video	Voice	Video		
0	0	70.5	0	10.54	81.1	—
1	3.74	73.16	3.39	11.34	91.6	80.5
2	4.2	72.12	6.97	11.66	99.2	78.5
3	4.56	68.3	10.64	11.83	104.5	74.5
4	5.01	62.98	14.47	12.1	112.6	68

we placed the video station far from the AP, while all voice stations are close to it. The results are in Table 3.13 and Figure 3.8. No more than two bidirectional voice calls can be admitted if the NUC is our admission criterion. According to the R-factor, however, there is still capacity in the network for a third call. Furthermore, the analysis of packet loss suggests that even adding a single voice connection is enough to spoil the quality of the video, since the losses in the uplink direction jump to about 4%. So, in this scenario, the three control parameters (NUC, R-factor and losses) give different answers.

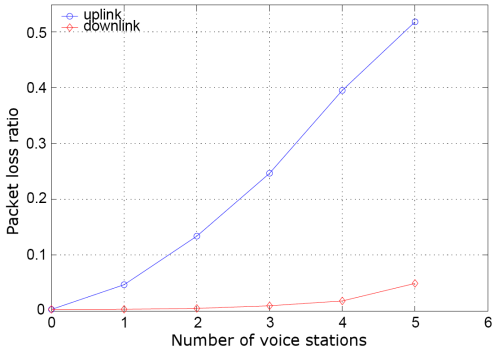
The explanation on the divergence of R and packet loss can be found in their different targets and in the nature of the access protocol. IEEE 802.11e differentiates the two kinds of frames, giving audio more chances to access the medium. The two measurement parameters, each addressing a single type of traffic, necessarily have a different perception of the network, with the R factor measuring a more favourable situation. NUC, embracing all traffic on the network, is in between, thus giving a more objective picture of the network state.

It is impressive to note that video traffic alone would consume more than 80% of the capacity. This amount increases slowly as we add voice calls, but as soon as the network gets saturated, it starts decreasing. On the

contrary, NUC of voice traffic continuously increases. More in detail, it is only the NUC of the video station that decreases, whereas the video part of the NUC at the AP stays roughly constant. This is actually straightforward and again springs from the purpose of the IEEE 802.11e standard, which offers voice higher priority in accessing the medium. Hence, when the network saturates, the video station is the first to suffer. The same occurs at the AP as well, but it is mitigated by the capture of the channel, as described in I.



(a) end-to-end delay



(b) packet loss

Figure 3.8: Average delay and packet loss of video frames for the mixed audio-video scenario with the video station working at low bit rate

Summarising, in most cases the NUC is sufficiently accurate to control the admission of new flows. This is also true for mixed audio and video traffic and in cases of non homogeneous transmission rates, even though a slight divergence has appeared when there are video stations transmitting at low bit rate. In this case, NUC, the R-factor and video packet losses indicates different numbers of admissible voice stations (from 0 to 3), but the NUC indeed shows the more objective perception of the network.

3.4 Conclusions

Determining the admission region of multimedia streams (VoIP and video-conference services) for the 802.11e EDCA mode has been the basis for the development of the following admission control schemes. The study on the admission region put in evidence the good differentiation capabilities of 802.11e as the presence of TCP traffic was not so hindering as it would have been for the basic 802.11 standard. On the other hand, it also showed the bottleneck role played by the AP and the scarce exploitation of network resources, since the admission region is exclusively bound by the downlink traffic. The bottleneck problem could be partly solved with the scheduler proposed in Section 2.3. The degradation springing from the excessive number of users is instead the focus of the proposed a.c. schemes.

As we have seen, the first a.c. method uses an analytical model of the 802.11e EDCA mode in non-saturation conditions, whereas the second builds on a parameter (the NUC) computed through measurements of the current state of the network. The two approaches are therefore almost antipodal but share the common principle of measuring channel occupancy in terms of time instead of bandwidth. In Section 2.1 we have exposed the theoretical foundations of the advantages of a time-based approach. The two proposed a.c. put these concepts into practice. The good performance

registered through the simulation of a network carrying VoIP and video services confirmed the effectiveness and feasibility of this approach.

Moreover, as for the first model, the use of a model accounting for non-saturation conditions allowed to get much closer to the real network behaviour than solutions based on commonly used saturation models. As for the second model, the chosen parameter turned out to be much more accurate in perceiving the state of the network in a mixed service environment than service-specialized metrics such as the R-factor, packet loss, etc.

Chapter 4

Conclusions

Though IEEE 802.11 is a very popular and widely adopted technology, it still presents several flaws which prevent it from fully exploiting the bandwidth offered by the ever increasing physical data rate. Two are the major issues encumbering the standard that we have taken into account for the work presented in this thesis.

The first problem is the so-called performance anomaly. In presence of variable link conditions, terminals working with different version of the standard (e.g. *b* or *g*), and automatic rate adaptation algorithms, the throughput of all the stations tends to align to that of the slowest one. Unfortunately this behaviour is inherent in the standard, being a direct consequence of its fairness philosophy. By the way, not even the new *e* amendment is immune from the anomaly.

We decided to face the problem at its root, changing the fairness objective of 802.11 MAC. We recognised that proportional fairness, striking a balance between extreme fairness and extreme bandwidth exploitation, could be an attractive design goal. So we designed a scheduler to put it into practice. We changed the design metric from bandwidth to time, since this is the way to achieve the intended goal. And we provided the

scheduler with multi-queue capability, in order to avoid the annoying flow inter-dependency caused by the commonly used FIFO queue. Finally, the scheduler was meant to be placed on top of the MAC layer, to keep the compatibility with the existing hardware, and below the network layer, to be transparent to existing software.

The resulting scheduler, named DTT (Deficit Transmission Time), was implemented and tested. A prototype AP equipped with DTT (the first working implementation of an algorithm actualising proportional fairness) provided excellent results in a small office network with mobile terminals and TCP traffic. The capability of supporting voice services was investigated via simulation, which revealed that DTT can considerably increase the capacity of the network (in terms of number of services transported with a satisfactory speech quality). A distributed version of the scheduler, $(DT)^2$ was shown to further improve this result, with, in addition, the possibility of reducing the bottleneck effect.

The second problem regards the support of real-time multimedia services over 802.11 WLANs. This topic has captured the eye of industry and academia researchers due to the increasingly market demand and the recent ratification of the *e* amendment. The newly introduced traffic differentiation strategy proved to be effective in delivering priority to voice and video frames, but do not offer any strict guarantee in terms of QoS parameters, which is required to properly transport this kind of service. A way to overcome this drawback is to set a limit to the number of users/services admitted to the transport facility. To this aim, the presence of an admission control (a.c.) mechanism is essential.

After evaluating the admission region of an 802.11e network, we have devised two a.c. schemes. Both exploits the experience matured in designing the DTT scheduler and adopted the time occupancy of the medium as the admission criterion. They differ in the method to estimate the available

channel time when deciding on the admission of a new service. One scheme follows an analytical approach, exploiting a model of the distributed access function (the EDCA) in non-saturation conditions. The other computes a parameter on the basis of live measurements on the current state of the network. We have tested both approaches via simulation, in voice and mixed voice/video scenarios, and proved that both works very well, keeping very close to the actual capacity bounds of the network (the admission region).

In conclusion, at the light of the work presented in this thesis, we can state that using time as the capacity metric of 802.11 networks is undoubtedly more simple, flexible and efficient than bandwidth. We proved this concept through the application to two major and apparently distant issues, performance anomaly and admission control, obtaining in both cases excellent results. In addition, the proposed solutions instantiates two remarkable advancements. DTT is the first working prototype actualising proportional fairness, and the scheme presented in Section 3.2 is the first a.c. tool to employ a non-saturation model.

As a final note, we wish to point out that the two solutions explored in the thesis (scheduling and a.c.) can easily and profitably complement each other, particularly in enhancing the support of multimedia services over 802.11e networks. We remind that the performance anomaly affects the whole range of 802.11 standards, including the *e* amendment. Since DTT works on top of the MAC layer, it can also be applied to this version. Furthermore it can natively help the MAC layer in delivering more appealing QoS figures by employing a customisable distribution of the transmission credits (or "water") to the different flows (see Section 2.2.2).

References

[1] *ANSI/IEEE Std 802.11, 1999 Edition. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std., Mar. 1999.

[2] *IEEE Std 802.11e-2005. Part 11, Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, IEEE Std., Nov. 2005.

[3] Q. Ni, L. Romdhani, and T. Turletti, "A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN," *Wiley Journal of Wireless Communication and Mobile Computing (JWCMC)*, vol. 4, no. 5, pp. 547–566, 2004.

[4] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[5] G. Bianchi and I. Tinnirello, "Remarks on IEEE 802.11 DCF Performance Analysis ," *IEEE Communications Letters*, vol. 9, no. 8, pp. 765–767, Aug. 2005.

[6] B. P. Crow, I. Widjaja, L. G. Kim, and P. T. Sakai, "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116–126, 1997.

[7] W. Pattara-Aukom, S. Banerjee, and P. Krishnamurthy, "Starvation prevention and quality of service in wireless LANs," in *5^{th} International Symposium on Wireless Personal Multimedia Communications*, Honolulu, Oct. 2002.

[8] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service," in *European Wireless*, Florence, Feb. 2002.

[9] S. Xu, "Advances in WLAN QoS for 802.11: an overview," in *14^{th} IEEE Personal, Indoor and Mobile Radio Communications (PIMRC)*, Beijing, Sept. 2003.

[10] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, "IEEE 802.11 Ad Hoc Networks: Performance Measurements," in *23^{rd} International Conference on Distributed Computing Systems*, Providence (USA), May 2003.

[11] S. Lucetti, R. G. Garroppo, and S. Giordano, "IEEE 802.11b performance evaluation: convergence of theoretical, simulation and experimental results," in *Networks*, Wien, 2004.

[12] A. Doufexi, S. Armour, B. Lee, A. Nix, and D. Bull, "An Evaluation of the Performance of IEEE 802.11a and 802.11g Wireless Local Area Networks in a Corporate Office Environment," in *IEEE International Conference on Communications (ICC)*, Anchorage, May 2003.

[13] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," in *IEEE Infocom*, San Francisco, Apr. 2003.

[14] Y. Xiao, "Performance analysis of IEEE 802.11e EDCF under saturation condition," in *IEEE International Conference on Communications (ICC)*, Paris, June 2004.

[15] P. E. Engelstad and O. N. Østerbø, "Delay and Throughput Analysis of IEEE 802.11e EDCA with Starvation Prediction," in *30th IEEE Conference on Local Computer Networks*, Sydney, Nov. 2005.

[16] S. Choi, J. del Prado, S. Shankar, and S. Mangold, "IEEE 802.11e Contention-Based Channel Access (EDCF) Performance Evaluation," in *IEEE International Conference on Communications (ICC)*, Anchorage, May 2003.

[17] D. He and C. Q. Shen, "Simulation study of IEEE 802.11e EDCF," in *57th IEEE Semiannual Vehicular Technology Conference (VTC)*, Jeju (Korea), Apr. 2003.

[18] G. Bianchi, I. Tinnirello, and L. Scalia, "Understanding 802.11e Contention-Based Prioritization Mechanisms and Their Coexistence with Legacy 802.11 Stations," *IEEE Network*, vol. 19, no. 4, pp. 28–34, July 2005.

[19] Y. Cao and V. Li, "Scheduling algorithms in broadband wireless networks," *Proceedings of the IEEE*, vol. 89, no. 1, pp. 76–87, Jan. 2001.

[20] J. Hartwell and A. Fapojuwo, "Modeling and Characterization of Frame Loss Process in IEEE 802.11 Wireless Local Area Networks," in *IEEE Vehicular Technology Conference (VTC) 2004-Fall*, Los Angeles, Sept. 2004.

[21] L. B. Jiang and S. C. Liew, "Proportional Fairness in Wireless LANs and Ad Hoc Networks," in *IEEE Wireless Communications and Network Conference (WCNC)*, New Orleans, Mar. 2005.

[22] M. Portoles, Z. Zhong, and S. Choi, "IEEE 802.11 Downlink Traffic Shaping Scheme For Multi-User Service Enhancement," in *IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Beijing, Sept. 2003.

[23] J. Mo and J. Walrand, "Fair End-to-End Window-Based Congestion Control," *IEEE/ACM Trans. Networking*, vol. 8, no. 5, pp. 556–567, Oct. 2000.

[24] B. Radunovic and J. L. Boudec, "Rate Performance Objectives of Multihop Wireless Networks," *IEEE Trans. on Mobile Computing*, vol. 3, no. 4, pp. 334–349, Oct. 2004.

[25] J. Malinen. Host AP driver for Intersil Prism2/2.5/3. [Online]. Available: http://hostap.epitest.fi

[26] R. G. Garroppo, S. Giordano, S. Lucetti, and L. Tavanti, "Providing Air Time Usage Fairness in 802.11 Networks with the Deficit Transmission Time (DTT) Scheduler," *Wireless Networks - Special Issue on Broadband Wireless Multimedia*, June 2006.

[27] The 4G AccessCube (aka "MeshCube"). [Online]. Available: http://www.meshcube.org

[28] The multi-generator (mgen). [Online]. Available: http://cs.itd.nrl.navy.mil/work/mgen

[29] M. Coupechoux, V. Kumar, and L. Brignol, "Voice over IEEE 802.11b Capacity," in *ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*, Antwerp, Aug. 2004.

[30] *"The E-model, a computational model for use in transmission planning"*, ITU-T Recommendation G.107, Mar. 2003.

[31] The omnet++ discrete event simulation system. [Online]. Available: http://www.omnetpp.org

[32] Mobility framework for omnet++. [Online]. Available: http://mobility-fw.sourceforge.net/hp/index.html

[33] R. G. Garroppo, S. Giordano, S. Lucetti, and F. Russo, "IEEE 802.11b Performance Evaluation: Convergence of Theoretical, Simulation and Experimental Results," in *Networks*, Wien, June 2004.

[34] D. D. Vleeschauwer and J. Janssen, "Voice Performance over packet-based networks, Alcatel White Paper," Oct. 2002.

[35] G. Bianchi and I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network," in *IEEE Infocom*, San Francisco, Apr. 2003.

[36] B. Dong and X. Wang, "Bayesian Monte Carlo Estimators of Competing Terminals in an Ad Hoc Wireless Network," in *38^{th} annual Conference on Information Sciences and Systems*, Princeton University (USA), Mar. 2004.

[37] R. G. Garroppo, S. Giordano, S. Lucetti, and L. Tavanti, "Admission Region of Multimedia Services for EDCA in IEEE 802.11e Access Networks," *Lecture Notes in Computer Science*, vol. 3427, pp. 105–120, 2005.

[38] F. T. D. Hole, "Capacity of an IEEE 802.11b wireless LAN supporting VoIP," in *IEEE International Conference on Communications (ICC)*, Paris, June 2004.

[39] X. Chen, H. Zhai, X. Tian, and Y. Fang, "Supporting QoS in IEEE 802.11e wireless LANs," *IEEE Transactions on Wireless Communications*, vol. 5, no. 8, pp. 2217–2227, Aug. 2006.

[40] *"IP Packet Transfer and Availability Performance Parameters"*, ITU-T Recommendation Y.1540, Dec. 2002.

[41] *"Network Performance Objectives for IP-Based Services"*, ITU-T Recommendation Y.1541, May 2002.

[42] The network simulator – ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns

[43] D. Gao, J. Cai, and K. N. Ngan, "Admission Control in IEEE 802.11eWireless LANs," *IEEE Network*, vol. 19, no. 4, pp. 6–13, July 2005.

[44] P. E. Engelstad and O. N. Østerbø, "The Delay Distribution of IEEE 802.11e EDCA and 802.11 DCF," in *25th IEEE International Performance, Computing, and Communications Conference*, Phoenix, Apr. 2006.

[45] Opnet modeler. [Online]. Available: http://www.opnet.com/products/modeler/

[46] S. Garg and M. Kappes, "On the Throughput of 802.11b Networks for VoIP," Avaya Labs Research, Tech. Rep. ALR-2002-012, Mar. 2002.

[47] ——, "A New Admission Control Metric for VoIP Traffic in 802.11 Networks," Avaya Labs Research, Tech. Rep. ALR-2002-021, May 2002.

[48] A. Banchs and L. Vollero, "A Delay Model for 802.11e EDCA," *IEEE Communications Letters*, vol. 9, no. 6, June 2005.