Improved performance of voice and data using the 802.11e EDCF

Peter Clifford, Ken Duffy, John Foy, Doug Leith, David Malone
Hamilton Institute, NUI Maynooth, Ireland

Abstract— **In 802.11 networks substantial unfairness can exist between competing voice and data traffic, and between competing upload/download data flows. In this paper we develop a new analytic model of 802.11e networks that is capable of capturing the behaviour of voice and data traffic and the impact of the 802.11e prioritisation mechanisms on network behaviour. Using the insight gained we propose a soundly-based strategy for selecting 802.11e MAC parameters in networks carrying mixed voice and data traffic.**

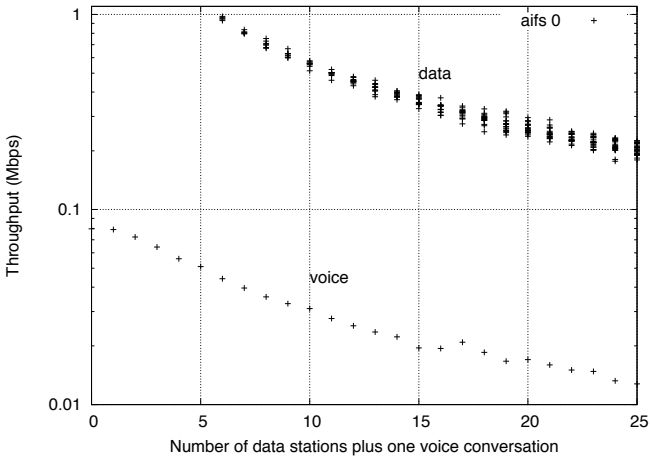
I. Introduction

In recent years, 802.11 wireless LANs have become pervasive. While providing wire-free connectivity at low cost, it is widely recognised that the 802.11 MAC layer requires greater flexibility and the new 802.11e standard consequently allows tuning of MAC parameters that have previously been constant. Although the 802.11e standard provides adjustable parameters within the MAC layer, the challenge is to use this flexibility to achieve enhanced network performance.

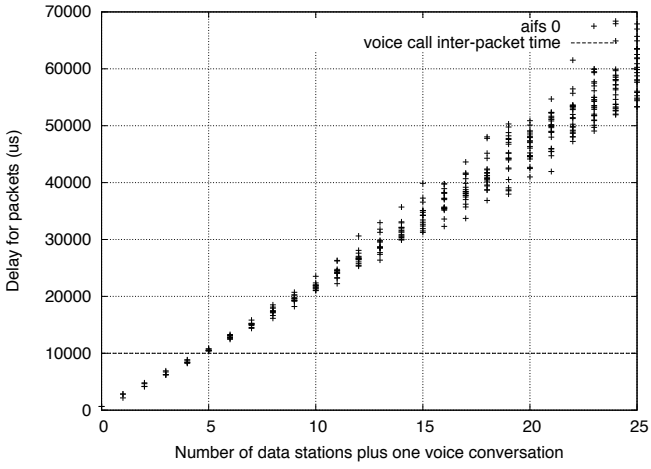
Existing work on 802.11e tuning algorithms is largely informed by the quality of service (QoS) requirements of newer applications such as voice over IP. However, network traffic is currently dominated by data traffic (web, email, media downloads, etc.) carried via the TCP reliable transport protocol and this situation is likely to continue for some time. A key question, therefore, is how can we use the flexibility provided by the 802.11e MAC to support a mixture of both voice and data traffic in a manner that provides the necessary QoS to both the voice and data traffic while also making efficient use of the channel capacity.

That the potential exists for negative interactions between voice and data traffic is readily verified. Indeed, even a relatively small amount of data traffic is sufficient to disrupt voice calls. This behaviour is illustrated, for example, in Figure 1 which plots throughput and delay for a single voice call as the number of competing FTP/TCP flows is increased (one flow per wireless station). It can be seen that TCP flows are able to seize bandwidth from the voice call so that, for example, approximately five competing TCP flows are sufficient to half the throughput achieved by the voice call. With five competing TCP flows, it can also be seen that the MAC delay experienced by the voice traffic exceeds the call inter-packet interval (marked by a solid line on the plot), and so this is already beyond the stable queueing regime.

This behaviour is associated with an unfairness in the 802.11 contention mechanism whereby greedy flows (that always have a packet to send) are able to seize bandwidth from low rate flows. The voice call has a peak rate of only 64Kbits/s, which



(a)



(b)

Fig. 1. Throughput and delay where one voice call competes with TCP stations. The MAC delay for the voice call and data transfers are similar, but the voice call throughput is reduced by approximately 50% in the presence of five competing TCP stations. (NS simulation, 802.11b MAC, G711 two-way voice call with silence suppression, voice call inter-packet spacing is 10ms - marked by the solid line on the delay plot.)

in this example is far below its "fair" share, i.e. the share of throughput that could potentially be obtained by a greedy flow in the same circumstances. We study this unfairness mechanism in more detail later. However, we note here that this issue is associated with MAC layer behaviour and so is perhaps most naturally addressed at the MAC layer.

Unfairness in current 802.11 networks is not confined to interactions between competing voice and data traffic. Cross-layer interactions between the 802.11 MAC and the flow/congestion control mechanisms employed by TCP typically lead to gross unfairness between competing flows, and indeed sustained lockout of flows. Figure 2 illustrates the behaviour of competing TCP upload flows over an 802.11b WLAN. Gross unfairness between the throughput achieved by competing flows is evident. Unfairness also exists between competing upload and download TCP flows. This is illustrated for example in Figure 3 where it can be seen that upload flows achieve nearly two orders of magnitude greater throughput than competing download flows. We note that, while lacking the time critical aspect of voice traffic, data traffic server-client applications do place significant quality of service demands on the wireless channel. In particular, within the context of infrastructure WLANs in office and commercial environments there is a real requirement for efficient and reasonably fair sharing of the wireless capacity between competing data flows.

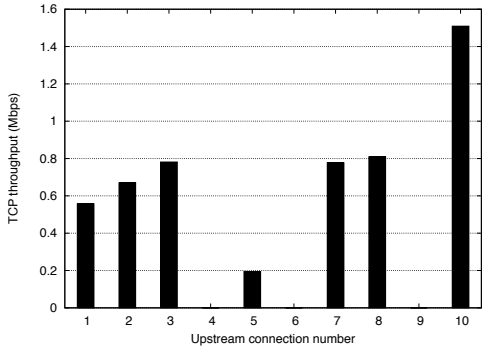


Fig. 2. Throughput of competing TCP uploads. (NS simulation, 10 upload TCP flows, single cell infrastructure mode 802.11b WLAN, TCP SACK variant.)

In this paper we investigate how we can use the flexibility provided by the new 802.11e MAC to resolve unfairness in infrastructure WLANs, with the aim of delivering acceptable quality of service to both voice and data traffic.

II. Related Work

There have been a number of previous studies of voice over 802.11 networks. Much of the work has been concerned with measuring the voice call capacity of 802.11 networks rather than operation with mixed voice/data traffic or adjustment the MAC layer behaviour itself. For example, in [9], a back-of-envelope calculation for maximum capacity of a WLAN is presented and shown to be a useful estimate. The authors also consider, using simulation, how delay constraints and bit error rates impact the capacity of the network. Other metrics for voice capacity are also used in, for example, [1], [7].

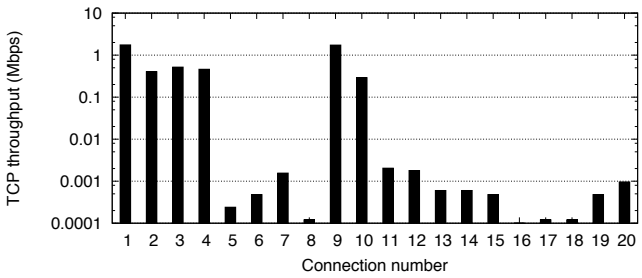


Fig. 3. Throughput of competing TCP uploads and downloads. Note the y-axis logscale. Flows 1-10 are uploads, flows 11-20 are downloads. (NS simulation, 10 upload TCP flows, 10 download TCP flows, infrastructure mode 802.11b WLAN, TCP SACK.)

Networks with mixed voice/data traffic are considered in [1] where an experimental study of the capacity of voice in an 802.11b network is performed. In [15] the implementation and validation of a priority queue scheme at the driver level above an 802.11b MAC is considered. Others have considered using the scheduled HCCF component (as opposed to the contention-based EDCF component considered here) of the 802.11e MAC for carrying voice and data, for example [6].

While the literature relating to WLAN fairness at the MAC layer is extensive, the issue of transport layer TCP fairness has received far less attention. Early work by Balakrishnan and Padmanabhan [2] studies the impact of path asymmetries in both wired and wireless networks, while more recently Detti et al. [8] and Pilosof et al. [12] have specifically considered TCP unfairness issues in 802.11 infrastructure WLANs and Wu et al. [14] study TCP in the context of single-hop 802.11 ad hoc WLAN's. With the exception of [14], all of these authors seek to work within the constraints of the basic 802.11 MAC and thus focus solely on approaches that avoid changes at the MAC layer.

III. Analytic Modelling

The 802.11 MAC layer uses a CSMA/CA algorithm with binary exponential back-off to regulate access to the shared wireless channel. Briefly, on detecting the wireless medium to be idle for a period $DIFS$, each station initializes a counter to a random number selected uniformly from the interval [0,CW-1]. Time is slotted and this counter is decremented each slot that the medium is idle. An important feature is that the countdown halts when the medium becomes busy and only resumes after the medium is idle again for a period $DIFS$. On the counter reaching zero, the station is permitted to transmit for a time $TXOP$ on the medium (defined to be one packet in 802.11). If a collision occurs (two or more stations transmit simultaneously), CW is doubled and the process repeated. On a successful transmission, CW is reset to the value CW_{min} and a new countdown starts for the next packet.

The new 802.11e MAC enables the values of $DIFS$ (called $AIFS$ in 802.11e), CW_{min} and $TXOP$ to be set on a per class basis for each station i.e. traffic is directed to up to four different queues at each station, with each queue assigned different MAC parameter values. (Note that the 802.11e standard specifies further MAC parameters in addition to $AIFS$, CW_{min} and $TXOP$ that may also be adjusted, but these are not considered here).

Following the seminal paper of Bianchi [4], much of the analytic work on 802.11 MAC performance has focused on saturated networks where each station always has a packet to send. In particular, recent work has extended the saturation modelling approach to include multi-class 802.11e networks, see [3], [13]. The saturation assumption is key to these models as it enables queueing dynamics to be neglected and avoids the need for detailed modelling of traffic characteristics, making these networks particularly tractable.

Networks do not typically operate in saturated conditions. Internet applications, such as web-browsing, e-mail and voice over IP exhibit bursty or on-off traffic characteristics. Creating an analytic model that includes fine detail of traffic-arrivals and queueing behavior, as well as 802.11 MAC operation, presents a significant challenge. In this paper we introduce an 802.11e EDCF model with traffic and buffering assumptions that make it sufficiently simple to give explicit expressions for the quantities of interest (throughput per station, delay, collision probabilities), but still capture key effects of non-saturated operation. Although our traffic assumptions form only a subset of the possible arrival processes, we will see they are useful in modelling a wide range of traffic, including voice conversations, TCP traffic and mixtures of both.

Details of our analytic model are contained in the Appendix - using specified per class 802.11e MAC parameters and per station arrival rates, the model predicts the per station transmission probability, collision probability and throughput. The predictive accuracy of the model is illustrated in Figure 4, where model predictions are compared with throughput data from NS packet-level simulations for a network with two traffic classes and a range of poisson traffic loads and $AIFS$ values. The accuracy of the model in predicting voice traffic behaviour is illustrated, for example, in Figures 5 and 6; the utility of the model for mixed voice and data traffic is demonstrated in Section VI.

IV. Voice calls only

Before proceeding to consider networks with both voice and data traffic, we first consider a network with voice only traffic. Our focus in this paper is on infrastructure mode networks where calls are routed through a single access point (AP). Owing to nature of the 802.11 contention mechanism, infrastructure networks might be expected to behave quite differently from ad hoc peer-to-peer networks.

Specifically, the 802.11 MAC enforces per station fairness, i.e. each station has approximately the same number of transmission opportunities. This includes not only the wireless stations, but also the AP itself. The conversations that we consider are two-way. That is, we have n wireless stations each transmitting the voice of one speaker and n replies transmitted by the AP. Hence, we might expect that the n wireless stations have roughly a $n/(n+1)$ share of the bandwidth while the AP has only a $1/(n+1)$ share. An obvious concern is that such an asymmetry would lead to a lower voice call capacity in an infrastructure mode network (compared to an ad hoc network) due to throttling of traffic through the AP.

Figure 5 shows the throughput and delay in an 802.11b network as the number of voice conversations is increased. Throughput and delay are shown both for the AP and the wireless stations. Throughout this paper we use G.711 voice calls with the parameters for the voice calls taken from [11]: 64kbs on-off traffic streams where the on and off periods are distributed with mean 1.5 seconds. Periods of less than 240ms are increased to 240ms in length, to reproduce the minimum talk-spurt period. Voice conversations are modelled as a two-way call with interleaved on-off periods. It can be seen that for less than about 10 voice conversations, the AP and the wireless stations achieve similar throughput and delay. For larger numbers of calls, the throughput achieved by the AP rapidly falls and losses rise. As noted previously, this throttling of the AP is to be expected given the per station fairness imposed by the 802.11 MAC.

We explore this further by noting that, using the flexibility provided by the 802.11e MAC, the AP can readily be prioritised to remove the asymmetry between it and the wireless stations. Specifically, we consider setting the AP $TXOP$ to be equal to the number of active downlink voice calls[1]. Figure 6 shows throughput and delay with this scheme as the number of voice calls is increased. It can seen that now both the AP and wireless stations achieve similar throughputs, as expected. The network can now sustain approximately 15 voice calls. For larger numbers of calls the MAC delay increases beyond the 10ms inter-packet arrival time of the voice calls; that is, the system enters an unstable queueing regime where delays and loss rapidly increase.

We observe that this voice call capacity is almost identical to that achieved in an ad hoc network where voice calls are between pairs of wireless stations rather than routed via the AP (space restrictions prevent us including the relevant ad hoc plots here). The capacity is also in good agreement with the following simple calculation. Table I gives the overhead budget for the transmission of a small packet of payload 80 bytes. Since $649\mu s$ are needed for the transmission of 80 bytes, the maximum possible user throughput is approximately 0.98Mbs. Hence, the channel can at best support no more than 15.4 64Kbs voice conversations. Note that this figure is overoptimistic as it neglects the idle time spent during contention window countdown as well as many other details of the channel behaviour (such as packet collisions). Nevertheless, we can see that the measured capacity of 15 calls is remarkably close to this ideal value[2] which indicates that the scope for further performance

[1]This can be achieved in practice by queueing voice traffic in a separate traffic class. By inspecting the queue we can determine the number of distinct wireless stations to which queued packets are destined and this provides a direct measure of the number of active downlink calls.

[2]With $CW_{min} = 32$, the average station countdown time is $CW_{min}/2 = 16$ slots or $300\mu s$. Using this value the capacity of the wireless channel falls to 10 calls using our simple calculation. A measured capacity of 15 voice calls implies that the average countdown time is significantly less than $CW_{min}/2$.

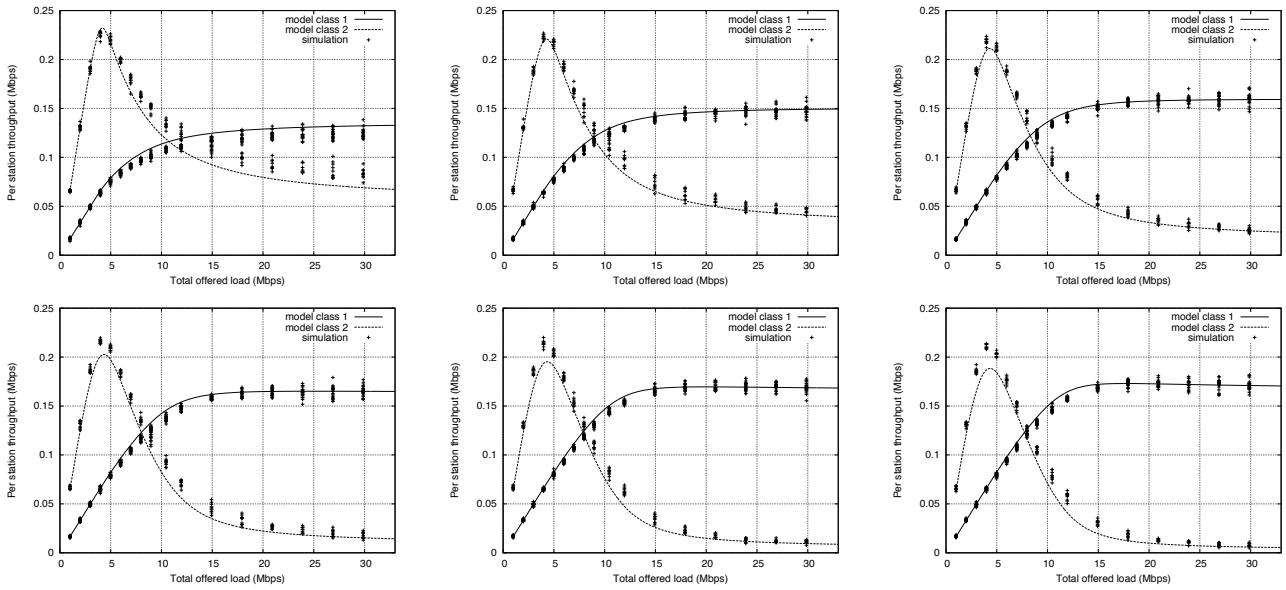


Fig. 4. Throughputs for two classes with varying load. The first class is prioritised by increasing the AIFS value for the second class. The total number of stations is 30. Each figure shows the throughput as the load in each class is kept in a fixed ratio of two and gradually increased.

	Duration(μs)
PLCP Header (1Mbs)	192
MAC Header	20.4
IP Header	14.5
Payload	58.2
DIFS	50
SIFS	10
ACK (1Mbs)	304
Total	649.1

TABLE I

OVERHEAD FOR A PACKET WITH A PAYLOAD OF 80 BYTES, BASIC RATE OF 1MBS AND DATA RATE OF 11MBS.

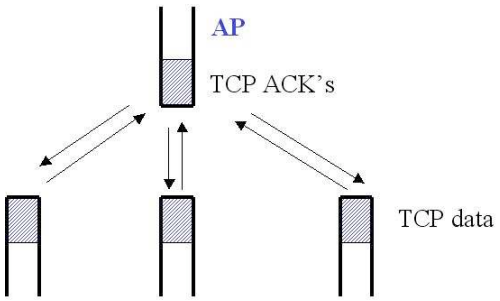


Fig. 7. TCP uploads. TCP data packets are queued at the wireless stations pending transmission to the access point. TCP ACK packets travel from a shared queue in the access point to the wireless stations.

improvement is limited.

V. DATA TRAFFIC ONLY

Figure 2 illustrates the behaviour of competing TCP upload flows over an 802.11b WLAN. Gross unfairness between the throughput achieved by competing flows is evident. Unfairness also exists between competing upload and download TCP flows. This is illustrated for example in Figure 3 where it can be seen that upload flows achieve nearly an order of magnitude greater throughput than competing download flows.

A. Unfairness between competing TCP upload flows

The source of the unfairness between competing TCP uploads is rooted in the interaction between the MAC layer contention mechanism (that enforces fair access to the wireless channel) and the TCP transport layer flow and congestion control mechanisms (that ensure reliable transfer and match source send rates to network capacity).

This is explained by statistical multiplexing, with active stations counting down simultaneously when the medium is idle.

At the transport layer, to achieve reliable data transfers TCP receivers return acknowledgement (ACK) packets to the data sender confirming safe arrival of data packets. During TCP uploads, the wireless stations queue data packets to be sent over the wireless channel to their destination and the returning TCP ACK packets are queued at the wireless access point (AP) to be sent back to the source station, see Figure 7. TCP's operation implicitly assumes that the forward (data) and reverse (ACK) paths between a source and destination have similar packet transmission rates. The basic 802.11 MAC layer, however, enforces station-level fair access to the wireless channel. That is, n stations competing for access to the wireless channel are each able to secure approximately a $1/n$ share of the total available transmission opportunities. Hence, if we have n wireless stations and one AP, each station (including the AP) is able to gain only a $1/(n+1)$ share of transmission opportunities. By allocating an equal share of packet transmissions to each wireless station, with TCP uploads the 802.11 MAC allows $n/(n+1)$ of transmissions to be TCP data packets yet only $1/(n+1)$ (the AP's share of medium access) to be TCP ACK packets. For larger numbers of stations, n, this MAC layer ac-

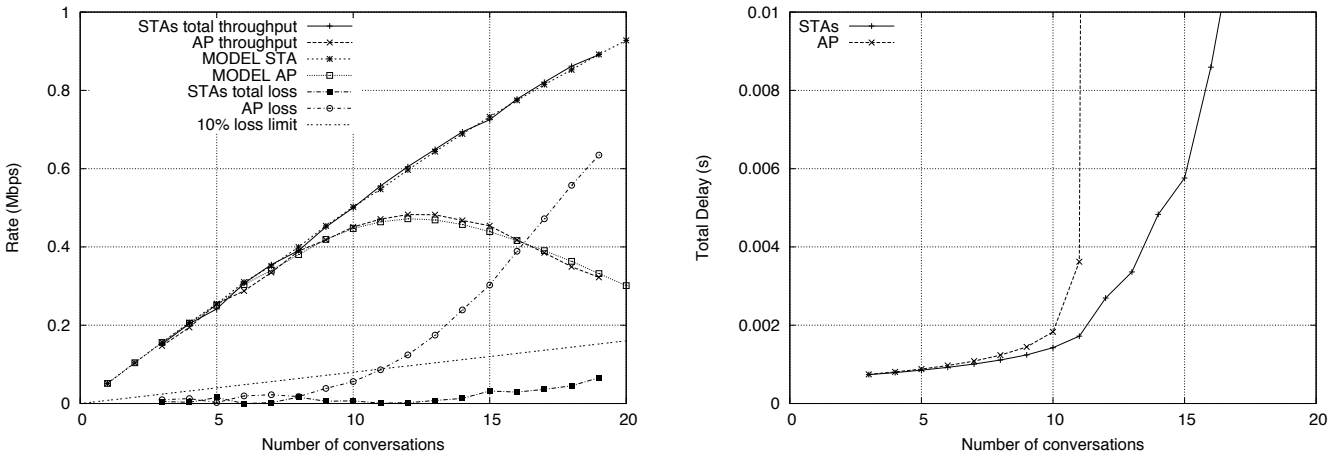


Fig. 5. Throughput and delay for competing voice calls in an infrastructure mode 802.11b WLAN.

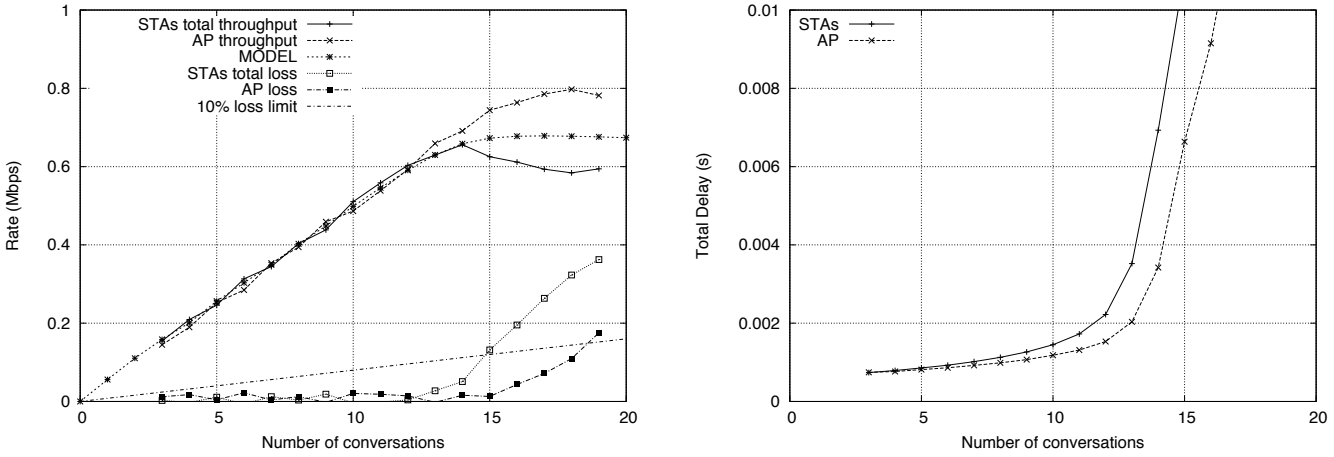


Fig. 6. Throughput and delay for competing voice with prioritisation of the AP using the 802.11e $TXOP$.

tion leads to substantial forward/reverse path asymmetry at the transport layer.

Asymmetry in the forward and reverse path packet transmission rate is a known source of poor TCP performance in wired networks, e.g. see [2]. If the reverse path ACK transmission rate is k times slower than the forward path data packet transmission rate, the reverse path is liable to become congested before the forward path causing TCP ACK packets to be dropped. On average, only one ACK will get through for every k data packets transmitted. This degrades performance in a number of ways. First, each ACK packet will on average acknowledge k data packets, thereby disrupting the ACK clocking within TCP and typically leading to increased burstiness in the rate at which the TCP sender transmits data packets. Second, infrequent ACKs can hamper congestion window growth at the TCP sender and hence interfere with the TCP congestion control algorithm that is seeking to match the TCP send rate to the available network capacity. Third, a pathological interaction with the TCP timeout mechanism is often created, which can be understood as follows: A TCP sender probes for extra bandwidth until a data packet is lost or a timeout occurs. A timeout is invoked at a TCP sender when no progress is detected in the arrival of data packets at the destination - this may be due to data packet loss (no data packets arrive at the destination), TCP ACK packet loss (safe receipt of data packets is not reported back to the sender), or both. TCP flows with only a small number of packets in flight (e.g. flows which have recently started or which are recovering from a timeout) are much more susceptible to timeouts than flows with large numbers of packets in flight since the loss of a small number of data or ACK packets is then sufficient to induce a timeout. Hence, on asymmetric paths where ACK losses are frequent a situation can easily occur where a newly started TCP flow loses the ACK packets associated with its first few data transmissions, inducing a timeout. The ACK packets associated with the data packets retransmitted following the timeout can also be lost, leading to further timeouts (with associated doubling of the retransmit timer) and so creating a persistent situation where the flow is completely starved for long periods; this is particularly prevalent in wireless networks, see for example Figure 2.

B. Unfairness between competing TCP upload and download flows

To understand this behaviour, consider the situation where we have only TCP downloads. Download data packets are transmitted by the AP and on receiving a data packet a wireless

station generates a TCP ACK (we ignore delayed acking for the moment to streamline the present discussion), see Figure 8. Importantly, wireless stations only generate TCP ACK packets on receipt of a TCP data packet and otherwise do not contend for medium access. Consequently, TCP downloads typically exhibit a quasi-polled behaviour. Namely, the AP transmits a data packet to a wireless station which then responds with a TCP ACK while the other wireless stations remain silent. Hence, regardless of the number of TCP download flows, generally only *two* stations (the AP and the most recent destination wireless station) contend for medium access at any time. This behaviour has also been noted in [5].

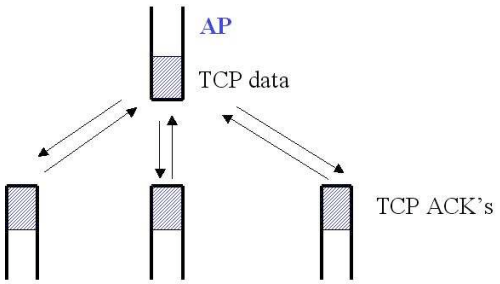


Fig. 8. TCP downloads. TCP data packets travel from access point to wireless stations. TCP ACK packets travel from stations to the access point.

Considering now a mix of competing upload and download TCP flows, suppose we have n_u upload flows and n_d download flows. Owing to their quasi-polling behaviour, we have that the download flows (regardless of the number n_d of download flows) gain transmission opportunities at the roughly same rate as a *single* TCP upload flow. That is, roughly $1/(n_u + 1)$ of the channel bandwidth is allocated to the download flows and $n_u/(n_u + 1)$ allocated to the uploads. As the number n_u of upload flows increases, gross unfairness between uploads and downloads can result.

C. Restoring fairness: TCP Uploads

Existing approaches to alleviating the gross unfairness between TCP flows competing over 802.11 WLANs work within the constraint of the current 802.11 MAC, resulting in complex adaptive schemes requiring online measurements and, perhaps, per packet processing. We instead consider how the additional flexibility present in the new 802.11e MAC might be employed to alleviate transport layer unfairness.

With regard to TCP uploads, unfairness arises from the asymmetry between the bandwidths of the forward and reverse paths. Symmetry can be restored by configuring the AP such that it effectively has unrestricted access to the wireless medium while the other stations divide the channel capacity not used by the AP fairly amongst themselves as per the standard 802.11 mechanism. Rather than allowing unrestricted access to all traffic sent by the AP, recall that in 802.11e the MAC parameter settings are made on a per class basis. Hence, we propose collecting TCP ACKs into a single class (i.e. queue them together in a separate queue at the AP) and confine prioritisation to this class.

The rationale for this approach to differentiating the AP makes use of the transport layer behaviour. Namely, allowing TCP ACKs unrestricted access to the wireless channel does not lead to the channel being flooded. Instead, it ensures that the volume of TCP ACKs is regulated by the transport layer rather than the MAC layer. In this way the volume of TCP ACKs will be matched to the volume of TCP data packets, thereby restoring forward/reverse path symmetry at the transport layer. When the wireless hop is the bottleneck, data packets will be queued at wireless stations for transmission and packet drops will occur there, while TCP ACKs will pass freely with minimal queuing i.e. the standard TCP semantics are recovered.

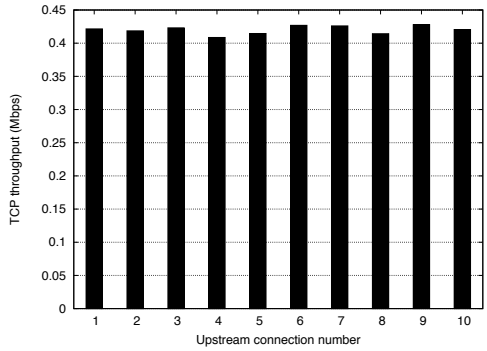


Fig. 9. Throughput of competing TCP uploads. (NS simulation, 802.11e WLAN, .11b PHY, 10 upload TCP flows, TCP ACKs prioritised at AP with $AIFS = 50\mu s$ and $CW_{min} = 2$, wireless stations $AIFS = 90\mu s$ and $CW_{min} = 32$.)

D. Restoring fairness: TCP Downloads

With regard to TCP downloads, recall that the primary source of unfairness arises from the quasi-polling behaviour of TCP downloads which means that if we have n_u uploads and n_d downloads then the download flows roughly win only a $1/(n_u+1)$ share of the available transmission opportunities. This suggests that to restore fairness we need to prioritise the download data packets at the AP so as to achieve an $n_d/(n_u + n_d)$ share.

While we might prioritise download data packets by using an appropriate value of CW_{min} at the AP for TCP data packets, the utility of CW_{min} is constrained by the availability of only a coarse granularity (CW_{min} can only be varied by powers of two in 802.11e). The $AIFS$ parameter might also be used, but seems better suited to strict prioritisation rather than proportional prioritisation. Instead, we propose that the $TXOP$ packet bursting mechanism in 802.11e provides a straightforward and fine grained mechanism for prioritising TCP download data packets. Since the download TCP data traffic gains a $1/(n_u+1)$ share of transmission opportunities, by transmitting n_d packets (one packet to each of the n_d download destination stations[3]) at each transmission opportunity it can be immediately seen that we restore the $n_d/(n_u + n_d)$ fair share to the TCP download traffic.

[3]Specifically, we queue TCP data packets in a separate traffic class at the AP. By inspecting this queue we can determine both the current number n_d of distinct destination stations. When the traffic class wins a transmission opportunity, we use a $TXOP$ value of n_d packets and transmit one packet to each of the destination stations. The effect is to dynamically track the number of active TCP download stations and always ensure the appropriate prioritisation of TCP download traffic

Comment: Packet Burst Size. With this $TXOP$ approach the AP transmits n_d packets in a single burst. For n_d large, this can result in the AP occupying the channel for a substantial consolidated period of time and this may, for example, negatively impact competing delay sensitive traffic. We can address this issue in a straightforward manner by using multiple smaller bursts instead of a single burst. When using smaller packet bursts, it is necessary to ensure a corresponding increase in the number of transmission opportunities won by the AP. This can be achieved by using a smaller value of CW_{min} for the TCP data packet traffic class at the AP. It is shown in [3] that competing traffic classes gain transmission opportunities approximately in inverse proportion to their values of CW_{min}. Let k denote the ratio of the wireless station TCP data class CW_{min} value to that of the AP TCP data class. Scaling k with the number of transmission opportunities required provides coarse (recall that in 802.11e k is constrained to be a power of two) prioritisation of download TCP flows. We then complement this with use of $TXOP$ for fine grained adjustment of the packet burst lengths, scaling $TXOP$ with $1/k$. Hence fine grained prioritisation can be achieved while avoiding unduly large packet bursts.

In addition to prioritisation of download data packets at the AP, in line with the discussion regarding TCP uploads it is also necessary to prioritise the TCP download ACKs using $AIFS$ to mitigate queueing and loss of TCP ACKs at the wireless stations. While in the case of TCP uploads the TCP ACKs are queued only at the AP and hence there is no contention (i.e no collisions) between the TCP ACKs of competing TCP flows in accessing the wireless channel, with TCP downloads the TCP ACK packets are queued at the wireless stations and thus can contend with each other. The 802.11 standard value of 32 for CW_{min} is therefore suggested for TCP download ACK traffic as providing a reasonable balance between number of collision and channel idle time.

Revisiting the example in Figure 3, the impact of the proposed prioritisation approach can be seen in Figure 11. Evidently, fairness is restored between the competing TCP flows. The 802.11e MAC parameter settings used in this example (with an 11Mbs PHY) for both TCP uploads and downloads are summarised in Table II.

Comment: CW_{min} Selection. We can verify that this choice of CW_{min} is sufficient, in combination with using an $AIFS$ value of zero, to prevent a backlog of TCP download ACKs building at the wireless stations. A sustained backlog will occur if, on average, the transmission rate of TCP download ACKs on the wireless channel is less than the transmission rate of TCP download data packets (neglecting delayed acking for simplicity). In this situation, the stations sending TCP download ACKs are in a so-called saturated condition where they always have a packet to send, and hence can be modelled using the approach in [3]. By starting with a large value of CW_{min} for the TCP ACK traffic (so that the TCP ACK's are backlogged) and reducing CW_{min} until the TCP data transmission rate just equals the TCP ACK transmission rate we can determine the stability boundary for TCP ACK queueing. The stability boundary determined in this way is shown in Figure 10, and provides an upper bound on the value of CW_{min} for TCP ACK traffic. It can be seen that a value of 32 lies within the stability region across the range of operating conditions of interest.

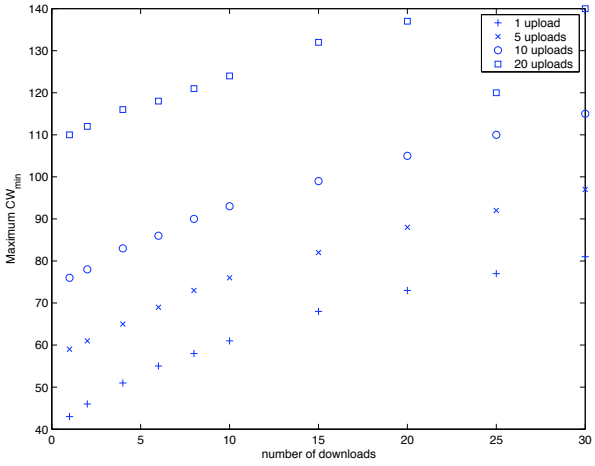


Fig. 10. Maximum CW_{min} vs number of upload and download stations, 802.11b PHY.

Comment: Bidirectional Flows. In the case of bidirectional TCP flows packets may contain both ACK information and carry a data payload. We suggest that such packets be assigned to the TCP data data packet class. Unlike ACK packets, loss of data packets results in backoff of the TCP congestion window and to maintain the correct congestion control semantics (and avoid packet reordering) all packets carrying a data payload should be queued together. With this approach, ACK information is piggy-backed on data packets subject to congestion control requirements. When data packet transmission is constrained by congestion, pure TCP ACK's are used.

Comment: Throughput Optimisation. The value for CW_{min} for TCP ACK traffic used here is not necessarily optimal with respect to throughput, but optimisation with traffic load is left for future consideration.

		$AIFS$ (slots)	CW_{min}	$TXOP$ (packets)
AP	TCP ACKS	0	1	1
	TCP data	4	32	n_d
wireless station	TCP ACKS	0	32	1
	TCP data	4	32	1

TABLE II
TCP UPLOAD/DOWNLOAD 802.11E MAC PARAMETERS

VI. VOICE AND DATA TRAFFIC

Previous sections have considered 802.11 networks carrying, respectively, only voice traffic and only data traffic. In this Sec-

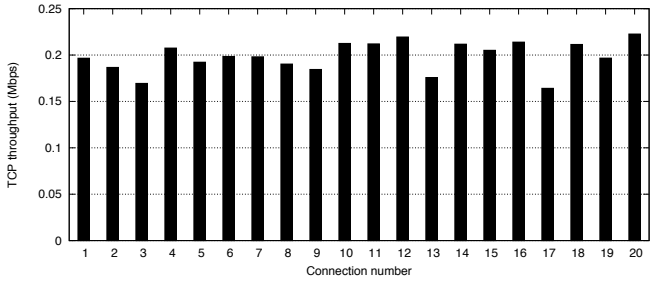


Fig. 11. Throughput of competing TCP uploads and downloads; first 10 flows are TCP uploads, remainder are TCP downloads. (NS simulation: single cell infrastructure mode 802.11e WLAN, parameters as in Table II, .11b PHY.)

tion we consider networks with a mixture of voice and data traffic. That the potential exists for negative interactions between voice and data traffic is readily verified. Indeed, we show that even a relatively small amount of data traffic is sufficient to disrupt voice calls, see Figure 1.

One fundamental difference between voice and data traffic is that TCP data sources may be greedy i.e. effectively always have a packet to send[4], while voice sources are not and have a relatively low (64Kbs or less) maximum transmission rate. This is a key observation as it means, subject to some policing, that voice traffic can safely be allowed strictly prioritised access to the wireless medium.

In this Section we consider how the mechanisms provided by the 802.11e MAC can be used to suitably prioritise voice traffic. Our requirements for a mixed voice/data network are

(i) **Voice QoS.** Voice traffic experiences similar throughput and delay as in a voice-only network (see Section IV).
(ii) **Data QoS.** Competing data flows are treated fairly.
(iii) **Efficiency.** Network capacity is efficiently used. That is, data traffic is able to utilise spare capacity left unused by competing voice traffic.

A. Prioritising voice over data

As the number of voice calls increases we require that data traffic makes space for the voice calls, and conversely that data traffic is free to grab spare capacity if the number of voice calls decreases. While this might be achieved via real-time measurements and adaptation of MAC parameters, the $AIFS$ mechanism provided by the 802.11e MAC seems potentially to provide this type of functionality in a more direct manner.

The 802.11 MAC employs a CSMA binary exponential backoff approach. Time is slotted and stations count down a random number of slots before transmitting. Countdown starts only after the wireless medium has been sensed silent for a period $AIFS$. Importantly, the countdown is paused when the medium is sensed busy, and resumes only after a period $AIFS$ of silence. In a network with a single traffic class where all flows have the same value of $AIFS$, the contention mechanism corresponds to the standard 802.11 approach. However, in a network with two traffic classes, each having different values of $AIFS$, the behaviour is different. Following a channel busy event, flows in the class with smaller $AIFS$ value (i.e. voice in our case) resume countdown more quickly than those with larger $AIFS$ value (i.e. data flows). When the channel is lightly loaded, so that channel busy events are relatively rare, the impact of this difference is small. However, at every channel busy event the voice flows gain an advantage over the data flows and thus as the channel becomes more heavily loaded this advantage quickly accumulates.

The $AIFS$ parameter therefore provides load sensitive prioritisation. When the channel is lightly loaded, voice and data traffic behave similarly. However, as the load increases voice traffic receives preferential treatment. Moreover, the advantage awarded to the voice traffic increases exponentially with the traffic load (and with the difference in $AIFS$ values) and so can be used to ensure effectively strict prioritisation of the voice traffic. We explore this behaviour in more detail below.

We consider an infrastructure mode WLAN with all voice and data traffic routed via the AP. Voice conversations are simulated as two-way on-off flows as discussed in Section IV. For simplicity, we assume one wireless station per voice call, and one wireless station per data flow. Data traffic is managed using the scheme discussed in Section V.

Figure 12 shows the MAC delay of a single voice call both as the number of competing data flows is increased and as the difference in $AIFS$ values of the voice and data flows is varied. When the $AIFS$ difference is zero, the delay of the voice call is similar to that of the data flows. As the $AIFS$ difference is increased, the delay experienced by the voice call decreases.

We can make the following observations. Firstly, notice that the benefit of increasing $AIFS$ is most pronounced when there is a large number of competing data flows, as expected from the foregoing discussion. Secondly, as $AIFS$ is increased beyond about 4 slots it can be seen that there is a rapidly diminishing return in terms of reduction in the delay. That is, the delay effectively becomes constant as the the $AIFS$ difference is increased and number of stations increases (the residual delay is associated with the countdown and CW_{min} value of the voice call - see below). This is a consequence of the exponential impact of $AIFS$, which means that we rapidly approach strict prioritisation of the voice call.

Another consequence of the exponential behaviour of $AIFS$ prioritisation is that a single, constant value of $AIFS$ can potentially be used across a wide range of operating conditions obviating the need for complex measurement-based adaptive strategies. Figure 13 shows how this is reflected in the throughput for a voice conversation prioritised with $AIFS = 4$ slots. It can be seen that the throughput of the voice stations remains practically unaffected by the increasing number of data flows.

The foregoing results are for a single voice call competing against varying numbers of data flows. Also marked on Figure 13 is the per station throughput as the number of data flows is held constant and the number of voice calls increased.

[4]We consider here TCP flows that are network rather than application constrained, i.e. that have a large quantity of data to transfer. Bursty TCP flows, such as web traffic, are considered later.

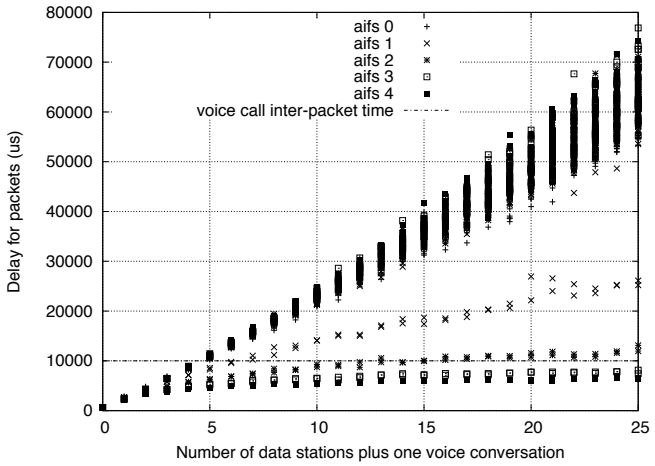


Fig. 12. MAC delay for stations where one voice call competes with a number of data stations. As the value of AIFS increases the voice call delay approaches an asymptotic value. The solid line marks the inter-packet interval for the voice calls - MAC delays above this are associated with an unstable queueing regime. (.11e MAC with CWmin = 31, AIFS voice = 0, AIFS data as shown, .11b PHY.)

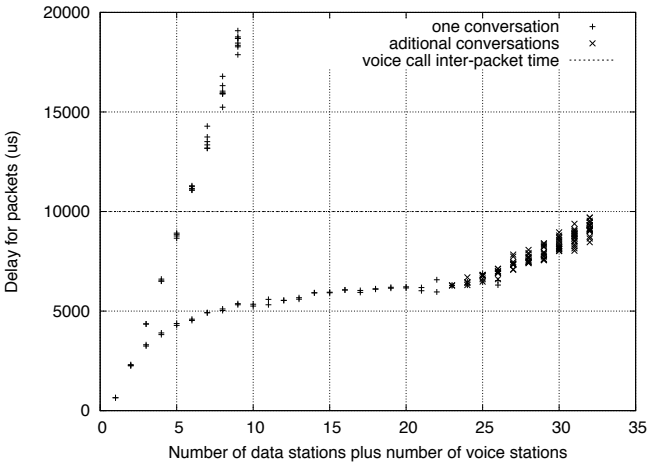


Fig. 14. For AIFS 4, the delay of one voice call competing with a number of data stations, and the delay for additional voice stations competing with 22 data stations. Note the linear increase in the delay as we increase the number of voice calls. The solid line marks the inter-packet interval for the voice calls. (.11e MAC with CWmin = 31, AIFS voice = 0, AIFS data = 4, .11b PHY.)

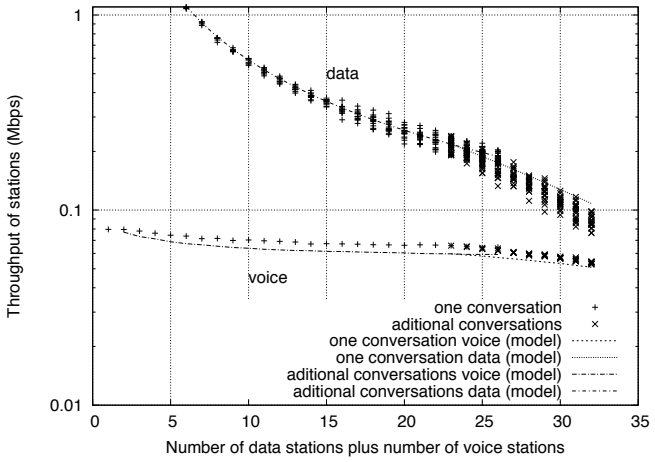


Fig. 13. For AIFS 4 we show the throughput of one voice call competing with a number of data stations, and the throughput of additional voice stations competing with 22 data stations. (.11e MAC with CWmin = 31, AIFS voice = 0, AIFS data = 4, .11b PHY.)

The voice calls are each able to achieve their demanded throughput while it can be seen that as the number of voice calls increases the data throughput falls, as expected. Figure 14 also shows the MAC delays. It can be seen that the delay for voice calls increases roughly linearly with the number of calls, reflecting the increased contention for channel access between high-priority traffic.

Comment: While the foregoing voice/data results are for large data transfers, we have obtained similar results with web traffic.

B. Reducing MAC delay

As noted previously, the limiting voice call MAC delay with $AIFS$ prioritisation is associated with the random backoff countdown and thus the CW_{min} value of the voice calls. In the foregoing plots the voice calls use the standard CW_{min} value of 32.

The value of CW_{min} directly affects the throughput and delay of a traffic class. When CW_{min} is reduced, the probability of packet collisions increases. When CW_{min} is increased, the channel idle time increases as stations spend more time counting down between transmissions. These factors, and the optimal value of CW_{min}, are load dependent and the standard value of $CW_{min} = 32$ reflects a reasonable trade-off between these factors over a range of channel loads. We know, however, that with an 11Mbs PHY then even in a network with purely voice traffic only around 15 voice calls can be supported. This raises the question of whether we can use this information to fine tune the CW_{min} value for voice traffic to reduce MAC delay.

Figure 15 and Figure 16 show the voice call delay and throughput as CW_{min} is varied. It can be seen that reducing the voice call CW_{min} to 8 reduces the MAC delay by almost a factor of three. Looking at Figure 16 we see that the changes in CW_{min} have little impact on the achieved throughput of the voice calls, although decreasing CW_{min} decreases the data traffic throughput.

It can be seen from Figure 15 that with a CW_{min} of 8 we can support 15 voice conversations before the MAC delay exceeds the voice packet inter-arrival time (at which point the queueing delay grows). This is the same capacity as observed in Section IV in the context of voice-only networks. Hence, no loss in voice capacity is incurred in the mixed voice/data case.

C. Overall scheme

In addition to prioritisation of voice traffic over data, a second requirement in mixed voice/data environments is to ensure appropriate quality of service for the data flows. That is, we require reasonably fair sharing of the available wireless capacity between competing data flows. The unfairness between TCP uploads and downloads in 802.11 networks has already been noted and solutions discussed. While the solutions in Section V

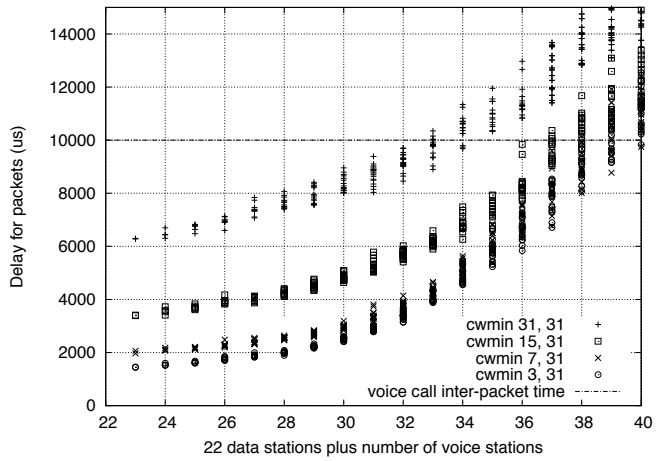


Fig. 15. For AIFS 4, show the delay for N voice stations competing with 22 data stations for different combinations of CW_{min}. Note that adjusting the CW_{min} of the higher (voice) class can reduce the waiting time. (.11e MAC with CWmin as shown, AIFS voice = 0, AIFS data = 4, .11b PHY.)

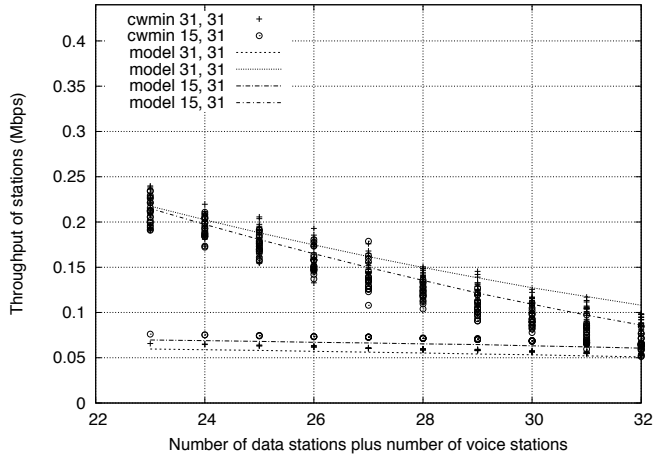


Fig. 16. For AIFS 4, the throughput for N voice stations competing with 22 data stations for different combinations of CW_{min}. (.11e MAC with CWmin as shown, AIFS voice = 0, AIFS data = 4, .11b PHY.)

were derived in the context of data-only networks, they can be readily extended to mixed voice/data networks.

Specifically, we assign voice traffic, TCP data packets and TCP ACK packets to separate traffic classes at the AP and at the wireless stations. The proposed 802.11e MAC parameter settings for these traffic classes are given in Table III. The settings for the TCP data and TCP ACK traffic are identical to those studied previously in Section V. In line with the previous discussion, we prioritise voice packets with an $AIFS$ advantage of 4 slots over TCP data packets and use a CW_{min} value of 8 to reduce MAC delay. Note that our analysis indicates that a constant $AIFS$ prioritisation of 4 slots is effective across a wide range of network conditions and further adaptation of $AIFS$ is not necessary.

With this approach, voice packets and TCP ACK's are prioritised in a similar manner. Our aim in prioritising TCP ACK's is, however, quite different from our aim in prioritising the voice packets. In the case of TCP ACK's we are seeking to avoid damaging interactions between the action of the transport layer congestion control and MAC layer contention mechanism. The volume of TCP ACK's is then regulated by the transport layer to be proportional to the volume of TCP data packets. Since the TCP data packets are lower priority than the voice packets, *both* TCP data packets and TCP ACK packets are throttled as the level of voice traffic increases and strict prioritisation of the voice traffic is thereby maintained.

We note that the simulation results previously presented in Sections VI-A and VI-B are with data traffic configured in this manner and confirm the effectiveness of the approach for prioritising voice traffic. The fairness achieved between data flows is illustrated in Figure 17. Voice call throughput remains constant, irrespective of the presence of TCP stations. The mean delay experienced by voice call packets increases, but remains below the inter-packet arrival time of 10ms.

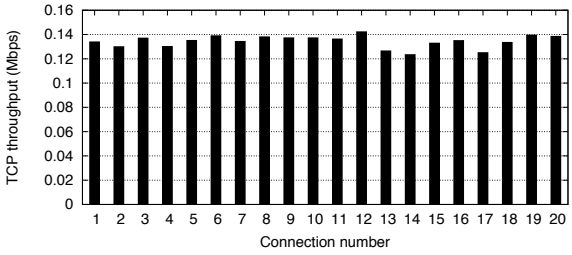


Fig. 17. Throughput of 10 competing TCP uploads, 10 downloads and 5 voice conversations. In histogram, first 10 points correspond to the TCP uploads, the remainder are TCP downloads. (NS simulation, 802.11e WLAN, parameters as in Table III, .11b PHY).

Comment: CW_{min} **Selection.** CW_{min} influences the tradeoff between channel idleness and packet collisions and thus the optimal value is load dependent. Similarly to the current 802.11 approach, here we use fixed values of CW_{min} that yield good (sub-optimal) performance performance across a range of traffic conditions - consideration of on-line adaptation of CW_{min} is left as future work. The TCP data packets use the standard 802.11 CW_{min} value of 32 slots. We have found that this provides close to optimal throughput (within 15 % for an 802.11b PHY) across a wide range of traffic conditions. TCP ACKs also use a CW_{min} value of 32 slots at the wireless stations. The choice of this value is discussed in detail in Section V. Voice traffic uses a CW_{min} value of 8. This lower CW_{min} value reflects our knowledge that the maximum number of voice calls that can be supported with an 11Mbs PHY is only around 15, which consequently limits channel contention.

Comment: Efficiency When we increase the value of $AIFS$ for the data traffic we can expect a loss in data throughput owing to the increase in idle time. Our analysis indicates a reduction of about 3% in data throughput for $AIFS = 4$ with 1500 byte packets and an .11b PHY.

		$AIFS$ (slots)	CW_{min} (slots)	$TXOP$ (packets)
AP	TCP ACKS	0	1	1
	Voice	0	8	n_{d_v}
	TCP data	4	32	$n_{d_{tcp}}$
wireless station	TCP ACKS	0	32	1
	Voice	0	8	1
	TCP data	4	32	1

TABLE III

VOICE AND DATA 802.11E MAC PARAMETERS; n_{d_v} DENOTES THE NUMBER OF ACTIVE VOICE DOWNLINK CALLS, $n_{d_{tcp}}$ DENOTES THE NUMBER OF ACTIVE TCP DOWNLINK FLOWS.

D. Scope of our results

Our principal aim in this paper has been to develop a soundly-based strategy for selecting 802.11e MAC parameters in order to meet the QoS requirements of voice and data traffic. We do not claim that this is the only approach that might be taken to meeting voice and data QoS requirements, but we do argue that the proposed approach has the merit of being very straightforward and of imposing only a small computational burden (e.g. we use simple fixed parameter settings and complex measurement-based adaptation is avoided). While the approach taken is quite general, we have focussed on using an 11Mbs PHY to demonstrate results. We have also confined attention to G.711 voice traffic both in order to streamline the development and because this codec places the greatest demands upon the wireless channel. More efficient codecs can be expected to increase voice capacity, but are not expected to qualitatively change the results. Admission control of voice calls is necessary but is already the subject of an extensive literature and is therefore not considered in detail here.

VII. CONCLUSIONS

In this paper our objective has been to develop a soundly-based strategy for selecting 802.11e MAC parameters in networks carrying mixed voice and data traffic. Specific contributions include the following

- We demonstrate that unfairness exists in the 802.11 DCF when we have a mix of greedy and non-greedy flows, e.g. voice plus data.
- A new analytic model of 802.11e networks is developed that is capable of capturing the behaviour of voice and data traffic and the impact of the 802.11e prioritisation mechanisms on network behaviour.
- An 802.11e prioritisation strategy for voice-only networks that avoids throttling of flows at the AP.
- An 802.11e prioritisation strategy for data-only networks that restores fairness between competing upload and download flows and mixtures of both.
- An 802.11e prioritisation strategy for networks carrying a mix of voice traffic, data uploads and data downloads. This strategy ensures that (i) voice traffic experiences similar throughput and delay as in a voice-only network (ii) competing data flows are treated fairly and (iii) network capacity is efficiently used (data traffic is able to utilise spare capacity left unused by competing voice traffic).

The proposed 802.11e prioritisation approaches have the merit of being very straightforward and of imposing only a small computational burden (e.g. we use simple fixed parameter settings and complex measurement-based adaptation is avoided). The approaches are compatible with the WME subset of 802.11e that is supported by currently available hardware.

VIII. ACKNOWLEDGEMENTS

This work was supported by Science Foundation Ireland grant 03/IN3/I396.

REFERENCES

[1] F. Anjum, M. Elaoud, D. Famolari, A. Ghosh, R. Vaidyanathan, A. Dutta, P. Agrawal, T. Kodama, and Y Y. Katsube. Voice performance in WLAN networks — an experimental study. In *IEEE GLOBECOM*, volume 6, pages 3504–3508, 2003.

[2] H. Balakrishnan and V. Padmanabhan. How network asymmetry affects TCP. *IEEE Communications Magazine*, pages 60–67, 2001.

[3] Roberto Battiti and Bo Li. Supporting service differentiation with enhancements of the IEEE 802.11 MAC protocol: models and analysis. Technical Report DIT-03-024, University of Trento, May 2003.

[4] G. Bianchi. Performance analysis of IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, March 2000.

[5] R. Bruno, M. Conti, and E. Gregori. Throughput analysis of TCP clients in wi-fi hot spot networks. In *Proceedings of WONS, Trento, Italy*, 2004.

[6] D. Chen, S. Garg, M. Kappes, and K. Trivedi. Supporting voip traffic in ieee 802.11 wlan with enhanced medium access control (mac) for quality of service. In *Avaya Labs Technical Report*, number ALR-2002-025, 2002.

[7] M. Coupechoux, V. Kumar, and L. Brignol. Voice over ieee 802.11b capacity. In *16th ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Networks*, 2004.

[8] A. Detti, E. Graziosi, V. Minichiello, S. Salsano, and V. Sangregorio. TCP fairness issues in IEEE 802.11 based access networks. *submitted paper*, 2005.

[9] D.P. Hole and F.A. Tobagi. Capacity of an IEEE 802.11b wireless LAN supporting VoIP. In *International Conference on Communications*, 2004.

[10] David Malone, Ken Duffy, and Douglas J. Leith. Modelling the 802.11 distributed coordination function with heterogenous finite load. In *Proceedings of RAWNET 2005, Trento, Italy*, 2005.

[11] A.P. Markopoulou, F.A. Tobagi, and M.J. Karam. Assessing the quality of voice communications over internet backbones. *IEEE Transactions on Networking*, 11(5):747–760, October 2003.

[12] S. Pilosof, R. Ramjee, Y. Shavitt, and P. Sinha. Understanding TCP fairness over wireless LAN. In *Proceedings of INFOCOM, San Francisco, USA*, 2003.

[13] Jeffrey W. Robinson and Tejinder S. Randhawa. Saturation throughput analysis of IEEE 802.11e enhanced distributed coordination function. *IEEE Journal on selected areas in communications*, 22(5):917–928, June 2004.

[14] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma. Performance of reliable transport protocol over ieee 802.11 wireless lan: Analysis and enhancement. In *Proceedings of INFOCOM, New York, USA*, 2002.

[15] J. Yu, S. Choi, and J. Lee. Enhancement of VoIP over IEEE 802.11 WLAN via dual queue strategy. In *International Conference on Communications*, 2004.

APPENDIX

Our mean-field Markov model is an advancement based on combining the non-saturated 802.11b model of Malone, Duffy and Leith [10] and the saturated 802.11e model of Battiti and Li [3], which are themselves both developments of the saturated 802.11b model of Bianchi [4]. Due to space constraints, we do not describe fully how to solve the model. Instead we define the model completely and present equations that result from its

solution. We assume there are two classes of stations, labeled 1 and 2. Those in the class 1 are assumed to have a smaller or equal AIFS value to those in class 2, and are therefore of higher priority.

Stations in each class are modelled by distinct Markov chains whose transition probabilities are functions of their system parameters. The stationary distributions of these Markov chains are then coupled by the operation of the network. States in the Markov chain model for class 1 stations are labeled by a pair of integers (i,k) or $(0,k)_e$. The variable i represents the back-off stage, which is incremented to a maximum m when attempted transmission results in collision and set to 0 when transmission is successful. After attempted transmission the variable k is chosen randomly with a uniform distribution on the integers in the range $[0, W_i - 1]$, where $W_i = 2^i W$ and W is the minumum contention window. While the medium is idle, k is decremented. If a packet is present, transmission is attempted when $k = 0$. The empty states $(0,k)_e$ represent post-backoff. After successful transmission if a higher layer does not provide a packet, the MAC layer continues to decrement k to 0. When a higher layer provides a packet, if the medium is sensed idle, transmission is attempted immediately. If the medium is busy, a stage 0 back-off is initiated, now with a packet.

The chain for class 2 stations has to be augmented because their larger AIFS value results in class 1 stations counting down before class 2 stations treat the medium as idle. Let D be the difference in AIFS between class 2 and class 1. We model the behavior of a class 2 stations with a three dimensional Markov chain indexed (i,k,d) and $(0,k,d)_e$ if the MAC layer is empty, i.e. there is no packet in the MAC. The variable $d \in \{0,\ldots,D\}$ represents hold states for class 2. That is, $d > 0$ represents states in which the class 2 stations cannot decrement k while class 1 flows do, as they are not treating the medium as idle. When in a hold state class 2 stations must count up to D before returning to a non-hold state with $d = 0$.

Our main assumptions are: no errors are experienced on the channel other than those caused by collisions; conditioned on attempted transmission, stations in each class have a fixed probability of collision, p_i, $i = 1, 2$, irrespective of the network's history; for stations in each class, there is a fixed probability, q_i, of a packet arriving to the MAC during transitions in the Markov chains. In the following two subsections we define the transition probabilities for the chains describing stations in each class. The chains' stationary distributions then lead to the equations in Section C.

A. Class 1 stations' Markov chain

For notational convenience we drop class indicating subscripts; p is the probability of collision given attempted transmission, τ is the probability of transmission, and q is the probability a higher layer presents a packet to the MAC. The transition probabilities of a class 1 station's Markov chain are listed in full below. They are determined by straight-forward logic. For $0 < k < W_i$ we have

$$\begin{array}{rrcl} 0 < i \le m, & P((i,k-1)|(i,k)) & = & 1, \\ & P((0,k-1)_e|(0,k)_e) & = & 1-q, \\ & P((0,k-1)|(0,k)_e) & = & q. \end{array}$$

For $0 \le i \le m$ and $k \ge 0$ we have

$$\begin{array}{rcl} P((0,k)_e|(i,0)) & = & \frac{(1-p)(1-q)}{W_0}, \\ P((0,k)|(i,0)) & = & \frac{(1-p)q}{W_0}, \\ P((\min(i+1,m),k)|(i,0)) & = & \frac{p}{W_{\min(i+1,m)}}. \end{array}$$

The most complex transitions occur from the $(0,0)_e$ state

$$\begin{array}{rrcl} & P((0,0)_e|(0,0)_e) & = & 1-q+\frac{q(1-p)(1-p)}{W_0}, \\ k>0, & P((0,k)_e|(0,0)_e) & = & \frac{q(1-p)(1-p)}{W_0}, \\ k\ge 0, & P((1,k)|(0,0)_e) & = & \frac{q(1-p)p}{W_1}, \\ k\ge 0, & P((0,k)|(0,0)_e) & = & \frac{qp}{W_0}. \end{array}$$

B. Class 2 stations' Markov chain

For notational convenience class based subscripts are supressesd for the class 2 probabilities q and p. There are n_i stations in class i and the probability of transmission by a class i station is τ_i. Define P_{S_1} to be the probability that all class 1 stations are silent

$$P_{S_1} = (1-\tau_1)^{n_1}.$$

For $0 < k < W_i$ and $i > 0$ we have

$$\begin{array}{rcl} P((i,k-1,0) \mid (i,k,0)) & = & 1-p, \\ P((i,k,1) \mid (i,k,0)) & = & p, \\ P((0,k-1,0)_e \mid (0,k,0)_e) & = & (1-p)(1-q), \\ P((0,k-1,0) \mid (0,k,0)_e) & = & (1-p)q, \\ P((0,k,1)_e \mid (0,k,0)) & = & p(1-q), \\ P((0,k,1) \mid (0,k,0)) & = & pq. \end{array}$$

For $k \ge 0$ and $i \ge 0$,

$$\begin{array}{rcl} P((0,k,1)_e \mid (i,0,0)) & = & \frac{(1-p)(1-q)}{W_0}, \\ P((0,k,1) \mid (i,0,0)) & = & \frac{(1-p)q}{W_0}, \\ P((i+1,k,1) \mid (i,0,0)) & = & \frac{p}{W_i}. \end{array}$$

The final set of non-hold states we need to consider are if the window counter reaches 0 and there is still no packet to send. We deal with them in a way that enables us to give the explicit expression in equation (2). We refine $(0,0,0)_e$ further into the two states $(0,0,0)_{e,sense}$ and $(0,0,0)_{e,trans}$. In $(0,0,0)_{e,sense}$ the source is sensing if the medium is busy. If it is busy it goes to a hold state. If it is idle and no packet arrives it remains in the original state. If a packet arrives it goes to the second new state $(0,0,0)_{e,trans}$. In $(0,0,0)_{e,trans}$ the source transmits and may be successful and may then receive a new packet but regardless goes to some hold state. Thus

$$\begin{array}{rcl} P((0,0,1)_{e,sense} \mid (0,0,0)_{e,sense}) & = & p, \\ P((0,0,1)_{e,trans} \mid (0,0,0)_{e,trans}) & = & 1. \end{array}$$

Then with P_{ph} denoting the probability that a packet arrived while the chain is in a hold state, for $k \geq 0$ and $1 \leq j \leq D$,

$$\begin{aligned}
P((0,k,0) \mid (0,0,D)_{e,sense}) &= \frac{1}{W_0} P_{ph}, \\
P((0,0,0)_{e,sense} \mid (0,0,D)_{e,sense}) &= 1 - P_{ph}, \\
P((1,k,0) \mid (0,0,D)_{e,trans}) &= \frac{p}{W_1}, \\
P((0,k,0) \mid (0,0,D)_{e,trans}) &= \frac{1-p}{W_0} q, \\
P((0,k,0)_{e,sense} \mid (0,0,D)_{e,trans}) &= \frac{1-p}{W_0}(1-q), \\
P((0,0,0)_{e,trans} \mid (0,0,0)_{e,sense}) &= (1-p)(1-q), \\
P((0,0,j)_{e,sense} \mid (0,0,j-1)_{e,sense}) &= P_{S_1}, \\
P((0,0,j)_{e,trans} \mid (0,0,j-1)_{e,trans}) &= P_{S_1}, \\
P((0,0,0)_{e,sense} \mid (0,0,0)_{e,sense}) & \\
= (1-p)(1-q) + p(1-P_{ph}). &
\end{aligned}$$

For $j < D$,

$$\begin{aligned}
P((0,0,1)_{e,sense} \mid (0,0,j)_{e,sense}) &= (1 - P_{S_1}), \\
P((0,0,1)_{e,trans} \mid (0,0,j)_{e,trans}) &= (1 - P_{S_1}).
\end{aligned}$$

For $1 \leq j+1 \leq D$,

$$\begin{aligned}
P((0,k,j+1)_e \mid (0,k,j)_e) &= P_{S_1}(1-q), \\
P((0,k,j+1) \mid (0,k,j)_e)) &= P_{S_1} q.
\end{aligned}$$

For $k > 0$,

$$\begin{aligned}
P((0,k-1,0)_e \mid (0,k,D-1)_e) &= P_{S_1}(1-q), \\
P((0,k-1,0) \mid (0,k,D-1)_e) &= P_{S_1} q.
\end{aligned}$$

For $1 \leq j+1 \leq D$,

$$\begin{aligned}
P((0,k,1)_e \mid (0,k,j)_e) &= (1 - P_{S_1})(1-q), \\
P((0,k,1) \mid (0,k,j)_e) &= (1 - P_{S_1}) q
\end{aligned}$$

and

$$\begin{aligned}
P((0,0,0)_e \mid (0,0,D-1)_e) &= P_{S_1}(1-q), \\
P((0,0,0) \mid (0,0,D-1)_e) &= P_{S_1} q.
\end{aligned}$$

For $1 \leq j+1 < D$,

$$\begin{aligned}
P((i,k,j+1) \mid (i,k,j)) &= P_{S_1}, \\
P((i,k-1,0) \mid (i,k,D-1)) &= P_{S_1}, \\
P((i+1,k,0) \mid (1,0,D-1)) &= \frac{P_{S_1} p}{W_{i+1}}, \\
P((0,k,0) \mid (i,0,D-1)) &= \frac{P_{S_1}(1-p)q}{W_0}, \\
P((0,k,0)_e \mid (i,0,D-1)) &= \frac{P_{S_1}(1-p)(1-q)}{W_0}, \\
P((i,k,1) \mid (i,k,j)) &= (1 - P_{S_1}).
\end{aligned}$$

C. Model equations

We label variables related to the higher priority class with a subscript 1 and the lower priority class with subscript 2. There are n_i stations in class i and their minimum contention window, W_0, is W_i. The probability of a packet arrival to a station's MAC during a typical model state transition is q_i. Equation (5) relates q_i to offered load.

Solving for the stationary distribution of the Markov chains describing class 1 and 2 stations MAC behavior relates p_i and τ_i as follows

$$\tau_i = \frac{1}{\mathrm{norm}} \left(\frac{q_i^2 \mathrm{W}_i}{(1-q_i)(1-p_i)(1-(1-q_i)^{\mathrm{W}_i})} - \frac{q_i^2(1-p_i)}{1-q_i} \right), \tag{1}$$

where the normalisation constant, norm, is

$$\begin{aligned}
\mathrm{norm} &= \frac{q_i \mathrm{W}_i}{1-(1-q_i)^{\mathrm{W}_i}} + \frac{q_i \mathrm{W}_i (q_i \mathrm{W}_i + 3q_i - 2)}{2(1-q)(1-(1-q)^{\mathrm{W}_i})} + (1-q) \\
&+ \frac{q(\mathrm{W}_i+1)(p_i(1-q_i) - q_i(1-p_i)^2)}{2(1-q_i)} \\
&+ \frac{p_i q_i^2}{(1-q_i)2(1-p_i)} \left(\frac{\mathrm{W}_i}{1-(1-q_i)^{\mathrm{W}_i}} - (1-p_i)^2 \right) \\
&\left(\frac{2\mathrm{W}_i(1-p_i-p_i(2p_i)^{M-1})}{(1-2p_i)} + 1 \right).
\end{aligned}$$

From the network model we determine the stationary hold probability that in a typical slot the difference in $AIFS$, D, between the two classes is such that the higher class is counting down while the lower class is yet to consider the medium as idle. If D is zero, then the hold probability P_{hold} is zero. Otherwise it is

$$P_{\mathrm{hold}} = \frac{(1-(1-\tau_1)^{n_1}(1-\tau_2)^{n_2}) \sum_{i=1}^{D} (1-\tau_1)^{-in_1}}{1 + (1-(1-\tau_1)^{n_1}(1-\tau_2)^{n_2}) \sum_{i=1}^{D} (1-\tau_1)^{-in_1}}. \tag{2}$$

From the network model it is possible to deduce the following two non-linear equations, (3) and (4), that couple all stations in the network. Their solution completely determines p_i and τ_i, from which throughputs and other performance metrics can be determined:

$$p_1 = 1 - (1-\tau_1)^{n_1-1}(P_{\mathrm{hold}} + (1-P_{\mathrm{hold}})(1-\tau_2)^{n_2}) \tag{3}$$

$$p_2 = 1 - (1-\tau_1)^{n_1}(1-\tau_2)^{n_2-1}. \tag{4}$$

Having solved for p_1, p_2, τ_1, τ_2 and P_{hold} we can determine station throughputs, for example. We first determine the following probabilities, where $Q(n,m)$ is the probability that n from class 1 and m from class 2 attempt transmission in a typical slot and $Q(n+,m)$ is the probability that n or more of the higher class attempt transmission,

$$\begin{aligned}
Q(0,0) &= (1-\tau_1)^{n_1}(P_{\mathrm{hold}} + (1-P_{\mathrm{hold}})(1-\tau_2)^{n_2}), \\
Q(1,0) &= n_1\tau_1(1-\tau_1)^{n_1-1}(P_{\mathrm{hold}} + (1-P_{\mathrm{hold}})(1-\tau_2)^{n_2}), \\
Q(0,1) &= (1-\tau_1)^{n_1}(1-P_{\mathrm{hold}}) n_2 \tau_2 (1-\tau_2)^{n_2-1}, \\
Q(2+,0) &= (1-(1-\tau_1)^{n_1} - n_1\tau_1(1-\tau_1)^{n_1-1}) \\
&\quad (P_{\mathrm{hold}} + (1-P_{\mathrm{hold}}(1-\tau_2)^{n_2})).
\end{aligned}$$

The Markov chains do not run in real time. The time during state-transitions can be occupied by transmission, collision or an idle period. For ease of exposition, here we shall assume that class 1 packets are smaller than or equal to those of class 2.

To convert model quantities to real-world quantities, we rescale by the expected real time that lapses between chain transitions

$$E_s = Q(0,0)\sigma + Q(1,0)T_{S1} + Q(0,1)T_{S2} + Q(2+,0)T_{C1} + (1 - Q(0,0) - Q(1,0) - Q(0,1) - Q(2+,0))T_{C2},$$

where σ is the slot-length, T_{Si} is the time for successful transmission for a packet in class i and T_{Ci} is the collision time. These values are easily calculated from the payload size, physical data rate and MAC parameters. For example, see I.

With packet-lengths E_i in each class, assuming exponential inter-arrival times for packets to the MAC, the total offered loads to each class are

$$\frac{-\log(1-q_1)n_1E_i}{E_s} \quad \text{and} \quad \frac{-\log(1-q_2)n_2E_i}{E_s}. \quad (5)$$

Finally, the overall normalized throughputs in each class S_1 and S_2 are

$$S_1 = \frac{Q(1,0)E_i}{E_s} \quad \text{and} \quad S_2 = \frac{Q(0,1)E_i}{E_s}.$$