TCP Fairness in 802.11e WLANs

D. J. Leith, P. Clifford, D. Malone, and A. Ng

***Abstract*— We investigate the use of the 802.11e MAC EDCF to address transport layer unfairness in WLANs. A simple solution is developed that uses the 802.11e $AIFS$, $TXOP$ and CW_{min} parameters to ensure fairness between competing TCP uploads and downloads.**

***Index Terms*—**

I. Introduction

EXISTING work on 802.11e tuning algorithms is largely informed by the quality of service 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. Although lacking the time critical aspect of voice traffic, there is a real requirement for efficient and reasonably fair sharing of the wireless capacity between competing data flows. Unfortunately, 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. While the literature relating to WLAN fairness at the MAC layer is extensive, this issue of transport layer TCP fairness has received far less attention. Early work by Balakrishnan and Padmanabhan [1] studies the impact of path asymmetries in both wired and wireless networks, while more recently Detti et al.[2] and Pilosof et al.[3] have specifically considered TCP unfairness issues in 802.11 infrastructure WLANs and Wu et al. [4] study TCP in the context of single-hop 802.11 ad hoc WLAN's. With the exception of [4], 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. However, as we shall see, the roots of the problem lie in the MAC layer enforcement of per station fairness. Hence, it seems most natural to seek to resolve this issue at the MAC layer itself. In this paper we investigate how we might use the flexibility provided by the new 802.11e MAC to resolve the transport layer unfairness in infrastructure WLANs. The paper considers TCP uploads and downloads, and mixtures of both.

II. Experimental Setup

Recently, hardware supporting a useful subset of the 802.11e functionality has become available and so in this paper we investigate the behaviour of TCP traffic in a realistic network rather than via simulations. Our WLAN consists of a desktop PC acting as an access point (AP), and 12 PC-based embedded Linux boxes based on the Soekris net4801 [5] acting as client stations. All systems are equipped with an Atheros 802.11a/b/g PCI card with an external antenna. The system hardware configuration is summarised in Table I. All nodes, including the AP, use a Linux 2.6.8.1 kernel and a version of the MADWiFi [6] wireless driver modified to allow us to adjust the 802.11e CW_{min}, $AIFS$ and $TXOP$ parameters. Specific vendor features on the wireless card, such as turbo mode, are disabled. All of the tests are performed using the 802.11b physical maximal transmission rate of 11Mbit/sec with RTS/CTS disabled. The configuration of the various network buffers is also detailed in Table I. In particular, we have increased the size of the TCP buffers to ensure that we see true AIMD behaviour (with small TCP buffers TCP congestion control is effectively disabled as the TCP congestion window is determined by the buffer size rather than the network capacity). We have also carried out tests investigating the impact of the size of interface and driver queues and obtain similar results for a range of settings.

TABLE I
EXPERIMENTAL SETUP

Hardware		
1× AP	Dell GX 280	2.8Ghz P4
12× node	Soekris net4801	266Mhz 586
WLAN	D-Link DWL-G520	Atheros AR5212
Buffers	default	used
TCP	64KB	1MB
interface tx	199 packets	10 packets
driver tx	200 packets	10 packets

III. TCP Unfairness Over 802.11 WLANs

We consider, in turn, unfairness between competing TCP upload flows and between competing upload and download flows in 802.11 WLAN's.

A. Unfairness between competing TCP upload flows

Fig. 1 illustrates the behaviour of competing TCP upload flows over an 802.11b WLAN. Gross unfairness between the throughput achieved by competing flows is evident. The source of this highly undesirable behaviour 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).

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. TCP's operation implicitly assumes that the forward (data) and reverse (ACK)

Manuscript received April 25, 2005. The associate editor coordinating the review of this letter and approving it for publication was Dr. Aleksandra Smiljanic.
The authors are with the Hamilton Institute, NUI Maynooth, Ireland (e-mail: peter.clifford@nuim.ie).
Digital Object Identifier 10.1109/LCOMM.2005.xxxxx.

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 [2]. 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 node, 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 action 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 [1]. Asymmetry in the forward and reverse path packet transmission rate that leads to significant queueing and dropping of TCP ACKs can disrupt the TCP ACK clocking mechanism, hinder congestion window growth and induce repeated timeouts. With regard to the latter, 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, when 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.

B. Unfairness between TCP upload and download flows

Asymmetry also exists between competing upload and download TCP flows that can create unfairness. This is illustrated for example in Fig. 1 where it can be seen that upload flows achieve significantly greater throughput than competing download flows. Suppose we have n_u upload flows and n_d download flows. Since download flows must all be transmitted via the AP, 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.

IV. Restoring Fairness

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.

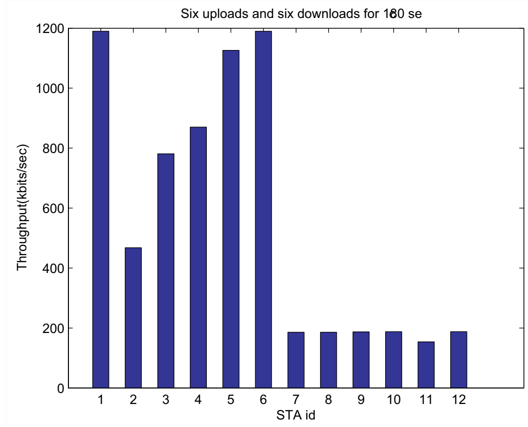

Fig. 1. Performance of six TCP uploads and six TCP downloads with default 802.11b parameters. Stations 1–6 are uploads and 7–12 are downloads.

TABLE II

TCP 802.11e MAC parameters

		$AIFS$	CW_{min}	TX_{op} (packets)
AP	Upload ACKS	DIFS	4	1
	Download data	4 slots + DIFS	32	n_d
wireless station	Download ACKS	DIFS	32	1
	Upload data	4 slots + DIFS	32	1

To address TCP's performance problems, two issues must be addressed; namely, asymmetry between the TCP data and TCP ACK paths that disrupts the TCP congestion control mechanism, and network level asymmetry between TCP upload and download flows.

Symmetry can be restored between the TCP data and TCP ACK paths by configuring the MAC such that TCP ACKs effectively have unrestricted access to the wireless medium. Recall that in 802.11e the MAC parameter settings are made on a per class basis. Hence, we collect TCP ACKs into a single class (i.e. queue them together in a separate queue) and confine prioritisation to this class[1]. The rationale for this approach 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.

In the case of competing TCP upload and download flows, recall that the primary source of unfairness arises from the

[1] In our tests packet classification is carried out based on packet size.

fact 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.

We therefore consider using the 802.11e MAC parameters detailed in Table II. Here, the 802.11e $AIFS$ and CW_{min} parameters are used to prioritise TCP ACKs. A small value of $AIFS$ and CW_{min} yields near strict prioritisation of TCP ACKs at the AP. A larger value of CW_{min} is used at the wireless stations in order to reduce contention between competing TCP ACKs. The $TXOP$ packet bursting mechanism in 802.11e provides a straightforward and fine grained mechanism for prioritising TCP download data packets. By transmitting n_d packets (one packet to each of the n_d download destination stations) 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. Note that the number n_d of distinct destination stations can be readily determined by inspection of the AP interface queue in real-time, with no requirement for monitoring of the wireless medium activity itself. The effect is to dynamically track the number of active TCP download stations and always ensure the appropriate prioritisation of TCP download traffic. Hence, this approach accommodates both bursty, short-lived traffic such as HTTP and long-lived traffic such as FTP in a straightforward and consistent manner (see later for examples).

Revisiting the example in Fig. 1, the impact of the proposed prioritisation approach can be seen in Fig. 2. 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. Although space restrictions prevent us from including the additional results, we have measured similar levels of fairness across a range of network conditions, including varying numbers of upload and download stations and situations where the number of uploads is not the same as the number of downloads, confirming the effectiveness of the proposed solution.

The performance of the proposed approach with short-lived TCP flows is illustrated in Fig. 3. Here we model a client-server application where each user opens TCP upload flows (client "requests") and, in response, corresponding downloads are initiated. Since, as observed previously, lockout of TCP flows is common in 802.11b WLANs we model user impatience by restarting a client-server session if it fails to complete within a period of 10 seconds. The average time to completion of a client-server session is plotted in Fig. 3 versus the number of wireless stations. As we would expect the MAC load to increase linearly with the number of users, we normalise by dividing by the number of users. It can be seen that in 802.11b the normalised completion time remains constant until about 15 users and then increases rapidly. In contrast using the 802.11e approach the normalised completion time remains small until we reach about 40 users, indicating more than a doubling in useful capacity for the same physical channel rate. Note that we have presented NS packet-level simulation results here rather than results from our experimental network as the network lacks sufficient nodes to explore the performance boundary with short-lived flows.

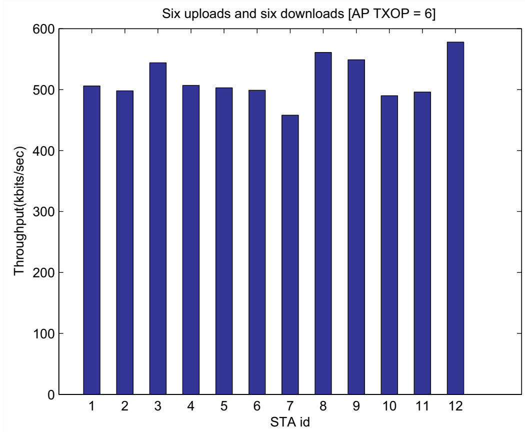

Fig. 2. Performance of six TCP uploads and six TCP downloads with suggested 802.11e parameters. Stations 1–6 are uploads and 7–12 are downloads.

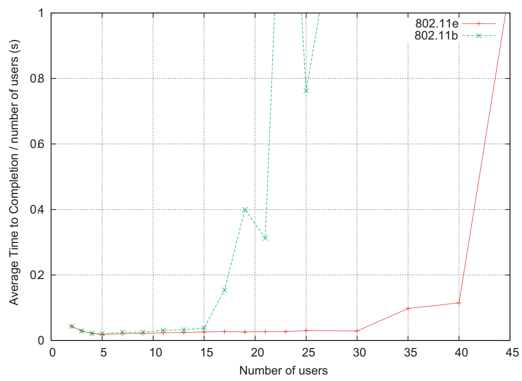

Fig. 3. Example of client-server performance in 802.11b and prioritised 802.11e WLANs.

V. Conclusions

In this paper we investigate how we might use the flexibility provided by the new 802.11e MAC to resolve the transport layer unfairness in WLANs. A simple solution is developed that uses the 802.11e $AIFS$, $TXOP$ and CW_{min} parameters to ensure fairness between competing TCP uploads and downloads. The effectiveness of the proposed solution is demonstrated in an experimental wireless network testbed as well as via packet-level simulation tests.

VI. Acknowledgements

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

References

[1] H. Balakrishnan and V. Padmanabhan, "How network asymmetry affects TCP," *IEEE Commun. Mag.*, vol. 39, pp. 60-67, Apr. 2001.

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

[3] S. Pilosof, R. Ramjee, Y. Shavitt, and P. Sinha, "Understanding TCP fairness over wireless LAN," in *Proc. INFOCOM 2003*, April 2003.

[4] 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," *Proc. INFOCOM 2002*, June 2002.

[5] Soekris Engineering, `http://www.soekris.com/`.

[6] Multiband Atheros Driver for WiFi(MADWiFi), `http://sourceforge.net/projects/madwifi/`, Oct. 2004 version.