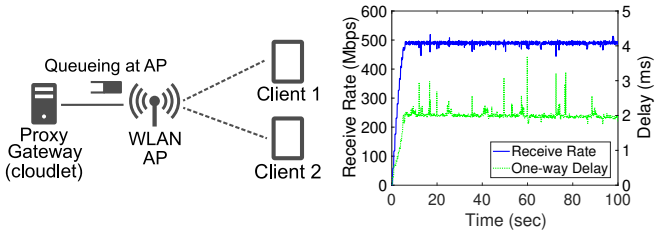
Regulating Queueing Delay in 802.11ac WLANs: Nonlinear Controller Analysis & Design

Francesco Gringoli, Douglas J. Leith

***Abstract*—In this paper we consider the analysis and design of a feedback controller to regulate queueing delay in a next generation edge transport architecture for 802.11ac WLANs. We begin by developing a simplified system model suited to control analysis and design, validated against both simulation and experimental measurements. Note that modelling of aggregation behaviour in WLANs is of course challenging in its own right. Using this model we develop a novel nonlinear control design inspired by the solution to an associated proportional fair optimisation problem. The controller compensates for plant nonlinearities and so can be used for the full envelope of operation. The robust stability of the closed-loop system is analysed and the selection of control design parameters discussed. We develop an implementation of the nonlinear control design and use this to present a performance evaluation using both simulations and experimental measurements.**



(a) (b)

Fig. 1. (a) Cloudlet-based edge transport architecture with bottleneck in the WLAN hop (therefore queueing of downlink packets occurs at the AP as indicated on schematic) and (b) Illustrating low-latency high-rate operation in an 802.11ac WLAN (measurements are from a hardware testbed located in an office environment).

I. Introduction

In this paper we consider the analysis and design of a transport layer feedback controller to regulate queueing delay in 802.11ac Wireless LANs (WLANs). The TCP transport layer in networks has two main functions, namely (i) congestion control (i.e. regulating the send rate so as to mitigate congestion) and (ii) reliable in-order delivery (i.e. retransmitting or otherwise repairing lost packets as needed). Recently, [1] introduced the idea of using measurements of the aggregation level in wireless transmissions to regulate the transport layer send rate so as to achieve high rate while avoiding queue build-up and so maintaining low delay. The setup considered is a next generation edge transport architecture of the type illustrated in Figure 1(a). Traffic to and from client stations is routed via a proxy located close to the network edge (e.g. within a cloudlet), which creates the freedom to implement new transport layer behaviour over the path between proxy and clients, which in particular includes the last wireless hop. While the standard TCP transport congestion control algorithm needs to be able to operate robustly across a wide range of network configurations, from dial-up connections to multi-gigabit datacentre backbones, a more specialised algorithm can be used in this edge architecture since the range of network configurations is much more limited.

In [1] this scope for specialisation is exploited with the aim of developing a new congestion control algorithm that simultaneously achieves high rate and low delay across the path from proxy to client. This is motivated by one of the most challenging requirements in next generation networks is the provision of connections with low end-to-end latency. In most use cases the target is for $<$100ms latency, while for some applications it is $<$10ms [2, Table 1]. In part, this reflects the fact that low latency is already coming to the fore in network services, but the requirement for low latency also reflects the needs of next generation applications such as augmented reality and the tactile internet. Note that modifications at the transport layer, rather than at the MAC layer, are appealing for several reasons but primarily because of the relative ease of rollout. Rollout of changes in the edge architecture of Figure 1(a) requires (i) implementation of the proxy, which can be readily carried out by spinning up a VM in the cloud, and (ii) installation of an app in the client stations, which also can be readily carried out via standard app store mechanisms. In contrast, MAC layer changes require firmware (and perhaps hardware) modifications to both the access point (AP) and the clients (i.e.user phones, laptops etc). Such changes are much more intrusive and experience indicates that it can take many years to achieve their wide rollout.

In [1] a simple linear PI controller is used to regulate the send rate based on the measured aggregation level of wireless transmissions. No stability analysis is provided and consideration is confined to a small operational envelope where the system exhibits linear behaviour. In the present paper we extend this line of work in a number of directions.

Firstly, we develop a simplified system model suited to control analysis and design. This model is validated against both simulation and experimental measurements. Note that modelling of aggregation behaviour in WLANs involves analysis of queueing at the AP and so of the complex interaction between the packet arrival rate at the AP and the wireless transmission rate. We sidestep these technical difficulties in two ways. Firstly, since we control the sender we can make use of packet pacing i.e. transmission of packets at fixed

F. Gringoli (francesco.gringoli@unibs.it) is with University of Brescia, Italy.
D. J. Leith (doug.leith@tcd.ie) is with Trinity College Dublin, Ireland.

intervals, thereby simplifying the packet arrival process at the AP. Secondly, since our focus is very much on low delay operation we can confine consideration to the associated queueing regime, namely where the queue to each station is cleared by each transmission. The resulting simplified model includes an associated characterisation of noise and uncertainty, well suited for control design.

Secondly, we develop a novel nonlinear control design inspired by the solution to an associated proportional fair optimisation problem. The controller compensates for plant nonlinearities and so can be used for the full envelope of operation. The robust stability of the closed-loop system is analysed and the selection of control design parameters discussed.

Thirdly, we develop an implementation of the nonlinear control design and use this to present a performance evaluation using both simulations and experimental measurements. Figure 1(b) shows typical results obtained from a hardware testbed located in an office environment. It can be seen that the one-way delay is low, at around 2ms, while the send rate is high, at around 500Mbps (this data is for an 802.11ac downlink using three spatial streams and MCS 9). Increasing the send rate further leads to sustained queueing at the AP and an increase in delay, but the results in Figure 1(b) illustrate the practical feasibility of operation in the regime where the rate is maximised subject to the constraint that sustained queueing is avoided. As part of this evaluation we briefly compare the performance of the new controller with TCP Cubic, the default congestion control used in Linux and Android, and TCP BBR, a new protocol recently proposed by Google with the aim of achieving high rate and low delay. Note that both of these target operation in the general internet and so cannot exploit the specialisation possible with the edge architecture considered here,.

II. Related Work

In recent years there has been an upsurge in interest in userspace transports due to their flexibility and support for innovation combined with ease of rollout. This has been greatly facilitated by the high efficiency possible in userspace with the support of modern kernels. Notable examples of new transports developed in this way include Google QUIC [3], UDT [4] and Coded TCP [5]–[7]. ETSI has also recently set up a working group to study next generation protocols for 5G [8]. The use of performance enhancing proxies, including in the context of WLANs, is also not new e.g. RFC3135 [9] provides an entry point into this literature. However, none of these exploit the use of aggregation in WLANs to achieve high rate, low delay communication.

Interest in using aggregation in WLANs pre-dates the development of the 802.11n standard in 2009 but has primarily focused on analysis and design for wireless efficiency, managing loss etc. For a recent survey see for example [10]. The literature on throughput modelling of WLANs is extensive but much of it focuses on so-called saturated operation, where transmitters always have a packet to send, see for example [11] for early work on saturated throughput modelling of 802.11n with aggregation. When stations are not saturated (so-called finite-load operation) then for WLANs which use aggregation (802.11n and later) most studies resort to the use of simulations to evaluate performance due to the complex interaction between arrivals, queueing and aggregation with CSMA/CA service. Notable exceptions include [12], [13] which adopt a bulk service queueing model that assumes a fixed, constant level of aggregation and [14] which extends the finite load approach of [15] for 802.11a/b/g but again assumes a fixed level of aggregation.

While measurements of round-trip time might be used to estimate the onset of queueing and adjust the send rate, it is known that this can be inaccurate when there is queueing in the reverse path [16]. Furthermore, using RTT to detect queueing is known to give inaccurate results in 802.11 networks [17]. Accurately measuring one-way delay is also known in general to be challenging[1]. In contrast, the number of packets aggregated in a frame is relatively easy to measure accurately and reliably at the receiver, as already noted.

TCP BBR [18] is currently being developed by Google and this also targets high rate and low latency, although not specifically in edge WLANs. The BBR algorithm tries to estimate the bottleneck bandwidth and adapt the send rate accordingly to try to avoid queue buildup. The delivery rate in BBR is defined as the ratio of the in-flight data when a packet departed the server to the elapsed time when its ACK is received. This may be inappropriate, however, when the bottleneck is a WLAN hop since aggregation can mean that increases in rate need not correspond to increases in delay plus a small queue at the AP can be benificial for aggregation and so throughput.

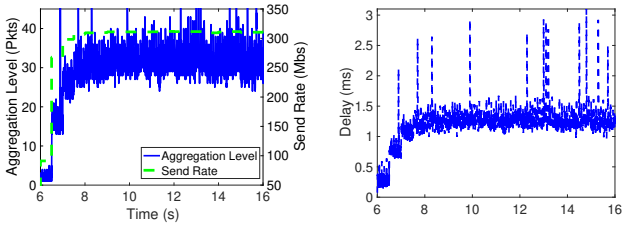
The closest related work is [1], which introduced the idea of using aggregation to regulate send rate in WLANs. They use a linear PI controller but do not provide any stability analysis. They confine consideration to a small operation envelope where the system exhibits linear behaviour.

Control theoretic analysis of WLANs has received relatively little attention in the literature, and has almost entirely focussed on MAC layer resource allocation, see for example [19]–[22] and references therein. In contrast, there exists a substantial literature on control theoretic analysis of congestion control at the transport layer, see for example the seminal work in [23], [24]. However, this has mainly focussed on end-to-end behaviour in wired networks with queue overflow losses and has largely ignored detailed analysis of operation over WLANs. This is perhaps unsurprising since low delay operation at the network edge has only recently come to the fore as a key priority for next generation networks.

III. Brief Overview Of 802.11ac

WLANs based on the 802.11 family of standards are ubiquitous at the network edge. They use CSMA/CA to share access

[1]The impact of clock offset and skew between sender and receiver applies to all network paths. In addition, when a wireless hop is the bottleneck then the transmission delay can also change significantly over time depending on the number of active stations e.g. if a single station is active and then a second station starts transmitting the time between transmission opportunities for the original station may double.



(a) (b)

Fig. 2. Example time histories of frame aggregation level and packet delay as send rate is varied. MCS 9, NSS 1, NS3 (see Section VII for details).

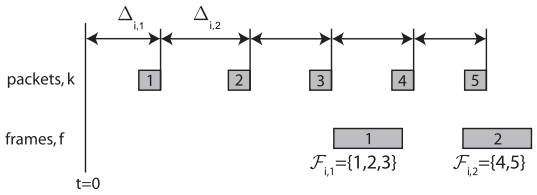


Fig. 3. Illustrating notation used. Packets arriving at the AP for transmission to station i are indexed $k = 1, 2, \ldots$ with the time between packet $k-1$ and packet k being $\Delta_{i,k}$. Frames transmitted by the AP to station i are indexed $f = 1, 2, \ldots$ and the set of packets sent in frame f is $\mathcal{F}_{i,f}$.

to the wireless channel amongst the stations in a WLAN. Briefly, in CSMA/CA each station maintains a contention window (CW) variable. Time is slotted and when a station wishes to transmit it initialises a counter to a value uniformly at random in interval [0,CW-1], decreases this counter for each slot that the channel is detected to be idle and then transmits once the counter reaches zero. This countdown is paused when the channel is detected to be busy, and so the MAC slots at which the counter is decremented are variable size and tend to become longer as more stations share the channel.

Since 2009, when the 802.11n standard was introduced, all WLANs make use of aggregation to improve efficiency. That is, transmitted frames may carry multiple packets, thereby amortising the fixed PHY/MAC overheads over multiple packets. To facilitate aggregation packets destined for different stations are queued separately at the transmitter. When a transmission opportunity occurs a frame is typically formed by aggregating the queued packets, up to the maximum allowed level N_{max} (typically 64 packets or 5.5ms frame duration, whichever is smallest), and so the level of aggregation used is closely linked to the size of the queue backlog at the transmitter.

Most modern WLANs now use 802.11ac, an extension of 802.11n that allows use of MIMO (transmission and reception using multiple antennas), higher modulation and coding scheme (MCS) rates (the rates at which data in a wireless frame is transmitted) and wider wireless channel widths of up to 160MHz. With N_a antennas, a transmitter can use up to N_a spatial streams for transmission i.e. the number of spatial streams (NSS) can vary from 1 up to N_a. The MCS rates are indexed form 0 to 9, e.g. with an 80MHz channel, long guard interval and N_a=NSS=3 the MCS rates vary from 87.8 to 1170Mbps.

IV. Modelling Aggregation Level & Delay

Figure 2 shows example time histories of the frame aggregation level and delay as the send rate is increased from around 50 to 300Mbps. These illustrate a number of features that will be of interest to us. Firstly, observe in Figure 2(a) that the send rate is updated every 0.5s and held constant in between updates, and this is the update interval that we will use in our control design. While the send rate varies relatively slowly it can be seen that there are rapid short-term fluctuations in the aggregation level about a value that roughly tracks the send rate. Observe that the magnitude of the fluctuations varies with the send rate e.g they are significantly lower in the early part of the time history around 6-6.5s, where the send rate is lower, than from 10s onwards. Hence, for control design we are interested in modelling the dependence on send rate of both the mean aggregation level (where is mean is taken over the short-term fluctuations) and the magnitude of the fluctuations in aggregation level.

With regard to delay, it can be seen from Figure 2(b) that the delay exhibits strikingly similar behaviour to the aggregation level. This is not by accident since, as we will see, both are intimately related.

In this section we develop simple analytic models, suitable for control design, of the dependence on send rate of the mean aggregation level and delay. We also characterise the dependence of the fluctuations in aggregation level with send rate and network configuration.

A. Basic Setup

We consider downlink transmissions in a WLAN (so no collisions) with n client stations indexed by $i = 1, 2, \ldots, n$. Index the packets arriving at the AP for transmission to station i by $k = 1, 2, \ldots$ and let $\Delta_{i,k}$ denote the inter-arrival time between packet $k-1$ and packet k. Recall that we control the packet sender and so for simplicity we assume that this uses packet pacing. That is, the sender aims to transmit packets with fixed spacing, although end host constraints typically mean that this aim is only approximately achieved and the packet spacing has some jitter. We can therefore assume that the $\Delta_{i,k}$ are i.i.d. with $E[\Delta_{i,k}] = \Delta_i = 1/x_i$ where x_i is the send rate to station i in packets/sec.

Packets are transmitted to station i by the AP within 802.11 frames. Index these frames by $f = 1, 2 \ldots$ ($f = 1$ is the first frame sent, and so on) and let $\mathcal{F}_{i,f} \subset \{1, 2, \ldots\}$ denote the set of packets aggregated within frame f transmitted to station i. Then $N_{i,f} = |\mathcal{F}_{i,f}|$ is the number of packets aggregated. Since a minimum of one packet must be contained within a frame and a maximum of N_{max} (typically 32 or 64 packets) then $1 \leq N_{i,f} \leq N_{max}$. The setup is illustrated in Figure 3.

B. Frame Transmission Timing

Suppose, for simplicity, that all packets are of length l bits (this can be easily relaxed). The airtime used by frame f transmitted to station i is then given by

$$T_{air,i,f} := T_{oh,i,f} + \frac{l + l_{oh}}{R_{i,f}} N_{i,f} \qquad (1)$$

where $R_{i,f}$ is the PHY data rate used to transmit the payload of the frame, $T_{oh,i,f}$ is the time used for transmission overheads which do not depend on the aggregation level (namely, CSMA/CA channel access, PHY and MAC headers plus transmission of the ACK by the receiver) and l_{oh} is the MAC framing overhead (in bits) for each packet in the frame.

Assume that the AP transmits frames to the n client stations in a round-robin fashion. We will also assume that the packet arrival rate is high enough that $N_{i,f} \geq 1$, $i = 1, \ldots, n$ i.e. the AP transmits at least one packet to each station in every round. This is reasonable since our primary interest here is in the high rate operation needed for next generation edge-assisted applications and $N_{i,f} \geq 1$ is ensured for sufficiently high rates (we give a lower bound on the rate needed at the end of the next section). Then the duration Ω_f of round f is given by,

$$\Omega_f = \sum_{j=1}^{n} T_{air,j,f} = C_f + \sum_{j=1}^{n} \frac{(l + l_{oh})}{R_{j,f}} N_{j,f} \tag{2}$$

where $C_f = \sum_{j=1}^{n} T_{oh,j,f}$. Index stations by the order in which they are serviced by the AP scheduler, i.e. within a round the i'th frame transmitted is to station i. In general, the interval $\Omega_{i,f}$ between transmission of frames f and $f+1$ to station i is not equal to Ω_f, but under reasonable assumptions $\Omega_{i,f}$ has the same distribution as Ω_f i.e. $\Omega_{i,f} \sim \Omega_f$.

In more detail, we have that

$$\begin{aligned}\Omega_{i,f} = &\sum_{j=i}^{n} (T_{oh,j,f} + \frac{l + l_{oh}}{R_{j,f}} N_{j,f}) \\ &+ \sum_{j=1}^{i-1} (T_{oh,j,f+1} + \frac{l + l_{oh}}{R_{j,f+1}} N_{j,f+1})\end{aligned} \tag{3}$$

The fixed CSMA/CA overhead $T_{oh,j,f}$ associated with channel access etc is i.i.d across stations j and frames f by virtue of the 802.11 MAC operation (fluctuations in $T_{oh,j,f}$ are due to the CSMA/CA channel access which is uniformly distributed between 0 and $CW - 1$ MAC slots, where CW is the 802.11 contention window). We therefore have that $\sum_{j=i}^{n} T_{oh,j,f} + \sum_{j=1}^{i-1} T_{oh,j,f+1} \sim C_f$. Assume the channel is stationary so that the MCS rate $R_{i,f}$ is identically distributed across frames f (but of course may vary amongst stations). Assume also that changes in the distribution of the aggregation level $N_{i,f}$ occur at a much slower time-scale than an AP scheduler round so that $N_{i,f}$ and $N_{j,f+1}$ can be approximated as being identically distributed. This is the key modelling approximation that we make but, as will see later, the control actions we consider which change the distribution of $N_{i,f}$ occur on a time-scale of 0.5-1s, whereas a scheduler round takes no more than 2-3ms, so this assumption is reasonable. It then follows that $\Omega_{i,f} \sim \Omega_f$.

Assuming $N_{i,f}$ and $R_{i,f}$ are independent we now have that

$$\begin{aligned}E[\Omega_{i,f}] = E[\Omega_f] &= c + \sum_{j=1}^{n} \frac{l + l_{oh}}{\mu_{R_j}} E[N_{j,f}] \\ &= c + \boldsymbol{w}^T E[\boldsymbol{N}_f]\end{aligned} \tag{4}$$

where $c := E[C_f] = nE[T_{oh,j,f}]$, $\mu_{R_i} := 1/E[\frac{1}{R_{i,f}}]$ (note that in general[2] $E[\frac{1}{R_{i,f}}] \neq 1/E[R_{i,f}]$), $\boldsymbol{w} = (\frac{l+l_{oh}}{\mu_{R_1}}, \ldots, \frac{l+l_{oh}}{\mu_{R_n}})^T$, $\boldsymbol{N}_f = (N_{1,f}, \ldots, N_{n,f})^T$.

C. Mean Aggregation Level

To model the aggregation behaviour we proceed as follows. Recall $\Omega_{i,f}$ is the interval between transmission of frame f and frame $f+1$ to station i. Let $P_{i,f}$ denote the number of packets arriving at the AP during this interval. When the time between packets is constant with $\Delta_{i,k} = \Delta_i$ then $P_{i,f} = \Omega_{i,f}/\Delta_i$ (ignoring quantisation effects) but, as already noted, in general we expect some jitter between packet arrivals even when the sender paces its transmissions.

These packets are buffered in a queue at the AP until they can be transmitted. Letting $q_{i,f}$ denote the queue occupancy immediately after frame f is transmitted then $q_{i,f+1} = [q_{i,f} + P_{i,f} - N_{i,f+1}]^+$. It is reasonable to suppose that the AP aggregates as many as possible of these packets queued for transmission into the next frame f, in which case $N_{i,f} = \min\{q_{i,f} + P_{i,f}, N_{max}\}$ and

$$q_{i,f+1} = [q_{i,f} + P_{i,f} - \min\{q_{i,f} + P_{i,f}, N_{max}\}]^+ \tag{5}$$

$$= [q_{i,f} + P_{i,f} - N_{max}]^+ \tag{6}$$

There are three operating regimes to consider:

1) Firstly, when $E[P_{i,f}] > N_{max}$ then the queue is unstable. The queue occupancy grows and so $N_{i,f+1} = N_{max}$ eventually for all frames f. This regime is not of interest in the present work where our focus is on low delay operation.
2) Secondly, our main interest is in the regime where the queue backlog remains low i.e. $P_{i,f} < N_{max}$. The queue is cleared by each frame transmission so $q_{i,f+1} = 0$ and $N_{i,f+1} = P_{i,f}$.
3) Thirdly, there is the transition regime between regimes one and two where $E[P_{i,f}] < N_{max}$ but $P_{i,f}$ may sometimes be greater than N_{max} and $q_{i,f+1}$ may be non-zero.

In regime two, $N_{i,f+1} = P_{i,f}$. Taking expectations $E[N_{i,f+1}] = E[P_{i,f}] = E[\Omega_{i,f}]/E[\Delta_{i,k}]$ (by renewal-reward theory since the $\Delta_{i,k}$ are i.i.d and independent of $\Omega_{i,f}$). Let $\mu_{\boldsymbol{N}} = (\mu_{N_1}, \ldots, \mu_{N_n})^T$ with $\mu_{N_i} := E[N_{i,f+1}]$, and recall that $x_i := 1/E[\Delta_{i,k}] = 1/\Delta_i$ is the send rate of station i. Then substituting from (4) it follows that $\mu_{\boldsymbol{N}} = (c + \boldsymbol{w}^T \mu_{\boldsymbol{N}})\boldsymbol{x}$. Rearranging yields

$$\mu_{\boldsymbol{N}} = \frac{c\boldsymbol{x}}{1 - \boldsymbol{w}^T \boldsymbol{x}} \tag{7}$$

where $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ is the vector of station send rates.

To simplify the analysis we assume that the third regime can be lumped with regime two[3]. In regime three $E[P_{i,f}] > N_{max}$ and $E[N_{i,f+1}] = N_{max}$. Incorporating the N_{max} constraint

[2]Indeed, to first-order $E[\frac{1}{R_{i,f}}] \approx \frac{1}{E[R_{i,f}]} + \frac{Var(R_{i,f})}{E[R_{i,f}]^3}$

[3]Our measurements in Section IV-F support the validity of this simplifying assumption. In practice it amounts to assuming that the system transitions quickly between operating regimes one and two, i.e. regime three is only transient.

into (7) gives the following expression for the mean aggregation level,

$$\mu_{\boldsymbol{N}} = \Pi \circ \frac{c\boldsymbol{x}}{1-\boldsymbol{w}^T\boldsymbol{x}} = \Pi \circ \boldsymbol{F}(\boldsymbol{x}) \tag{8}$$

where Π denotes projection onto interval $[1, N_{max}]$ and $\boldsymbol{F}(\boldsymbol{x}) := \frac{c\boldsymbol{x}}{1-\boldsymbol{w}^T\boldsymbol{x}}$. Note that $x_i \geq 0$, $i = 1\ldots, n$ and $\boldsymbol{w}^T\boldsymbol{x} < 1$ are required for rate vector $\boldsymbol{x}$ to be feasible and so $\boldsymbol{F}(\boldsymbol{x}) \geq \boldsymbol{0}$ (element-wise). Also that $x_i \geq 1/(c+w_i) \geq (1-\sum_{i\neq j} w_i x_i)/(c+w_i)$, $i = 1\ldots, n$ is sufficient to ensure that $\boldsymbol{F}(\boldsymbol{x}) \geq \boldsymbol{1}$.

D. Mean Delay

Recall $\mathcal{F}_{i,f} \subset \{1, 2, \ldots\}$ is the set of packets in frame f sent to station i and that these packets arrive with inter-arrival times $\Delta_{i,k}$, $k \in \mathcal{F}_{i,f}$. In operating regime two (see above), the queue is cleared after each transmission. Hence, the first packet in frame f arrives to an empty queue and must wait $\sum_{k\in\mathcal{F}_{i,f}} \Delta_{i,k}$ seconds before the last packet in the frame arrives at the AP and so becomes available for aggregation. The delay experienced by the first packet in frame f (and so by all other packets sharing this frame) is at most $\sum_{k\in\mathcal{F}_{i,f}} \Delta_{i,k}$. This upper bound is attained if the frame is transmitted right before arrival of the first packet sent in the next frame $f+1$ since if frame f was transmitted later then this packet would have been added to frame f. That is, mean packet delay at the AP is upper bounded by,

$$\mu_{T_i} = E[\sum_{k\in\mathcal{F}_{i,f}} \Delta_{i,k}] = E[N_{i,f}]\Delta_i = \frac{\mu_{N_i}}{x_i} \tag{9}$$

$$= \max\{\min\{\frac{c}{1-\boldsymbol{w}^T\boldsymbol{x}}, \frac{N_{max}}{x_i}\}, \frac{1}{x_i}\} \tag{10}$$

where recall $N_{i,f} = |\mathcal{F}_{i,f}|$ is the number of packets in the frame and $x_i = 1/\Delta_i$.

E. Invertibility of Map From Rate To Aggregation Level

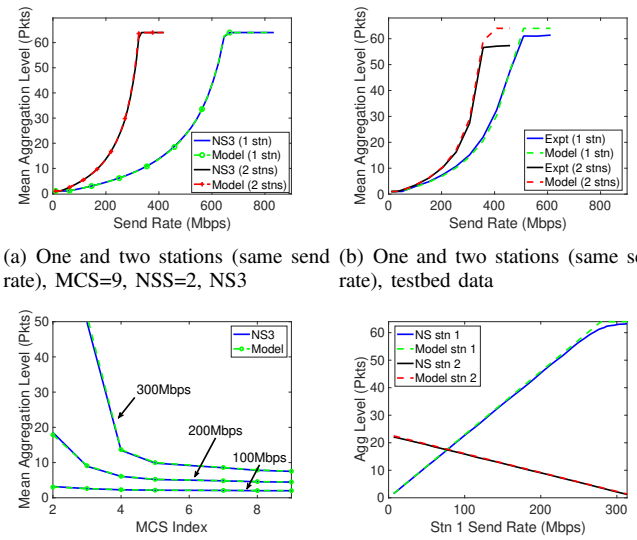
Observe that $\boldsymbol{F}(\boldsymbol{x})$ is monotonically increasing for feasible rate vectors $\boldsymbol{x}$ since $\frac{\partial F_i'(\boldsymbol{x})}{\partial x_i} = \frac{c}{(1-\boldsymbol{w}^T\boldsymbol{x})^2}(1-\boldsymbol{w}^T\boldsymbol{x}+w_i x_i) > 0$ and $\frac{\partial F_i'(\boldsymbol{x})}{\partial x_j} = \frac{cw_i x_i}{(1-\boldsymbol{w}^T\boldsymbol{x})^2} > 0$ when $x_i \geq 0$ and $\boldsymbol{w}^T\boldsymbol{x} < 1$. Hence, $\boldsymbol{F}(\boldsymbol{x})$ is one-to-one and so invertible. In particular,

$$\boldsymbol{F}^{-1}(\mu_{\boldsymbol{N}}) = \frac{\mu_{\boldsymbol{N}}}{c+\boldsymbol{w}^T\mu_{\boldsymbol{N}}} \tag{11}$$

and it can be verified that $\boldsymbol{F}(\boldsymbol{F}^{-1}(\mu_{\boldsymbol{N}})) = \mu_{\boldsymbol{N}}$.

Given rate vector $\boldsymbol{x}$ we can therefore obtain the corresponding aggregation level from $F(\boldsymbol{x})$ and, conversely, given aggregation level vector $\mu_{\boldsymbol{N}}$ we can obtain the corresponding rate vector from $\boldsymbol{F}^{-1}(\mu_{\boldsymbol{N}})$. This will prove convenient in the analysis below since it means we can freely change variables between $\boldsymbol{x}$ and $\mu_{\boldsymbol{N}}$. For example, substituting $\boldsymbol{x} = \boldsymbol{F}^{-1}(\mu_{\boldsymbol{N}})$ the term $\frac{c}{1-\boldsymbol{w}^T\boldsymbol{x}}$ in the mean delay (10) can be expressed equivalently in terms of $\mu_{\boldsymbol{N}}$ as,

$$\frac{c}{1-\boldsymbol{w}^T\boldsymbol{x}} = c + \boldsymbol{w}^T\mu_{\boldsymbol{N}} \tag{12}$$



(a) One and two stations (same send rate), MCS=9, NSS=2, NS3

(b) One and two stations (same send rate), testbed data

(c) One station, MCS and send rate varied, NSS=3, NS3

(d) Two stations with different send rates and MCSs. MCS 9 for station 1, MCS 3 for station 2, NSS=1, NS3

Fig. 4. Comparison of model (8) with measurements . Data is shown for sending UDP packets to one and two client stations, the same send rate being used to all stations and indicated on the x-axes of the plots. Plots (a),(c),(d) compare the mean aggregation level measured from NS3 simulations with the model, plot (b) compares measurements from an experimental testbed, see Section VII for details. In (a)-(c) when there are two stations they have the same send rate, in (d) the stations have different send rates: the send rate to station 1 increases from 5 to 310Mbps while the send rate to station 2 decreases from 100 to 5Mbps.

F. Validation Of Mean Model

As partial validation of the mean aggregation level model (8), in Figure 4(a) we compare its predictions against measurements from the NS3 detailed packet level simulator as the send rate is varied. Data is shown for the case of a single client station and when there are two client stations receiving with the same send rate. The values of the model parameters c and w are derived from the 802.11ac MAC/PHY settings (80MHz channel, MCS 9, NSS 2). It can be see that the model is in remarkably good agreement with the simulation data. We also collected measurements of aggregation level vs send rate in an experimental testbed, see Section VII for details. Figure 4(b) compares these experimental measurements against the model predictions[4] and again it can be seen that there is excellent agreement between the model and the measurements.

The model (8) predicts that the aggregation level scales as the reciprocal of $1-\boldsymbol{w}^T\boldsymbol{x} = 1-\sum_{i=1}^n L/\mu_{R_i}$. Figure 4(c) compares the model predictions as the MCS rate μ_{R_i} is varied (for the 802.11ac setting used $\mu_{R_i} = 87.8$Mbps at MCS index 0 increasing to 1170Mbps at MCS index 9). The model also predicts that the ratio of the aggregation level of two stations is proportional to the ratio of their send rates and this behaviour is evident in Figure 4(d) which plots the aggregation level for two stations when the send rate to the

[4]802.11ac settings: NSS=3, 80Mhz channel, the MCS used fluctuates over time due to the action of the 802.11ac rate controller and so an average value is used. The model c and w parameter values used in Figure 4(b) are $c = 270\mu s$, $\mu_R = 585$Mbps for the one station data and $c = 320\mu s$, $\mu_R = 850$Mbps for the two station data.

first station is increased from 5 to 310Mbps while that to the second station is decreased from 100 to 5Mbps.

In summary, the model (8) is in good agreement with measurements with regard to the dependence of aggregation level on overall send rate, MCS and ratio of station send rates.

G. Fluctuations Around Mean

1) Approximate Model: Equation (7) models the relationship between the arrival rate $\boldsymbol{x}$ and the mean aggregation level $\mu_{\boldsymbol{N}}$ when operating in regime two. We can also obtain an approximate model, useful for control design, of the fluctuations $\eta_{N_{i,f}} = N_{i,f} - \mu_{N_i}$ about the mean as follows. Neglecting the jitter in the packet inter-arrival times then the number of packets $P_{i,f}$ arriving at the AP during round f is approximately $\Omega_{i,f}x_i$. That is, the fluctuations in $P_{i,f}$ (and so $N_{i,f+1}$) are induced by fluctuations in the duration $\Omega_{i,f}$ of the scheduling round for station i. Neglecting the impact of the position of each station within a round then $\Omega_{i,f} \approx \Omega_f$ (this is exact in the case of a single station). Combining these we obtain the model

$$\boldsymbol{N}_{f+1} = (C_f + \boldsymbol{w}^T\boldsymbol{N}_f)\boldsymbol{x} \tag{13}$$

Since $\mu_{\boldsymbol{N}} = (c + \boldsymbol{w}^T\mu_{\boldsymbol{N}})\boldsymbol{x}$ it follows that

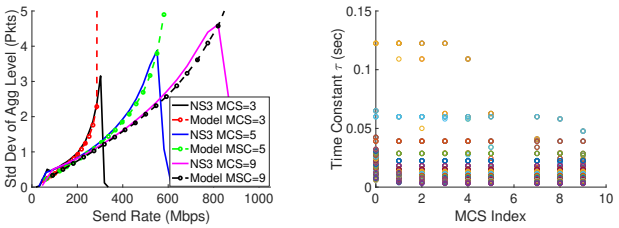
$$\eta_{\boldsymbol{N}_{f+1}} = \boldsymbol{x}\boldsymbol{w}^T\eta_{\boldsymbol{N}_f} + (C_f - c)\boldsymbol{x} \tag{14}$$

where $\eta_{\boldsymbol{N}_f} = [\eta_{N_{1,f}}, \ldots, \eta_{N_{n,f}}]^T$. Observe that $\eta_{\boldsymbol{N}_f}$ evolves according to first-order dynamics driven by i.i.d stochastic input $(C_f - c)\boldsymbol{x}$. In 802.11ac $C_f - c$ is a random variable uniformly distributed between 0 and 135μs (CWmin is 16 and a MAC slot is 9μs).

Figure 5(a) compares the predictions of the standard deviation of $\eta_{\boldsymbol{N}_f}$ calculated using the model (14) with measurements of the standard deviation of the aggregation level from NS3. Data is shown as the send rate and MCS rate are varied. It can be seen that the model predictions are in good agreement with the measurements except when the aggregation level hits its maximum value N_{max}, at which point the standard deviation of the measured data falls sharply to zero. That is, the model (14) is accurate within operating regime two but not in operating regime three, as expected.

Observe that the standard deviation of $\eta_{\boldsymbol{N}_f}$ increases with the send rate, which is intuitive. The main source of the fluctuations in $\boldsymbol{N}_f$ is the randomness in the channel access time associated with the operation of the CSMA/CA MAC. During a round where the channel access randomness leads to the round being of longer than average duration then more packets arrive than on average, with the number arriving increasing with the send rate. At the next round these packets form the next frame, which is therefore larger than average. The magnitude of the fluctuations $\eta_{\boldsymbol{N}_f}$ in the frame size therefore tends to increase with the send rate.

Note that larger frames also tends to make the next round longer than average since they take longer than average to transmit. This creates feedback whereby a random fluctuation in the duration of a round tends to create changes that persist for several rounds. It is this feedback that is reflected in the dynamics (14).



(a) (b)

Fig. 5. (a) Comparison of the standard deviation of $\eta_{\boldsymbol{N}_f}$ calculated using the model (14) with measurements from NS3, NSS=3. (b) time constant of dynamics (14) as the number n of stations is varied between 1 and 20, NSS is varied from 1 to 3 and the MCS index is varied fro 0 to 9 i.e. covering the full 802.11ac NSS/MCS range.

The measurement data in Figure 5(a) includes packet inter-arrival jitter of $\pm 6\mu$s. We also collected measurements for other values of jitter and found the standard deviation of $\eta_{\boldsymbol{N}_f}$ to be largely insensitive to the level of pacing jitter.

2) Time-Scale of Dynamics: The matrix $\boldsymbol{x}\boldsymbol{w}^T$ is rank one and has one zero eigenvalue $\boldsymbol{w}^T\boldsymbol{x} = \sum_{i=1}^n w_i x_i = \sum_{i=1}^n (l + l_{oh})x_i/\mu_{R_i}$ and an eigenvalue of zero with multiplicity $n - 1$. The time constant of the dynamics is therefore $\tau : - = E[\Omega_f]/\log(\boldsymbol{w}^T\boldsymbol{x})$. Substituting for $E[\Omega_f]$ and from (11) for the send rate $\boldsymbol{x}$ then gives $\tau = -(c+\boldsymbol{w}^T\boldsymbol{N})/\log(\boldsymbol{w}^T\boldsymbol{N}/(c+\boldsymbol{w}^T\boldsymbol{N}))$. Figure 5(b) plots the value of this time constant as the number of stations is varied from 1 to 20, NSS is varied from 1 to 3 and the MCS index from 0 to 9. For each configuration the aggregation level N is the minimum of N_{max} and the level for which the mean delay $\mu_{\boldsymbol{T}}$ is 5ms. It can be seen that the time constant is never more than about 0.12s, and tends to fall with increasing MCS rate. As we will see later, the online rate allocation algorithm we consider updates the packet send rate every Δ seconds, with Δ typically 0.5s or 1s and so operates at significantly longer time scales than the dynamics (14).

H. Measurement Noise & Main Source Of Model Uncertainty

1) Measurement Noise: The aggregation level $N_{i,f}$ can be observed at receiving station i via radiotap/prism libpcap packet headers [1]. As already noted, our online rate allocation algorithm updates the packet send rate every Δ seconds. We can therefore estimate the mean aggregation level μ_{N_i} via the empirical average $\tilde{\mu}_{N_i}(k) = \sum_f N_{i,f}$ of frames sent to station i over interval $[0, \Delta k)$, $k = 1, 2, \ldots$. As discussed in Section IV-G, $N_{i,f}$ fluctuates due to the MAC channel access randomness and this means that estimate $\tilde{\mu}_{N_i}$ is subject to significant measurement noise.

The model expressions (8) and (10) for the mean aggregation level and delay involve parameter $\boldsymbol{w} = (\frac{l+l_{oh}}{\mu_{R_1}}, \ldots, \frac{l+l_{oh}}{\mu_{R_n}})^T$. The packet size l and framing overhead l_{oh} are known and the MCS rate $R_{i,f}$ used to send frame f to station i can be observed by receiving station i (again via radiotap/prism libpcap packet headers) and we can therefore estimate μ_{R_i} via the empirical average $1/\tilde{\mu}_{R_i}$ of the $1/R_{i,f}$ over interval Δ. This estimate suffers from measurement noise induced by fluctuations in the empirical mean of $R_{i,f}$ over interval Δ. However, typically the channel is fairly stable over

short intervals and these fluctuations are small, thus the level of this measurement noise is low.

2) Model Uncertainty: The model expressions (8) and (10) also involve parameter $c = n\mu_{T_{oh}}$. The number n of stations to which downlink transmissions are ongoing is known but the mean channel access time $\mu_{T_{oh}}$ is harder to determine accurately since it cannot be measured directly (since we consider the transport layer we assume we do not have access to the MAC on the AP) and it depends on the channel state and so may be strongly affected by neighbouring WLANS, interference etc. Hence, only a fairly rough estimate of parameter c is generally available and this is the main source of model uncertainty.

V. Proportional Fair Low Delay Rate Allocation

A. Utility Fair Optimisation

Our interest is in achieving high rates while maintaining low delay at the AP. Formally, we consider the proportional fair low delay rate allocation that is the solution to the following optimisation P:

$$\max_{\boldsymbol{x}\in\mathbb{R}_+^n} \sum_{i=1}^{n} \log x_i \tag{15}$$

$$s.t.\ \mu_{T_i}(\boldsymbol{x}) \le \bar{T}, i = 1,\dots,n \tag{16}$$

$$\mu_{N_i}(\boldsymbol{x}) \le \bar{N}, i = 1,\dots,n \tag{17}$$

Constraint (16) ensures that the mean delay at the AP is no more than upper limit $\bar{T}$, where $\bar{T}$ is a QoS parameter. Constraint (17) ensures that we operate at an aggregation level no more than $\bar{N} < N_{max}$ and so the AP can clear the queue at each transmission opportunity i.e. there is no sustained queueing and we are operating in regime 2. Maximising objective (15) ensures utility fairness.

Substituting from (10) the constraints (16) can be written[5] as $\frac{c}{1-\boldsymbol{w}^T\boldsymbol{x}} \le \bar{T}$. Rearranging gives $c \le \bar{T}(1-\boldsymbol{w}^T\boldsymbol{x})$ i.e. $\boldsymbol{w}^T\boldsymbol{x} \le 1 - c/\bar{T}$. In this form it can be seen that the constraint is linear, and so convex. Similarly, substituting from (8) the constraints (17) can be written equivalently as $\frac{cx_i}{1-\boldsymbol{w}^T\boldsymbol{x}} \le \bar{N}$, $i = 1,\dots,n$. Rearranging gives $cx_i \le \bar{N}(1-\boldsymbol{w}^T\boldsymbol{x})$ i.e. $cx_i + \bar{N}\boldsymbol{w}^T\boldsymbol{x} \le \bar{N}$, which again is linear. Hence, optimisation P can be equivalently rewritten as optimisation P':

$$\max_{\boldsymbol{x}\in\mathbb{R}_+^n} \sum_{i=1}^{n} \log x_i \tag{18}$$

$$s.t.\ \boldsymbol{w}^T\boldsymbol{x} \le 1 - c/\bar{T} \tag{19}$$

$$cx_i + \bar{N}\boldsymbol{w}^T\boldsymbol{x} \le \bar{N}, i = 1,\dots,n \tag{20}$$

which is convex.

B. Characterising The Proportional Fair Solution

The Lagrangian of optimisation P' is $-\sum_{i=1}^{n}\log x_i + \theta(\boldsymbol{w}^T\boldsymbol{x} - (1-c/\bar{T})) + \sum_{i=1}^{n}\lambda_i(cx_i + \bar{N}\boldsymbol{w}^T\boldsymbol{x} - \bar{N})$ where θ and λ_i, $i = 1,\dots,n$ are multipliers associated with, respectively, (19) and (20). Since the optimisation is convex the KKT conditions are necessary and sufficient for optimality. Namely, an optimal rate vector $\boldsymbol{x}^*$ satisfies

$$-\frac{1}{x_i^*} + \lambda_i c + \sum_{j=1}^{n}\lambda_j\bar{N}w_i + \theta w_i = 0 \tag{21}$$

i.e.

$$x_i^* = \frac{1}{\lambda_i c + Dw_i} \tag{22}$$

where $D := (\bar{N}\sum_{j=1}^{n}\lambda_j + \theta)$.

Let $U = \{i : \mu_{N_i}(\boldsymbol{x}^*) < \bar{N}\}$ denote the set of stations for which the aggregation level is less than $\bar{N}$ at the optimal rate allocation. By complementary slackness $\lambda_i = 0$ for $i \in U$ and so $x_i^* = 1/(Dw_i)$. That is, $\mu_{N_i} = \frac{cx_i^*}{(1-\boldsymbol{w}^T\boldsymbol{x}^*)} = \frac{c}{D(1-\boldsymbol{w}^T\boldsymbol{x}^*)}\frac{1}{w_i}$. Observe that the first term is invariant with i and so the aggregation level of station $i \in U$ is proportional to $1/w_i = \mu_{R_i}/L$ i.e. to the mean MCS rate of the station. For stations $j \notin U$ the aggregation level $\mu_{N_j}(\boldsymbol{x}^*) = \bar{N}$.

Putting these observations together, it follows that

$$\mu_{N_i}(\boldsymbol{x}^*) = \min\{\frac{c}{D(1-\boldsymbol{w}^T\boldsymbol{x}^*)}\frac{1}{w_i}, \bar{N}\},\ i = 1,\dots,n \tag{23}$$

Assume without loss that the station indices are sorted such that $w_1 \ge w_2 \ge \cdots \ge w_n$. Then

$$\mu_{N_i}(\boldsymbol{x}^*) = \min\{\mu_{N_1}(\boldsymbol{x}^*)\frac{w_1}{w_i}, \bar{N}\},\ i = 2,\dots,n \tag{24}$$

Hence, once the optimal $\mu_{N_1}(\boldsymbol{x}^*)$ is determined we can find the optimal aggregation levels for the rest of the stations. With these we can then use inverse mapping (11) to recover the proportional fair rate allocation, namely $x_i^* = \mu_{N_i}/(c + \boldsymbol{w}^T\boldsymbol{\mu_N})$.

It remains to determine μ_{N_1}. We proceed as follows.

Lemma 1. *At an optimum $\boldsymbol{x}^*$ of P' then either (i) $\mu_{N_i}(\boldsymbol{x}^*) = \bar{N}$ for all $i = 1,\dots,n$ or (ii) $\mu_{T_i}(\boldsymbol{x}^*) = \bar{T}$ for all $i = 1,\dots,n$.*

Proof. We proceed by contradiction. Suppose at an optimum $\mu_{N_i}(\boldsymbol{x}^*) = \frac{cx_i^*}{1-\boldsymbol{w}^T\boldsymbol{x}^*} < \bar{N}$ for some i and $\mu_{T_i}(\boldsymbol{x}^*) = \frac{c}{1-\boldsymbol{w}^T\boldsymbol{x}^*} < \bar{T}$. Then we can increase x_i^* without violating the constraints (with this change $\frac{c}{1-\boldsymbol{w}^T\boldsymbol{x}^*}$ and $\frac{cx_i^*}{1-\boldsymbol{w}^T\boldsymbol{x}^*}$ will both increase, but since the corresponding constraints are slack if the increase in x_i^* is sufficiently small then they will not be violated). Hence, we can improve the objective which yields the desired contradiction since we assumed optimality of $\boldsymbol{x}^*$. Hence when $\mu_{N_i}(\boldsymbol{x}^*) < \bar{N}$ for at least one station then $\mu_{T_i}(\boldsymbol{x}^*) = \bar{T}$. Alternatively, $\mu_{N_i}(\boldsymbol{x}^*) = \bar{N}$ for all stations. □

It follows from Lemma 1 that $\mu_{N_1} = \min\{\bar{T}x_1^*, \bar{N}\}$. Substituting into (24) and combining with inverse mapping (11) it follows that

$$\mu_{N_1} = \min\{\bar{T}x_1^*, \bar{N}\} \tag{25}$$

$$\mu_{N_i} = \min\{\mu_{N_1}\frac{w_1}{w_i}, \bar{N}\},\ i = 2,\dots,n \tag{26}$$

$$\boldsymbol{x}^* = F^{-1}(\boldsymbol{\mu_N}) \tag{27}$$

The complete vector $\boldsymbol{\mu_N}$ can now be found by solving equations (25)-(27).

[5]Note that constraint (17) ensures $\mu_{N_i}(\boldsymbol{x}) \le \bar{N} < N_{max}$ and so $\mu_{T_i}(\boldsymbol{x}) < N_{max}/x_i$. Since our interest is primarily in applications requiring high rates we assume for simplicity that $\frac{c}{1-\boldsymbol{w}^T\boldsymbol{x}} \ge \frac{1}{x_i}$ although this could be added as the additional linear constraint $cx_i + \boldsymbol{w}^T\boldsymbol{x} \ge 1$ if desired.

C. Examples

We illustrate the nature of the proportional fair solution (25)-(27) via some brief examples.

Example 1: $\bar{N} = +\infty$: In this case there is no limit to the allowed aggregation level. It follows from (25)-(27) that $\mu_{N_i} = \mu_{N_1}\frac{\mu_{R_i}}{\mu_{R_1}} = T\frac{\mu_{R_i}}{\mu_{R_1}}x_1^*$ since $\bar{N}$ does not act to constrain the aggregation level. We know from Lemma 1 that the delay constraint is tight, $\mu_{T_i}(\boldsymbol{x}^*) = \bar{T}$. That is, by (12), that $c + \boldsymbol{w}^T\mu_{\boldsymbol{N}} = T$. Substituting for $\mu_{\boldsymbol{N}}$ this yields $c + \sum_{i=1}^n \frac{l+l_{oh}}{\mu_{R_i}}T\frac{\mu_{R_i}}{\mu_{R_1}}x_1^* = c + \frac{n(l+l_{oh})T}{\mu_{R_1}}x_1^* = T$ i.e $x_1^* = \frac{(T-c)\mu_{R_1}}{nLT}$ and so

$$\mu_{N_i} = \frac{(T-c)}{n(l+l_{oh})}\mu_{R_i} \tag{28}$$

i.e. the aggregation levels scale proportionally to the station MCS. The corresponding rates are

$$x_i = \frac{\mu_{N_i}}{c + \boldsymbol{w}^T\mu_{\boldsymbol{N}}} = \frac{(T-c)}{n(l+l_{oh})T}\mu_{R_i} \tag{29}$$

Recall that the mean airtime taken to send a frame to station i is $\frac{c}{n} + w_i x_i = \frac{c}{n} + \frac{(T-c)}{nT}$ which is the same for all stations i.e. the proportional fair rate allocation is an equal airtime one. The overall delay is $\mu_{T_i} = c + \boldsymbol{w}^T\mu_{\boldsymbol{N}} = c + \sum_{i=1}^n \frac{l+l_{oh}}{\mu_{R_i}}\frac{(T-c)}{n(l+l_{oh})}\mu_{R_i} = T$ i.e. equal to the target delay, as already noted.

Example 2: $T = +\infty$: Suppose now that the target delay $T = +\infty$ i.e. we seek the rate allocation that maintains the aggregation level at target value $\bar{N}$. This corresponds to the situation considered in [1]. From (25)-(27) we have that the proportional fair rate allocation yields aggregation levels that satisfy $\mu_{N_i} = \mu_{N_1}\frac{\mu_{R_i}}{\mu_{R_1}}$. Recall we assume the stations are ordered such that $w_1 \geq w_2 \geq \cdots \geq w_n$, and since $w_i = (l + l_{oh})/\mu_{R_i}$ it follows that $\mu_{R_1} \leq \mu_{R_2} \leq \cdots \leq \mu_{R_n}$ i.e. station 1 has the lowest MCS rate and n the highest. Hence, the aggregation level μ_{N_n} of station n is largest. We know that the aggregation levels of all stations are no more than $\bar{N}$, and in fact this limit will be attained since this maximises throughput. Hence, $\mu_{N_n} = \bar{N}$ and so

$$\mu_{N_i} = \bar{N}\frac{\mu_{R_i}}{\mu_{R_n}} \tag{30}$$

i.e. once again the aggregation levels scale proportionally to the station MCS. The corresponding rates also scale proportionally to the station MCS,

$$x_i = \frac{\mu_{N_i}}{c + \boldsymbol{w}^T\mu_{\boldsymbol{N}}} = x_n\frac{\mu_{R_i}}{\mu_{R_n}} \tag{31}$$

The mean airtime taken to send a frame to station i is therefore $\frac{c}{n} + w_i x_i = \frac{c}{+}x_n(l + l_{oh})/\mu_{R_n}$ which is the same for all stations i.e. the proportional fair rate allocation when $T = +\infty$ is again an equal airtime one.

VI. Inner-Outer Feedback Control

While we can solve convex optimisation P' using any standard online algorithm, it turns out that we can use the extra insight into the structure of the proportional fair solution gained in Section V-B to construct efficient and robust feedback-based approaches for solving online solution of P'.

In particular, from the solution structure (25)-(27) we have that the proportional fair rate allocation is also the solution to the following nested optimisation,

$$\boldsymbol{N}^* \in \arg\min_{\boldsymbol{N}\in\nu\boldsymbol{W},\nu\geq 1}(\nu - \min\{\bar{T}x_1^*(\boldsymbol{N}), \bar{N}\})^2 \tag{32}$$

$$s.t.\ \boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathbb{R}_+^n}\sum_{i=1}^n(\mu_{N_i}(\boldsymbol{x}) - \min\{N_i^*, \bar{N}\})^2 \tag{33}$$

where $\boldsymbol{W} = [1, \frac{w_1}{w_2}, \ldots, \frac{w_1}{w_n}]^T$. As we will shortly see, it turns out that this reformulation lends itself to an elegant feedback control implementation.

A. Inner Loop Controller

We begin by considering inner optimisation $\min_{\boldsymbol{x}\in\mathbb{R}_+^n}\sum_{i=1}^n(\mu_{N_i}(\boldsymbol{x}) - \min\{N_i, \bar{N}\})^2$. While the solution is trivial our interest is using the optimisation to derive a feedback update that is robust to model uncertainty. With this in mind therefore we change variables to $\boldsymbol{z} = \mu_{N_i}(\boldsymbol{x})$. Then the optimisation becomes $\min_{\boldsymbol{z}\in\mathbb{R}_+^n}\sum_{i=1}^n(z_i - \min\{N_i, \bar{N}\})^2$. Gradient descent now yields the following iterative update,

$$\boldsymbol{z}(k+1) = \boldsymbol{z}(k) + K_1(\boldsymbol{N}_{target} - \boldsymbol{z}(k)) \tag{34}$$

where the i'th element of vector $\boldsymbol{N}_{target}$ equals $\min\{N_i, \bar{N}\}$, K_1 is the step size and time is slotted with $\boldsymbol{z}(k)$ denoting the value in slot k. Using (11) we recover the rate from $\boldsymbol{x}(k) = \boldsymbol{F}^{-1}(\boldsymbol{z}(k)) = \boldsymbol{z}(k)/(c + \boldsymbol{w}^T\boldsymbol{z}(k))$.

We convert this update to a feedback control loop by substituting the measured aggregation level for $\boldsymbol{z}(k)$ over time interval $[0, \Delta k)$, $k = 1, 2, \ldots$, to obtain

$$\boldsymbol{z}(k+1) = \boldsymbol{z}(k) + K_1(\boldsymbol{N}_{target} - \tilde{\mu}_{\boldsymbol{N}}(k)) \tag{35}$$

$$\boldsymbol{x}(k) = \boldsymbol{F}^{-1}(\boldsymbol{z}(k)) = \boldsymbol{z}(k)/(c + \boldsymbol{w}^T\boldsymbol{z}(k)) \tag{36}$$

where $\tilde{\mu}_{\boldsymbol{N}}(k)$ is the measured mean aggregation level over time slot k when the send rate is held constant at $\boldsymbol{x}(k)$ over the slot.

It can be seen that update (35) is an integral controller that adjusts $\boldsymbol{z}$ to try to regulate $\boldsymbol{e} = \boldsymbol{N}_{target} - \tilde{\mu}_{\boldsymbol{N}}$ about zero. Namely, when $\boldsymbol{e} > 0$ then $\boldsymbol{z}$ is increased, which in turn tends to increase $\tilde{\mu}_{\boldsymbol{N}}$ and so decrease $\boldsymbol{e}$. Conversely, when $\boldsymbol{e} < 0$ then $\boldsymbol{z}$ is decreased which tends to increase $\boldsymbol{e}$. Since $\boldsymbol{z}$ etc are vectors (35)-(36) is a multiple-input multiple-output (MIMO) feedback loop. Since $\tilde{\mu}_{\boldsymbol{N}}$ is a nonlinear function of the send rate $\boldsymbol{x}(k)$ and $\boldsymbol{x}(k)$ is a nonlinear function of $\boldsymbol{z}(k)$ the feedback loop is also nonlinear.

1) Converting Between Slots and Frames: Update (35)-(36) is in term of time slots $[0, \Delta k)$, $k = 1, 2, \ldots$. To embed it within the real system we given rate $\boldsymbol{x}(k)$ over slot k we fix the sender inter-packet time between packets sent during interval $[0, \Delta k)$ to be $1/\boldsymbol{x}(k)$.

Conversely, given the sequence of observed individual frame aggregation levels $\boldsymbol{N}_f = [N_{1,f}, \ldots, N_{n,f}]$, $f = 1, 2, \ldots$ we calculate $\tilde{\mu}_{\boldsymbol{N}}(k)$ as the empirical mean of the frames received during interval $[0, \Delta k)$. That is,

$$\tilde{\mu}_{N_i}(k) = \frac{1}{|\Phi_i(k)|}\sum_{f\in\Phi_i(k)} N_{i,f} \tag{37}$$

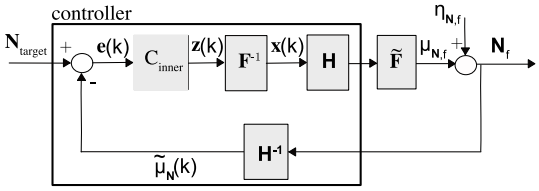


Fig. 6. Schematic of inner feedback loop. Controller updates occur at the start of time slots $[0, \Delta k)$, $k = 1, 2, \ldots$. Controller update $\boldsymbol{z}(k+1) = \boldsymbol{z}(k) + K_1 \boldsymbol{e}(k)$, nonlinear function $\boldsymbol{F}^{-1}$ is given by (11), $\boldsymbol{H}$ is a zero-order hold (i.e. holds the packet send rate constant at $\boldsymbol{x}(k)$ for packets sent during slot $[0, \Delta k)$), $\mu_{\boldsymbol{N}_f}$ is the mean frame aggregation level and $\eta_{\boldsymbol{N}_f}$ is the disturbance to this mean induced by MAC channel access randomness (see Section IV-G). $\boldsymbol{H}^{-1}$ maps from the sequence $\boldsymbol{N}_f = \mu_{\boldsymbol{N}_f} + \eta_{\boldsymbol{N}_f}$, $f = 1, 2 \ldots$ of individual frame aggregation levels to the empirical mean $\tilde{\mu}_{\boldsymbol{N}}(k)$ of the frame aggregation level over slots $[0, \Delta k)$, $k = 1, 2, \ldots$.

where $\Phi_i(k)$ is the set of frames received at station i during interval $[0, \Delta k)$.

Figure 6 shows schematically the resulting feedback loop corresponding to (35)-(36). $\boldsymbol{H}$ holds the sender inter-packet time equal to $1/\boldsymbol{x}(k)$ during controller update slot $[0, \Delta k)$. $\boldsymbol{H}^{-1}$ maps from the sequence of individual frame aggregation levels $\boldsymbol{N}_f$ to the empirical average aggregation level $\tilde{\mu}_{\boldsymbol{N}}$ over slots $[0, \Delta k)$, $k = 1, 2, \ldots$.

Since $\boldsymbol{N}_f = \mu_{\boldsymbol{N}_f} + \eta_{\boldsymbol{N}_f}$ the empirical mean$\tilde{\mu}_{\boldsymbol{N}}(k)$ over slot k is $\tilde{\boldsymbol{F}}_k + \eta_{\tilde{\mu}_{\boldsymbol{N}}}(k)$ where $\tilde{\boldsymbol{F}}_k = \frac{1}{|\Phi_i(k)|} \sum_{f \in \Phi_i(k)} \mu_{\boldsymbol{N}_f}$ and $\eta_{\tilde{\mu}_{\boldsymbol{N}}}(k) = \frac{1}{|\Phi_i(k)|} \sum_{f \in \Phi_i(k)} \eta_{\boldsymbol{N}_f}$. That is, $\tilde{\boldsymbol{F}}_k$ is the true mapping from rate to mean aggregation level at send rate $\boldsymbol{x}(k)$ and $\eta_{\tilde{\mu}_{\boldsymbol{N}}}(k)$ is the measurement noise. Due to mismatches between the model and the real system, in general $\tilde{\boldsymbol{F}}_k \neq \boldsymbol{F}$.

2) Linearising Action of Controller: It can be seen from Figure 6 that to compensate for the nonlinearity $\tilde{\boldsymbol{F}}_k$ we insert its (approximate) inverse $\boldsymbol{F}^{-1}$ so that $\tilde{\mu}_{\boldsymbol{N}}(k) = \tilde{\boldsymbol{F}}_k(\boldsymbol{F}^{-1}(\boldsymbol{z}(k)))$ and the system dynamics become

$$\boldsymbol{z}(k+1) = \boldsymbol{z}(k) + K_1(\boldsymbol{N}_{target} - \tilde{\boldsymbol{F}}_k(\boldsymbol{F}^{-1}(\boldsymbol{z}(k)))) \quad (38)$$

(neglecting the additive measurement noise $\eta_{\tilde{\mu}_{\boldsymbol{N}}}(k)$ for now). When $\tilde{\boldsymbol{F}}_k(\boldsymbol{F}^{-1}(\boldsymbol{z}(k))) \approx \boldsymbol{z}_k$ then the resulting linearised loop dynamics are $\boldsymbol{z}(k+1) \approx \boldsymbol{z}(k) + K_1(\boldsymbol{N}_{target} - \boldsymbol{z}(k))$. That is, the controller transforms the nonlinear system to have first-order linear dynamics.

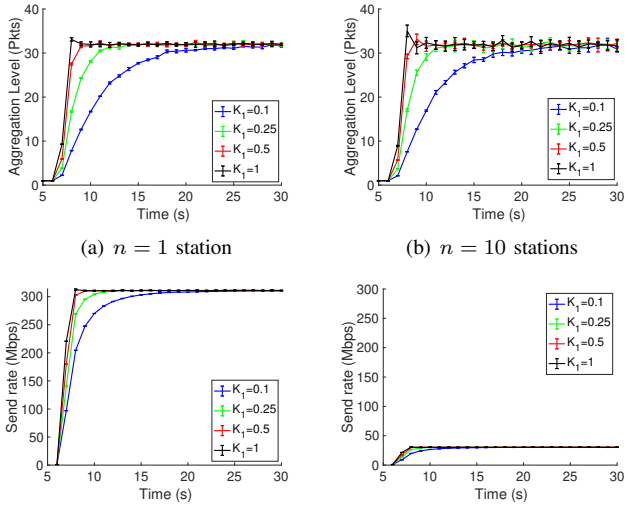
3) Robust Stability: Recall that the main source of model uncertainty is parameter c. That is, $\boldsymbol{F}(\boldsymbol{x}) = \frac{c\boldsymbol{x}}{1 - \boldsymbol{w}^T \boldsymbol{x}}$ whereas to a good approximation $\tilde{\boldsymbol{F}}_k(\boldsymbol{x}) = \Pi \circ \frac{\tilde{c}(k)\boldsymbol{x}}{1 - \boldsymbol{w}^T \boldsymbol{x}}$ with $\tilde{c}_k \neq c$ (recall projection Π captures the saturation constraint that $\tilde{\mu}_{\boldsymbol{N}}(k) \in [1, N_{max}]$). Hence, $\tilde{\boldsymbol{F}}_k(\boldsymbol{F}^{-1}(\boldsymbol{z}(k))) = \Pi \circ (\frac{\tilde{c}(k)}{c}\boldsymbol{z}(k))$ and dynamics (38) are

$$\boldsymbol{z}(k+1) = \boldsymbol{z}(k) + K_1(\boldsymbol{N}_{target} - \Gamma(k)\boldsymbol{z}(k)) \quad (39)$$

where $\Gamma(k) = diag\{\gamma_1(k), \ldots, \gamma_n(k)\}$ and $\gamma_i(k) = \frac{\Pi \circ (\tilde{c}(k) z_i(k)/c)}{z_i(k)}$.

Neglecting the input $\boldsymbol{N}_{target}$ for the moment, it is easy to see[6] that the dynamics $\boldsymbol{z}(k+1) = (I - \Gamma(k))\boldsymbol{z}(k)$ are exponentially stable provided $0 < \gamma_i(k) < 2$ for all $i = 1, \ldots, n$. Note that this stability holds for arbitrary time-variations in the $\gamma_i(k)$. Projection Π satisfies $0 \leq \frac{\Pi \circ z}{z} \leq 1$ and $\tilde{c}(k), c$ are both non-negative, so for stability it is sufficient that $\tilde{c}(k)/c < 2$. This condition is also necessary since for constant $\tilde{c}(k)$ the system will be unstable if this condition is violated.

[6]Try candidate Lyapunov function $V(k) = \boldsymbol{z}^T(k)\boldsymbol{z}(k)$. Then $V(k+1) = (I - \Gamma(k))^T(I - \Gamma(k))V(k)$ (since $\Gamma(k)$ is diagonal) and so is strictly decreasing when $0 < \gamma_i(k) < 2$.



(a) $n = 1$ station (b) $n = 10$ stations

(c) $n = 1$ station (d) $n = 10$ stations

Fig. 7. Impact of control gain K_1 on transient dynamics of aggregation level and send rate. Plots show average and standard deviation over 10 runs for each value of gain. NS3 simulation, setup as in Section VII: $N_{target} = 32$, NSS=1, MCS=9.

In summary, time-variations in the $\gamma_i(k)$ affect stability in a benign fashion and control parameter c can safely be larger than the (uncertain) plant gain $\tilde{c}(k)$ (as this reduces the loop gain) but should not be too much smaller (since this increases the loop gain).

Time-variations in the gains $\gamma_i(k)$, $i = 1, \ldots, n$ also affect regulation of the aggregation level at N_{target}. It can be seen that when $\gamma_i(k)$ is constant the equilibrium of dynamics (39) is $z_i(k) = N_{target}/\gamma_i(k)$. When variations in $\gamma_i(k)$ are sufficiently slow relative to the loop dynamics then $z_i(k)$ will still roughly track this equilibrium [25] although faster changes may lead to $z_i(k)$ only staying in a ball around it. Hence, when $\gamma_i(k) < 1$ the aggregation level tends to be larger than the desired value N_{target}, and vice versa when $\gamma_i(k) < 1$. Hence, adaptation of control parameter c to maintain $\gamma_i(k)$ close to 1 is desirable, and we will discuss this in more detail shortly.

4) Selecting Controller Gain K_1: Figure 7 plots the measured step response of the system aggregation level and send rate x as the gain K_1 and number of stations n are varied. This data is for a detailed packet-level simulation, see Section VII for details. It can be seen from Figures 7(a)-(b) that, as expected, the aggregation level convergence time falls as K_1 is increased although the response starts to become oscillatory for larger values of K_1. It can also be seen from these figures that step response is effectively invariant with the number of stations due to the linearising action of the controller. Figures 7(c)-(d) show the send rate time histories corresponding to Figures 7(a)-(b) and the impact of the nonlinearity $\tilde{F}_k$ relating aggregation level and send rate is evident with the send rate being an order of magnitude smaller for the same aggregation

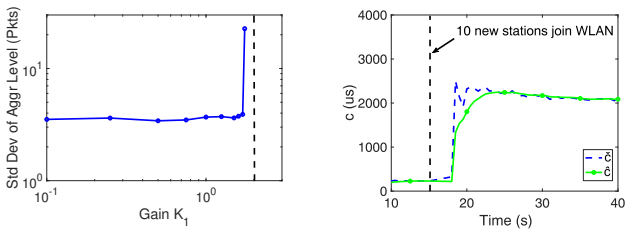


Fig. 8. (a) Impact of control gain K_1 on standard deviation of fluctuations in aggregation level. (b) Illustrating c estimator (40) tracking a sharp change in the number of stations from $n = 1$ to $n = 11$ at time 15s. NS3 simulation, setup as in Section VII: one client station, $N_{target} = 32$, NSS=1, MCS=9

level with $n = 10$ stations compared to with $n = 1$ station. Similar results are obtained when the MCS is varied.

Figure 8(a) plots the standard deviation of the frame aggregation level when the system is in steady-state vs the gain K_1. It can be seen that the controller starts to amplify the fluctuations in frame aggregation level as K_1 gets closer to the stability boundary at $K_1 = 2$ (indicated by the dashed line on the figure) but otherwise the standard deviation is insensitive to the choice of K_1. Recall that the fluctuations in the aggregation level are mainly induced by the randomness of the CSMA/CS channel access and occur on time-scales which are too short to be regulated by the controller, see Section IV-G.

In the remainder of this paper we select $K_1 = 0.5$ unless otherwise stated since this strikes a reasonable balance between response time and robustness to uncertainty in c (with $K_1 = 0.5$ the value of c can be out by a factor of 4, corresponding to a gain margin of 12 dB, and the system dynamics will remain stable).

5) Adapting c: The controller depends on parameter $c = n\mu_{T_{oh}}$. The average channel access time for each frame transmission is $CW/2 \times S$ where CW is the MAC contention window, typically 16 in 802.11ac, and S is the MAC slot duration in seconds. The PHY slot length is typically $9\mu s$, but the MAC slot duration can be significantly longer when other transmitters share the channel since the AP will defer access upon detecting the channel to be busy and it is this which makes it challenging to estimate $\mu_{T_{oh}}$.

Note that an exact value for c is not necessary since the feedback loop can compensate for uncertainty in c, i.e. an estimator that roughly tracks any large changes in c is sufficient. Recall that $\mu_{N_i} = \frac{cx_i}{1-\boldsymbol{w}^T\boldsymbol{x}}$, i.e. $c = \frac{\mu_{N_i}}{x_i}(1-\boldsymbol{w}^T\boldsymbol{x})$. Motivated by this observation we use the following as an estimator of c,

$$\hat{c}(k+1) = (1-\beta)\hat{c}(k) + \beta\check{c}(k) \tag{40}$$

with $\check{c}(k) := \frac{\tilde{\mu}_{N_1}(k)}{x_1(k)}(1 - \boldsymbol{w}^T\boldsymbol{x}(k))$, where β is a design parameter which controls the window over which the moving average is calculated (a typical value is $\beta = 0.05$).

Figure 8(b) illustrates the ability of this estimator to track a fairly significant change in the network conditions, namely 10 new stations joining the WLAN at time 15s and starting downlink transmissions. These new stations cause a change in c from a value of around $200\mu s$ to around $2200\mu s$ i.e. a change of more than an order of magnitude. It can be seen that estimator (40) tracks this large change without difficulty. We observe similar tracking behaviour for changes in MCS and also when the channel is shared with other legacy WLANs.

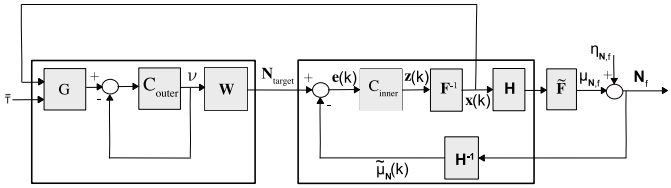


Fig. 9. Schematic of coupled feedback loops. $\bar{T}$ is the target delay, $G(\boldsymbol{x}(k)) = \min\{\bar{T}x_1(k), \bar{N}\}$, C_{outer} denotes the outer control update (43) and $\boldsymbol{W}$ update (44). Other quantities are as in Figure 6.

B. Outer Loop Controller

We turn now to the outer optimisation $\min_{\boldsymbol{N}\in\{\nu\boldsymbol{W},\nu\in[1,\infty)\}}(\nu - \min\{\bar{T}x_1^*(\boldsymbol{N}), \bar{N}\})^2$ in (32)-(33). The corresponding gradient descent update is

$$\nu(k+1) = \max\{\nu(k) - K_2(\nu(k) - G(\boldsymbol{x}^*(\boldsymbol{N}(k)))), 1\} \tag{41}$$

$$\boldsymbol{N}(k+1) = \nu(k+1)\boldsymbol{W}(k) \tag{42}$$

where $G(\boldsymbol{x}) = \min\{\bar{T}x_1, \bar{N}\}$ and $\boldsymbol{W}(k) = [1, \frac{w_1(k)}{w_2(k)}, \ldots, \frac{w_1(k)}{w_n(k)}]^T$. Step size K_2 and delay target T are design parameters and $x_1^*(\boldsymbol{N}(k))$ is the solution to optimisation (33) with $\boldsymbol{N}^* = \boldsymbol{N}(k)$.

Replacing $\boldsymbol{x}^*(\boldsymbol{N}(k))$ by $\boldsymbol{x}(k) = F^{-1}(\boldsymbol{z}(k))$ from the inner loop and projecting $\nu(k+1)\boldsymbol{W}red(k)$ onto interval $[0, \bar{N}]$ so that the input to the inner loop is well-behaved, then we obtain the following coupled feedback loops,

$$\nu(k+1) = \max\{\nu(k) + K_2(G(\boldsymbol{x}(k)) - \nu(k)), 1\} \tag{43}$$

$$\boldsymbol{N}_{target}(k+1) = \min\{\nu(k+1)\boldsymbol{W}(k), \bar{N}\} \tag{44}$$

$$\boldsymbol{z}(k+1) = \boldsymbol{z}(k) + K_1(\boldsymbol{N}_{target}(k) - \tilde{\mu}_{\boldsymbol{N}}(k)) \tag{45}$$

$$\boldsymbol{x}(k+1) = \boldsymbol{F}^{-1}(\boldsymbol{z}(k+1)) \tag{46}$$

This setup is shown schematically in Figure 9. It can be seen that we "bootstrap" from the inner loop and use $G(\boldsymbol{x}(k))$ as the set point for outer loop control variable $\nu(k)$. We then map from $\nu(k)$ to the target aggregation level N_{target} using $\nu(k)\boldsymbol{W}$. Since the first element W_1 of vector $\boldsymbol{W}$ equals 1 we can identify $\nu(k)$ with the target aggregation level for station 1 i.e. the station with lowest MCS rate, and the target aggregation levels of the other stations are proportional to $\nu(k)$.

1) Sufficient Conditions For Stability: Substituting from (39) the system dynamics (43)-(46) can be rewritten equivalently as,

$$\nu(k+1) = \max\{\nu(k) - K_2(\nu(k) - G(\boldsymbol{F}^{-1}(\boldsymbol{z}(k)))), 1\} \tag{47}$$

$$\boldsymbol{z}(k+1) = \boldsymbol{z}(k) + K_1(\min\{\nu(k)\boldsymbol{W}, \bar{N}\} - \boldsymbol{\Gamma}_k\boldsymbol{z}(k)) \tag{48}$$

Assume the dynamics of the inner $\boldsymbol{z}$ loop are much faster than those of the outer ν loop (e.g. by selecting $K_2 \ll K_1$) so that $\boldsymbol{z}(k) = \nu(k)\boldsymbol{W}$. Then the system dynamics simplify to

$$\begin{aligned}
&\nu(k+1)\\
&= \max\{\nu(k) - K_2(\nu(k) - \min\{\bar{T}\frac{\nu(k)}{c+\nu(k)nw_1}, \bar{N}\}), 1\} \quad (49)\\
&= \max\{\nu(k) - K_2(\nu(k) - \gamma_0(k)\bar{T}\frac{\nu(k)}{c+\nu(k)nw_1}), 1\} \quad (50)\\
&= \max\{(1 - K_2\frac{c+\nu(k)nw_1 - \gamma_0(k)\bar{T}}{c+\nu(k)nw_1})\nu(k), 1\} \quad (51)
\end{aligned}$$

where $0 < \gamma_0(k) \leq 1$ captures the impact of the $\bar{N}$ constraint i.e $\gamma_0(k)$ equals 1 when $\bar{T}\frac{\nu(k)}{c+\nu(k)nw_1} \leq \bar{N}$ and decreases as $\bar{T}\frac{\nu(k)}{c+\nu(k)nw_1}$ increases above $\bar{N}$. We have also used the fact that $\boldsymbol{w}^T\boldsymbol{W} = nw_1$.

We can gain useful insight into the behaviour of the system dynamics from inspection of (51). Namely, ignoring the constraints for the moment (i.e. $\gamma_0(k) = 1$ and $\nu(k) \geq 1$) and assuming that $0 < K_2 < 1$ then it can be seen that when $c + \nu(k)nw_1 - \bar{T} < 0$ then $1 - K_2\frac{c+\nu(k)nw_1-\bar{T}}{c+\nu(k)nw_1}) > 1$ and so $\nu(k+1)$ increases (since $\nu(k) \geq 1$). Hence $c + \nu(k)nw_1 - \bar{T}$ increases until it equals 0 or becomes positive. Conversely, when $c + \nu(k)nw_1 - \bar{T} > 0$ then $1 - K_2\frac{c+\nu(k)nw_1-\bar{T}}{c+\nu(k)nw_1}) < 1$ and $\nu(k+1)$ decreases. Hence, $c + \nu(k)nw_1 - \bar{T}$ decreases until it equals 0 to becomes negative. That is, the dynamics (51) force $c+\nu(k)nw_1-\bar{T}$ to either converge to 0 or oscillate about 0.

With the above in mind the impact of the constraints is now easy to see. When $\bar{T} > c + \bar{N}nw_1$ then the delay target is hit at an aggregation level above $\bar{N}$. It can be seen that $c + \nu(k)nw_1 - \bar{T} < 0$ for all admissible $\nu(k)$ and so $\nu(k)$ increases until it equals $\bar{N}$. When $\bar{T} < c + nw_1$ then the target delay is violated even when the aggregation level is the minimum possible $\nu(k) = 1$. It can be seen that $c + \nu(k)nw_1 - \bar{T} > 0$ for all admissible $\nu(k)$ and so $\nu(k)$ decreases until it equals 1.

To establish stability we need to show that persistent oscillations about $c + \nu(k)nw_1 - \bar{T} = 0$ cannot happen. We have the following lemma:

Lemma 2. *Suppose gain $0 < K_2 < 1$ and initial condition $1 \leq \nu(1) \leq \bar{N}$. Then for the dynamics (51) we have: (i) when $c + nw_1 < \bar{T} < c + \bar{N}nw_1$ then $\nu(k)$ converges to $(\bar{T} - c)/(nw_1)$, (ii) when $\bar{T} \geq c + \bar{N}nw_1$ then $\nu(k)$ converges to upper limit $\bar{N}$ and (iii) when $\bar{T} < c + nw_1$ then $\nu(k)$ converges to lower limit 1.*

Proof. Case(i): $c + nw_1 < \bar{T} < c + \bar{N}nw_1$. Try candidate Lyapunov function $V(k) = (c + \nu(k)nw_1 - \bar{T})^2/(nw_1)^2$. Letting $\nu^* = (\bar{T} - c)/nw_1$ then this can be rewritten as $V(k) = (\nu(k) - \nu^*)^2$ and since $c + nw_1 < \bar{T} < c + \bar{N}nw_1$ then $1 < \nu^* < \bar{N}$. In addition, to take care of gain $\gamma_0(k)$ we will show by induction that $\gamma_0(k) = 1$. By assumption $1 < \bar{T}\nu(1)/(c + \nu(1)nw_1) \leq \bar{N}$ and so $\gamma_0(1) = 1$. Suppose $\gamma_0(k) = 1$. Substituting from (51) it follows that

$$\begin{aligned}
V(k+1) &= (\nu(k+1) - \nu^*)^2\\
&= (\max\{(1 - K_2\frac{c+\nu(k)nw_1 - \bar{T}}{c+\nu(k)nw_1})\nu(k), 1\} - \nu^*)^2\\
&\overset{(a)}{\leq} ((1 - K_2\frac{c+\nu(k)nw_1 - \bar{T}}{c+\nu(k)nw_1})\nu(k) - \nu^*)^2\\
&= ((1 - K_2\frac{(c-\bar{T})/nw_1 + \nu(k)}{c+\nu(k)nw_1}nw_1)\nu(k) - \nu^*)^2\\
&= (\nu(k) - K_2\frac{\nu(k) - \nu^*}{c+\nu(k)nw_1}nw_1\nu(k) - \nu^*)^2\\
&\overset{(b)}{=} (1 - K_2\frac{\nu(k)nw_1}{c+\nu(k)nw_1})^2V(k)
\end{aligned}$$

where (a) follows because $\nu^* > 1$. Since $0 < K_2 < 2$ and $0 < \frac{\nu(k)nw_1}{c+\nu(k)nw_1} < 1$ it follows from (b) that $0 < (1-K_2\frac{\nu(k)nw_1}{c+\nu(k)nw_1})^2 < 1$ and so $V(k+1)$ is strictly decreasing unless $V(k) = 0$. Further, since $K_2 < 1$ then $\nu(k+1)$ has the same sign as $\nu(k)$ i.e. $\nu(k+1) > 0$. Putting these observations together, we have that $\nu(k+1)$ is closer than $\nu(k)$ to $\nu^* < \bar{N}$. Since $\nu(1) \leq \bar{N}$ then $\nu(2) < \bar{N}$, while when $\nu(k) \leq \bar{N}$ for $k > 1$ then $\nu(k+1) < \bar{N}$. So by induction $\nu(k) \leq \bar{N}$ for all $k \geq 1$ and thus $\gamma_0(k) = 1$ for all $k \geq 1$. Since $V(k+1) < V(k)$ when $V(k) > 0$ then $V(k)$ decreases monotonically to 0 i.e. the system converges to the point $c + \nu(k)nw_1 - \bar{T} = 0$ as claimed.

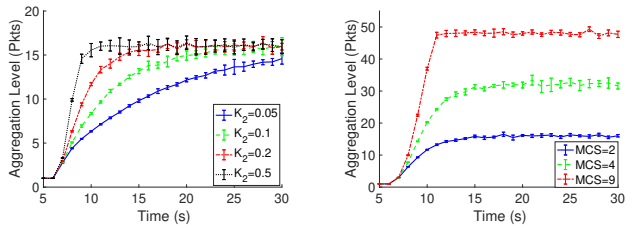
Cases (ii) and (iii). When $\bar{T} \geq c + \bar{N}nw_1$, respectively $\bar{T} < c + nw_1$, then $c + \nu(k)nw_1 - \bar{T} < 0$, respectively $c + \nu(k)nw_1 - \bar{T} > 0$ for all $1 \leq \nu(k) \leq \bar{N}$. The stated result now follows. □

Note that while the above analysis makes use of time-scale separation between $\boldsymbol{z}$ and ν so that $\boldsymbol{z}(k) = \nu(k)\boldsymbol{W}$, in practice we observe that the system is well behaved even when this assumption is violated and conjecture that Lemma 2 also applies in such cases.

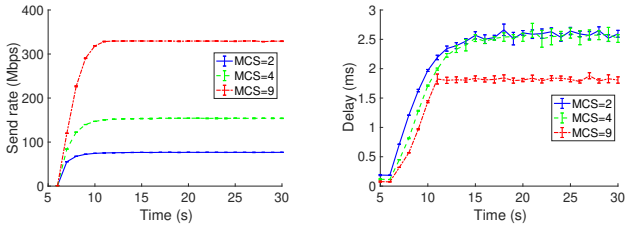
2) Selecting Control Gain K_2: Figure 10(a) plots the measured step response of the system aggregation level as the outer control gain K_2 is varied. It can be seen that the rise time falls with increasing gain, as expected. Although not shown on the plot to reduce clutter, we observe that for $K_2 \geq 1$ the response becomes increasing oscillatory suggesting that the sufficient condition for stability $K_2 < 1$ is in fact the stability boundary. In the rest of the paper we select $K_2 = 0.2$ as striking a reasonable compromise between responsiveness and robustness to uncertainty.

Figures 10(b)-(d) illustrate the adaptation by the outer feedback loop of N_{target} so as to regulate the delay about the target value $\bar{T}$. Figure 10(b) plots the aggregation level vs time, Figure 10(c) the send rate and Figure 10(d) the delay. Measurements are shown for three MCS values. It can be seen that as the MCS rate increases both the aggregation level and send rate increase while the delay is maintained close to the target value $\bar{T} = 2.5ms$[7]. This is as expected since

[7]The 802.11ac standard imposes a maximum frame duration of 5.5ms. In these tests with a single client station we use a target delay of 2.5ms so as to avoid hitting this upper limit on frame duration and thus allow the dynamics of the feedback loop to be seen more clearly.



(a) MCS 2, $\bar{T} = 2.5ms$ (b) $K_2 = 0.2$, $\bar{T} = 2.5ms$



(c) $K_2 = 0.2$, $\bar{T} = 2.5ms$ (d) $K_2 = 0.2$, $\bar{T} = 2.5ms$

Fig. 10. (a) Impact of outer loop gain K_2 on convergence time, (b) adapting N_{target} to regulate delay to below $\bar{T}$ as MCS is varied, (c), (d) send rate and delay measurements corresponding to (b). Plots show average and standard deviation over 10 runs for each value of gain. NS3, one client station, NSS=1, $\bar{T} = 2.5ms$, $\bar{N} = 48$.

the mean delay is just the mean duration of a scheduling round $c + \boldsymbol{w}^T \mu_{\boldsymbol{N}}$. As the MCS rate increases $\boldsymbol{w}$ decreases and so the aggregation level $\mu_{\boldsymbol{N}}$ can increase while keeping product $\boldsymbol{w}^T \mu_{\boldsymbol{N}}$ (which is the overall time to transmit the frame payloads) unchanged.

We can quickly verify the measurements as follows. For the network configuration in Figures 10(b)-(d) fixed overhead c is around 200μs. MCS index 2 with NSS=1 corresponds data rate 87.7Mbps, the packet size $l = 1500$B, overhead $l_{oh} = 48$B and from Figure 10(b) the aggregation level is approximately 16 packets, so $\boldsymbol{w}^T \mu_{\boldsymbol{N}} = (1500+48)\times 8 \times 16/87.7 \times 10^6 = 2.3$ms and adding c to this gives $\bar{T} = 2.5$ms. Similarly, MCS index 4 with NSS=1 corresponds to a data rate of 175.5Mbps and plugging this value into the previous expression along with aggregation level 23 packets again gives $\boldsymbol{w}^T \mu_{\boldsymbol{N}} = 2.3$ms. MCS index 9 with NSS=1 corresponds to data rate 390Mbps. At this data rate we hit the limit $\bar{N} = 48$ packets before delay target $\bar{T}$ is reached ($\boldsymbol{w}^T \mu_{\boldsymbol{N}} = 1.5$ms when the rate is 390Mbps and the aggregation level is 48 packets, adding $c = 200\mu$s to this gives a delay of 1.7ms as can be seen in Figure 10(d)).

VII. Experimental Measurements

A. Hardware & Software Setup

1) NS3 Simulator Implementation: We implemented the inner-outer controller in the NS3 packet-level simulator. Based on the received feedbacks it periodically configures the sending rate of `udp-client` applications colocated at a single node connected to an Access Point. Each wireless station receives a single UDP traffic flow at a `udp-server` application that we modified to collect frame aggregation statistics and periodically transmit these to the controller at intervals of Δ ms. We also developed a round-robin scheduler at the AP with separate queue for each destination, and we added new functions to let stations determine the MCS of each received frame together with the number of MPDU packets it contains. The maximum aggregation level permitted is N_{max}=64. We configured 802.11ac to use a physical layer operating over an $80MHz$ channel, VHT rates for data frames and legacy rates for control frames. The PHY MCS and the number of spatial streams NSS used can be adjusted. As validation we reproduced a number of the simulation measurements in our experimental testbed and found them to be in good agreement. The new NS3 code and the software that we used to perform experimental evaluations are available open-source[8].

2) Experimental Testbed: Our experimental testbed uses an Asus RT-AC86U Access Point (which uses a Broadcom 4366E chipset and supports 802.11ac MIMO with up to four spatial streams). It is configured to use the 5GHz frequency band with 80MHz channel bandwidth. This setup allows high spatial usage (we observe that almost always three spatial streams are used) and high data rates (up to MCS 9). By default aggregation supports AMSDU's and allows up to 128 packets to be aggregated in a frame (namely 64 AMSDUs each containing two packets).

A Linux server connected to this AP via a gigabit switch uses iperf 2.0.5 to generate UDP downlink traffic to the WLAN clients. Iperf inserts a sender-side timestamp into the packet payload and since the various machines are tightly synchronised over a LAN this can be used to estimate the one-way packet delay (the time between when a packet is passed into the socket in the sender and when it is received). Note, however, that in production networks accurate measurement of one-way delay is typically not straightforward as it is difficult to maintain accurate synchronisation between server and client clocks (NTP typically only synchronises clocks to within a few tens of milliseconds).

B. Simulation Measurements

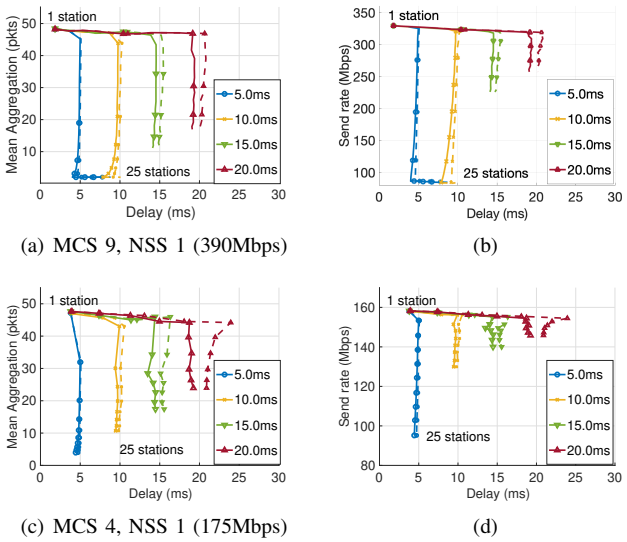
Figure 11 plots measured simulation performance of the controller as the target delay $\bar{T}$ is varied from 5-20ms, the number of client stations is varied from 1 to 25 and for two values of MCS rate. It can be seen that the controller consistently regulates the delay quite tightly around the target value except when the aggregation level hits the specified upper limit of 48 packets, as expected. Also shown on these plots are the 75th percentile values. These mostly overlay the mean values, indicating tight regulation of delay and rate.

C. Experimental Measurements

1) Single Station: Figure 1(b) plots typical a rate and delay time-history measured in our experimental testbed with a single client station. The rate is close to the maximum capacity while the delay is maintained at a low value of around 2ms.

2) Multiple Stations: Figure 12 plots measured time histories with three client stations (two PCs and an android tablet). It can be seen from Figure 12(a) that the rates to the stations quickly converge. Figure 12(b) shows the corresponding delays, which are regulated around the target value of $\bar{T} = 10$ms,

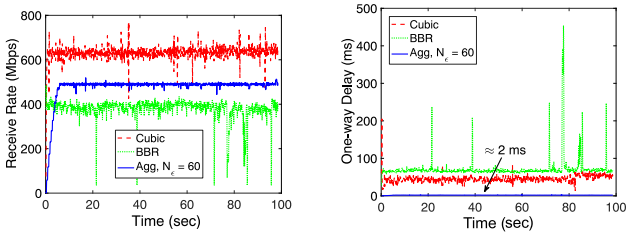
[8]Code can be obtained by contacting the corresponding author.



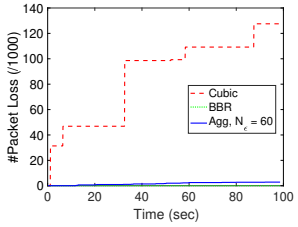
(a) MCS 9, NSS 1 (390Mbps) (b)

(c) MCS 4, NSS 1 (175Mbps) (d)

Fig. 11. Measured delay and rate of the controlled system as the target delay, number of client stations and MCS rate are varied. Solid lines indicate mean delay and rate, dashed lines the 75th percentile values. $\bar{N} = 48$ packets, NS3 data.

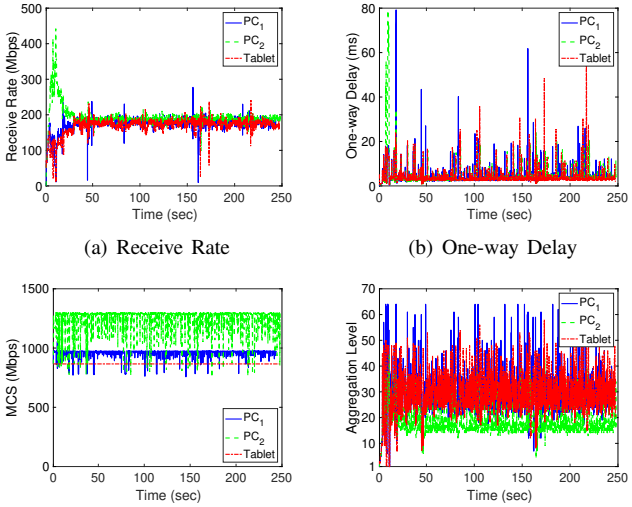


(a) Receive Rate (b) One-way Delay



(c) #Packet loss

Fig. 13. Compare the performance of aggregation-based rate control algorithm with TCP Cubic and BBR. The one-way delay in (b) is averaged over 100ms intervals. Experimental data.



(a) Receive Rate (b) One-way Delay

(c) MCS (d) Aggregation Level

Fig. 12. Managing an edge network using the nonlinear feedback controller. The one-way delay and MCS values are averaged over 100ms intervals. Experimental data.

although fluctuations due to MAC and channel randomness can also be seen. Figure 12(c) shows the measured MCS rates of the three stations, which reflect the radio channel quality (a higher rate indicating a better channel) and it can seen that PC2 has a significantly better channel than the two other clients (it is located closer to the AP). Figure 12(d) shows the measured aggregation levels, and since an equal airtime policy is enforced by the controller it can be seen that the aggregation level of PC2 is lower (since its MCS rate is higher).

3) Performance Comparison With TCP Cubic & BBR: We briefly compare the performance of the aggregation-based rate control algorithm with TCP Cubic [26], the default congestion control algorithm used by Linux and Android. In addition, we compare performance against TCP BBR [18] since this is a state-of-the-art congestion control algorithm currently being developed by Google and which also targets high rate and low latency.

Since TCP Cubic is implemented on Android we use a Samsung Galaxy tablet as client. However, TCP BBR is not currently available for Android and so we use a Linux box (Debian Stretch, 4.9.0-7-amd64 kernel) as the BBR client.

Figure 13 shows typical receive rate and one-way delay time histories measured for the three algorithms. It can be seen from Figure 13(a) that Cubic selects the highest rate (around 600Mbps) but from Figure 13(b) that this comes at the cost of high one-way delay (around 50ms). This is as expected since Cubic uses loss-based congestion control and so increases the send rate until queue overflow (and so a large queue backlog and high queueing delay at the AP) occurs. As confirmation, Figure 13(c) plots the number of packet losses vs time and it can be seen that these increase over time when using Cubic, each step increase corresponding to a queue overflow event followed by backoff of the TCP congestion window.

BBR selects the lowest rate (around 400Mbps) of the three algorithms, but surprisingly also has the highest end-to-end one-way delay (around 75ms). High delay when using BBR has also previously been noted by e.g. [27] where the authors propose that high delay is due to end-host latency within the BBR kernel implementation at both sender and receiver. However, since our focus is not on BBR we do not pursue this further here but note that the BBR Development team at Google is currently developing a new version of BBR v2.

Our low delay aggregation-based approach selects a rate (around 480 Mbps), between that of Cubic and BBR, consistent with the analysis in earlier sections. Importantly, the end-to-end one-way delay is around 2ms i.e. more than 20 times lower than that with Cubic and BBR. It can also be seen from Figure 13(c) that it induces very few losses (a handful out of the around 4M packets sent over the 100s interval shown).

VIII. Conclusions

In this paper we consider the analysis and design of a feedback controller to regulate queueing delay in a next generation edge transport architecture for 802.11ac WLANs. We develop a simplified system model suited to control analysis and design, validated against both simulation and experimental measurements. Using this model we develop a novel nonlinear control design inspired by the solution to an associated proportional fair optimisation problem. The controller compensates for plant nonlinearities and so can be used for the full envelope of operation. The robust stability of the closed-loop system is analysed and the selection of control design parameters discussed. We develop an implementation of the nonlinear control design and use this to present a performance evaluation using both simulations and experimental measurements.

Acknowledgements

DL would like to thank Hamid Hassani for discussions and carrying out the experimental tests in Section VII-C.

References

[1] F. G. Hamid Hassani and D. J. Leith, "Quick and plenty: Achieving low delay and high rate in 802.11ac edge networks," 2019, https://arxiv.org/abs/1806.07761.

[2] 5G White Paper. Next Generation Mobile Networks (NGMN) Alliance, 2015. [Online]. Available: https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf

[3] J. Iyengar and I.Swett, "QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2," IETF Internet Draft, 2015. [Online]. Available: https://tools.ietf.org/html/draft-tsvwg-quic-protocol-00

[4] Y. Gu and R. L. Grossman, "Udt: Udp-based data transfer for high-speed wide area networks," Comput. Netw., vol. 51, no. 7, pp. 1777–1799, May 2007. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2006.11.009

[5] M. Kim, J. Cloud, A. ParandehGheibi, L. Urbina, K. Fouli, D. J. Leith, and M. Medard, "Congestion control for coded transport layers," in Proc IEEE International Conference on Communications (ICC), 2014, pp. 1228–1234.

[6] M. Karzand, D. J. Leith, J. Cloud, and M. Medard, "Design of FEC for Low Delay in 5G," IEEE Journal Selected Areas in Communications (JSAC), vol. 35, no. 8, pp. 1783–1793, 2016.

[7] A. Garcia-Saavedra, M. Karzand, and D. J. Leith, "Low Delay Random Linear Coding and Scheduling Over Multiple Interfaces," IEEE Trans on Mobile Computing, vol. 16, no. 11, pp. 3100–3114, 2017.

[8] Next Generation Protocols – Market Drivers and Key Scenarios. European Telecommunications Standards Institute (ETSI), 2016. [Online]. Available: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp17_Next_Generation_Protocols_v01.pdf

[9] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance enhancing proxies intended to mitigate link-related degradations," Internet Requests for Comments, RFC Editor, RFC 3135, June 2001.

[10] R. Karmakar, S. Chattopadhyay, and S. Chakraborty, "Impact of ieee 802.11n/ac phy/mac high throughput enhancements on transport and application protocols: A survey," IEEE Communications Surveys Tutorials, vol. 19, no. 4, pp. 2050–2091, Fourthquarter 2017.

[11] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, and T. Turletti., "Aggregation with Fragment Retransmission for Very High-Speed WLANs," IEEE/ACM Transactions on Networking, vol. 17, no. 2, pp. 591–604, 2009.

[12] S. Kuppa and G. Dattatreya, "Modeling and Analysis of Frame Aggregation in Unsaturated WLANs with Finite Buffer Stations," in Proc. WCNC, 2006, pp. 967–972.

[13] B. Bellalta and M. Oliver, "A space-time batch-service queueing model for multi-user mimo communication systems," in Proceedings of the 12th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, ser. MSWiM '09. New York, NY, USA: ACM, 2009, pp. 357–364. [Online]. Available: http://doi.acm.org/10.1145/1641804.1641866

[14] B. Kim, H. Hwang, and D. Sung, "Effect of Frame Aggregation on the Throughput Performance of IEEE 802.11n," in Proc WCNC, 2008, pp. 1740–1744.

[15] D. Malone, K. Duffy, and D. Leith, "Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions," IEEE/ACM Trans. Netw., vol. 15, no. 1, pp. 159–172, Feb. 2007. [Online]. Available: http://dx.doi.org/10.1109/TNET.2006.890136

[16] A. Pathak, H. Pucha, Y. Zhang, Y. C. Hu, and Z. M. Mao, "A measurement study of internet delay asymmetry," in Proc 9th Int'l Conf on Passive and Active Network Measurement, ser. PAM'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 182–191.

[17] D. Malone, D. J. Leith, and I. Dangerfield, "Inferring queue state by measuring delay in a wifi network," in Traffic Monitoring and Analysis, M. Papadopouli, P. Owezarski, and A. Pras, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 8–16.

[18] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: Congestion-based congestion control," Commun. ACM, vol. 60, no. 2, pp. 58–66, 2017.

[19] A. Banchs, P. Serrano, and A. Azcorra, "End-to-end delay analysis and admission control in 802.11 DCF wlans," Computer Communications, vol. 29, no. 7, pp. 842–854, 2006. [Online]. Available: https://doi.org/10.1016/j.comcom.2005.08.006

[20] G. Boggia, P. Camarda, L. A. Grieco, and S. Mascolo, "Feedback-based control for providing real-time services with the 802.11e MAC," IEEE/ACM Trans. Netw., vol. 15, no. 2, pp. 323–333, 2007. [Online]. Available: http://doi.acm.org/10.1145/1279660.1279666

[21] A. Garcia-Saavedra, A. Banchs, P. Serrano, and J. Widmer, "Distributed opportunistic scheduling: A control theoretic approach," in Proceedings of the IEEE INFOCOM 2012, Orlando, FL, USA, March 25-30, 2012, 2012, pp. 540–548. [Online]. Available: https://doi.org/10.1109/INFCOM.2012.6195795

[22] P. Serrano, P. Patras, A. Mannocci, V. Mancuso, and A. Banchs, "Control theoretic optimization of 802.11 wlans: Implementation and experimental evaluation," Computer Networks, vol. 57, no. 1, pp. 258–272, 2013. [Online]. Available: https://doi.org/10.1016/j.comnet.2012.09.010

[23] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," Journal of the Operational Research Society, vol. 49, no. 3, pp. 237–252, Mar 1998. [Online]. Available: https://doi.org/10.1057/palgrave.jors.2600523

[24] C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: motivation, architecture, algorithms, and performance," in Proceedings IEEE INFOCOM 2004, The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, Hong Kong, China, March 7-11, 2004, 2004, pp. 2490–2501. [Online]. Available: https://doi.org/10.1109/INFCOM.2004.1354670

[25] D. Leith and W. Leithead, "Survey of gain-scheduling analysis and design," Int. J.Control, vol. 73, no. 11, pp. 1001–1025, 2000.

[26] S. Ha, I. Rhee, and L. Xu, "Cubic: A new tcp-friendly high-speed tcp variant," SIGOPS Oper. Syst. Rev., vol. 42, no. 5, pp. 64–74, 2008.

[27] Y. Im, P. Rahimzadeh, B. Shouse, S. Park, C. Joe-Wong, K. Lee, and S. Ha, "I sent it: Where does slow data go to wait?" in Proc Fourteenth EuroSys Conf, ser. EuroSys '19. New York, NY, USA: ACM, 2019, pp. 22:1–22:15.

PLACE PHOTO HERE

Francesco Gringoli received the Laurea degree in telecommunications engineering from the University of Padua, Italy, in 1998 and the PhD degree in information engineering from the University of Brescia, Italy, in 2002. Since 2018 he is Associate Professor of Telecommunications at the Dept. of Information Engineering at the University of Brescia, Italy. His research interests include security assessment, performance evaluation and medium access control in Wireless LANs. He is a senior member of the IEEE.

PLACE
PHOTO
HERE

Doug Leith graduated from the University of Glasgow in 1986 and was awarded his PhD, also from the University of Glasgow, in 1989. In 2001, Prof. Leith moved to the National University of Ireland, Maynooth and then in Dec 2014 to Trinity College Dublin to take up the Chair of Computer Systems in the School of Computer Science and Statistics. His current research interests include wireless networks, network congestion control, distributed optimization and data privacy.