Experimental Evaluation of FAST TCP Performance and Fairness in DOCSIS Cable Modem Networks

Irena Atov and David Kennedy
Centre for Advanced Internet Architectures. Technical Report 060523A
Swinburne University of Technology
PO Box 218, Vic. 3122, Australia
Email: {iatov,dkennedy}@swin.edu.au

***Abstract*— In this paper we present experimental results evaluating the performance and fairness of FAST TCP in a series of tests involving realistic low rate network access scenarios. Links both using the DOCSIS cable modem medium access control (MAC) cable modem and simple low rate links were investigated. We seek to compare our expectations from theory with the behavior of an actual access network implementation.**

***Index Terms*— TCP Congestion control, Evaluation of FAST TCP, Low-speed networks.**

I. Introduction

Motivated by the phenomenal growth of the Internet in the recent years, a number of ISPs are actively deploying various broadband access technologies, such as xDSL modems, cable modems and 802.11 wireless LANs, to offer high-speed data services to residential as well as mobile subscribers. One of the primary ways of characterising performance of the broadband data system as perceived by subscribers is in terms of throughput observed by applications operating above the Transmission Control Protocol (TCP) layer. From the networking perspective, the achieved throughput depends not only on the bandwidths available on the downstream and upstream channels but, it has become increasingly evident in the recent years, that the specific TCP implementations used at the communicating nodes greatly influence the achievable throughput. The current standard version of TCP (RFC 793, sometimes known as "TCP Reno") was designed two decades ago, reflecting the best understanding of network dynamics and congestion control at the time, and today it is increasingly becoming a limiting factor in network performance.

As a result, there have been many TCP (Transmission Control Protocol) proposals aiming to improve the current standard version of TCP (e.g., [5], [6], [9], [10]). One such popular proposal that has received significant attention in recent years is FAST (Fast AQM Scalable TCP) [7], [8], which has been designed at Caltech (California Institute of Technology) to improve performance in high speed networks, especially those with long propagation delays. Unlike proposals such as BIC [5], Scalable TCP (STCP) [6] and High-Speed TCP (HSTCP) [9], which follow TCP Reno's model of reacting to packet loss as a congestion indicator to drive their flow control decisions, FAST TCP follows the approach of TCP Vegas [10] and responds to queueing delay. This allows the equilibrium queue size to be orders of magnitude smaller than the buffer size, and avoids the waste incurred by packet losses.

IETF standardization and worldwide deployment requires that any new TCP variant must be tested and validated experimentally in real-world trials and in a wide variety of network environments. It is also crucial that independent groups repeat these tests. To date, FAST has been tested by Caltech and independent groups such as SLAC (Stanford Linear Accelerator Center) and CERN (The European Particle Physics Laboratory) in a wide range of high speed environments. Therefore, it is becoming increasingly important to experimentally evaluate the performance of TCP FAST in low rate (1-10 Mbps) access networks typical in the existing Internet.

Data Over Cable System Interface Specification (DOCSIS)[16] has emerged as a single standard for data communications over hybrid fiber/coax (HFC) cable networks and DOCSIS-enabled cable modems are currently the most widely deployed broadband access IP technology. Moreover, the cable modem access system is of particular interest to study as it is a shared medium i.e., it has a MAC protocol, and it is reasonable to expect that the delay this introduces could potentially interact with the delay-based control of FAST. A typical DOCSIS cable network consists of two key components: the Cable Modems (CM) located at the customer premises, and a Cable Modem Termination System (CMTS) located in

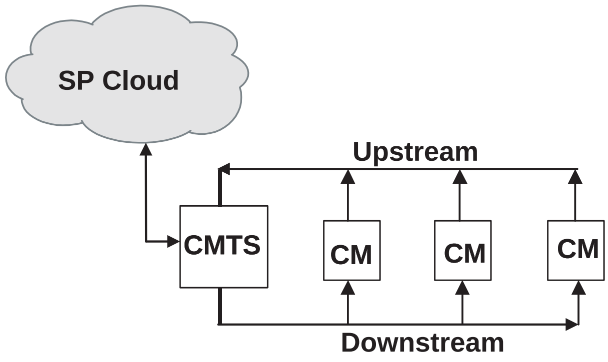

Fig. 1. Logical topology of DOCSIS network

the service provider's (SP) network. Transmission over the downstream and upstream channels is controlled by the CMTS. The upstream channel is a multipoint-to-point channel shared by all the cable modems (CM) using a time-slot structure. A centralized MAC protocol based on a reservation scheme, also known as a *Request-and-Grant cycle*, controls the access to the upstream channel which is shared by all CMs using a TDMA system (i.e., CMs request time to transmit and CMTS allocates time based on availability). Logical topology of DOCSIS network is illustrated in Fig. 1.

There has been little research exploring the impact that the DOCSIS MAC and physical layers has on the performance of TCP. Existing studies, e.g., [12], [13], [14] and the references therein, are confined to developing a model of DOCSIS using a simulator. Moreover, these studies have mainly focused on analyzing the standard TCP Reno protocol and other TCP variants, including FAST have not, to the best of our knowledge, been considered. We have constructed a testbed network using a CISCO DOCSIS 1.1 cable system [17] which allows us to investigate cable modem network operation in a real testing environment. Within the testbed we can simulate a variety of typical ISP scenarios, which allows us to explore interactions between TCP flow control behavior, various ISP-configured DOCSIS-based parameter settings (e.g., upstream (US) and downstream (DS) bandwidth limits) and end-user perceptions of overall system performance. In previous work [1], we provided an insight into the interaction between the DOCSIS MAC protocol and FAST TCP application performance where we experimentally characterized the impact of downstream and upstream bandwidth limits on the overall FAST TCP performance through the DOCSIS cable system. In order to get a better understanding of how the DOCSIS system interacts with the FAST flow control mechanism it was important to first consider a static environment i.e., a single FAST connection in the system. In this environment, [1] provided many important findings and observations which further stimulated our work. In this paper, we build on the work in [1] by performing extended analysis and experimentation and evaluating the performance and fairness properties of FAST in more realistic multi-flow *dynamic* access network scenarios.

The rest of the paper is organized as follows. Section II provides background and outlines our objectives. Section III describes the setup of our testbed. The results are presented and analysed in Sections IV and V, respectively. Finally, Section VI provides concluding remarks and discusses directions for future work.

II. Background and Objectives

TCP regulates a sources transmission rate by adapting its window size according to some congestion signal from the network. Most congestion control algorithms follow TCP Reno in adjusting a sources transmission rate based on the rate at which packets are lost, thus interpreting packet loss as an indication of congestion (i.e., use packet loss rate as an indication of congestion).

FAST follows from TCP Vegas [10] in adjusting flow rates in response to the measured delay. These *delay-based* algorithms adjust a source's window size w to attempt to maintain a constant number of its own packets, α, queued in nodes along its path. The queueing delay is estimated as the difference between the mean round trip time, denoted D, and the minimum round trip time observed by any packet, d.

FAST updates the window size according to [7], [8]

$$w(t+1) = \left\lfloor \frac{1}{2}\left(w(t) + \frac{d}{D}w(t) + \alpha\right)\right\rfloor . \qquad (1)$$

The alpha parameter is the main control parameter, which determines the equilibrium bandwidth share for a flow and the aggressiveness during the additive increase phase when queueing delay is zero. So, the performance of FAST is fully dependent on how this parameter is set, however, the optimal value of alpha is difficult to set in practice.

In [1] we focused on the problem of tuning alpha in two different low-speed environments involving the DOCSIS cable modem and simple low rate links, respectively. This problem has been addressed for the high-speed regime only, for which a simple rule of thumb (for tuning alpha) was proposed. For high speed links, it has been recommended that α be set to cause a given small queueing delay (that is just large enough to be reliably measured or detected), such as 2 ms [20]. In order to cause queueing delay of 2 ms, the rule of thumb is to set alpha to 2C where C is the capacity of the bottleneck link in packets/msec. The results in [1] showed that this rule of thumb does not work in low-speed environment, it gives insufficient queueing (and consequently FAST cannot achieve its maximum throughput), especially when DOCSIS links are used. The study presented in [1] only considered static scenarios where the bottleneck link in the access network carried either one or two FAST flows. In this paper, we build on this work and extend the analysis to network scenarios involving multiple flows of FAST interacting over a single bottleneck link in a variety of access network scenarios involving a single, as well as multiple CMs. Specifically, important points for investigation that we consider are:

- Maximum achievable utilisation of the system as a function of the number of flows and, most importantly, what setting of the main control parameter alpha for the individual FAST flows would be required to achieve that maximum.
- Investigation of the fairness properties of the FAST protocol in a dynamic environment where flows join in a random fashion.
- Investigation of a suitable parameter tunings to overcome difficulties or optimise FAST TCP for the above scenarios.

III. DOCSIS Testbed Configuration

We have experimentally evaluated the performance of FAST over two different access networks, each with a single bottleneck link. One contained a DOCSIS cable modem, and the other was a simple rate-limited link. We considered two different sets of experiments for the analysis of performance and fairness of FAST TCP on a testbed designed to simulate a typical customer attachment to ISP offering content from local servers on the ISP's network.

Figure 2 shows the testbed used for our experiments. The testbed uses real world DOCSIS equipment and is integrated with the existing Broadband Access Research Testbed at CAIA [7]. The implementation of the FAST TCP testbed employed in our study is described in detail in [2].

The testbed consists of four end hosts: one sender (TCP server), which run Linux with Caltech's FAST patches and three receivers, which run standard Linux. In addition to the DOCSIS cable network - comprised of a system of up to three cable modems (CMs) and

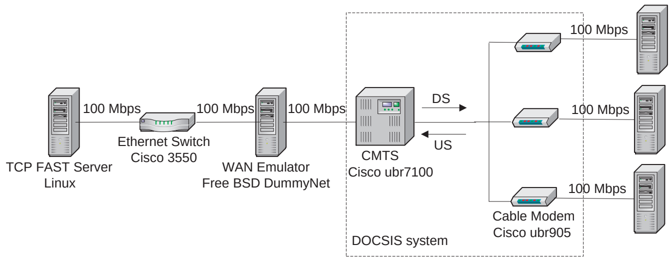

Fig. 2. Test setup

a cable modem termination system (CMTS) - a bridge running Dummynet [9] under FreeBSD and a standard Ethernet switch are used to emulate a typical ISP network. The sender, the receiver and the dummynet router are 2.4 GHz Intel Celerons with 256 MB of RAM and 100 Mbps Ethernet cards. The switch is a Catalyst 3550, the CMTS is a Cisco ubr7100 and the CM is a Cisco ubr905, which also acts as a router.

All the links in the network except for the bottleneck link have capacity 100 Mbps. The bottleneck DOCSIS link was configured with various bandwidths in the downstream (DS) and upstream (US) channels (in the range of 0.5-3Mbps) through adequate configuration of the CMTS, which governs transmission in the DOCSIS network. The buffering on the bottleneck link was 1024 ms, the maximum value of the Cisco CMTS [22] configuration (details of the CMTS channel capacity and buffer configuration are provided in [2]). The Dummynet was configured to emulate a high-speed Wide Area Network (WAN) path of 100 ms Round Trip Time (RTT) without imposing any limitation on the downstream (DS) and upstream (US) channel capacities. Additional constant delays, notably in the DOCSIS link, make the total RTT approximately 115 ms when no traffic is present. The dummynet used a buffer size of 2048 kbytes (involving two pipes in series, each of 1024 kbytes) in order to ensure that no packet loss occurred in the core network ([2]). The experiments consisted of running multiple TCP flows with 1500-byte packets on the downlink (The DOCSIS link also transmitted low-rate keep-alive messages). To generate network traffic and to measure throughput we used iperf [24] software tool and tcpdump [25] was used as a general network monitoring program.

For the experiments involving a simple low-speed link, the DOCSIS system was bypassed. Instead the same Dummynet that emulated the WAN delay was also configured to emulate the bottleneck capacity limits in both the DS and US, and the limited buffering on the bottleneck link. The dummynet RTT was still set to 100 ms.

IV. Multiple Flow Results

One of the main findings reported on in [1] was that the cable modem system introduces consistent additional delays when the link is highly, but not fully, utilised, and that these delays result in the need for a congestion window larger than the bandwidth-delay product. This in turn, requires that the main FAST TCP control parameter alpha (i.e., the target queue size) be set large enough to allow for the additional packets stored in the cable modem link. As a result of these delays, the throughput achieved by a FAST flow in a DOCSIS access system is much less than in an equal rate simple link. Other important observation was that the required α value *does not scale inversely* with n where n is the number of flows on the bottleneck link. Namely, from theory we expect when n FAST flows share a single bottleneck link, the total queueing at the link to be $n\alpha$. Thus, if the only reason to need $\alpha > 1$ were to ensure that the queueing delay was larger than the timing uncertainties, as is the case in high speed networks, we would expect the required α value to scale inversely with n. In other words, one wouldn't expect the total target queueing delay to change with the increase of the number of TCP flows.

The results in [1] showed that this expectation from theory does not hold in DOCSIS-based shared medium environment. For ease of reference we include the results from [1] for the total throughput obtained by one and two FAST flows as a function of α for both a simple link with downlink/uplink speed of 3 Mbps/512 kbps and a DOCSIS link of the same speed, shown in Fig. 3 and Fig. 4, respectively. It can be seen that a single flow needs $\alpha = 13$ or target queueing delay of 52 ms to achieve full utilisation on a 3 Mbps DOCSIS link. Contrary to our expectation, that when two flows are sharing the link each individual flow needs $\alpha = 7$ (which would again give a total queueing delay of 52 ms), the required α rather than decreasing by a factor of two, actually, increased to 22. That corresponds to a total target queue size of 44 packets or a delay of 176 ms. This trend of superlinear buffer requirements is concerning, in light of the fact that the cable modems had a default "traffic shaping" buffer with maximum delay 512 ms, which can be increased to at most 1024 ms [22]. Hence, the first point of investigation in this paper is to see if this trend of *increasing alpha* on DOCSIS link is continuing with the increase of the number of flows. Subsequently,

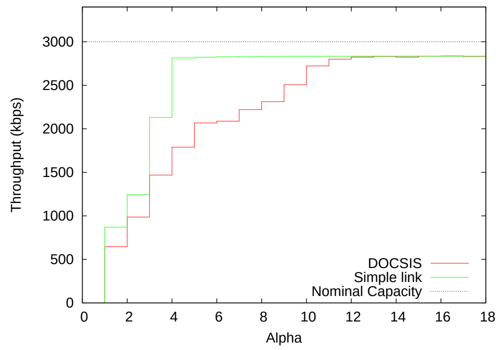

Fig. 3. Throughput vs. alpha for a single FAST flow for DS=3Mbps, US=512Kbps, DOCSIS and simple link [1].

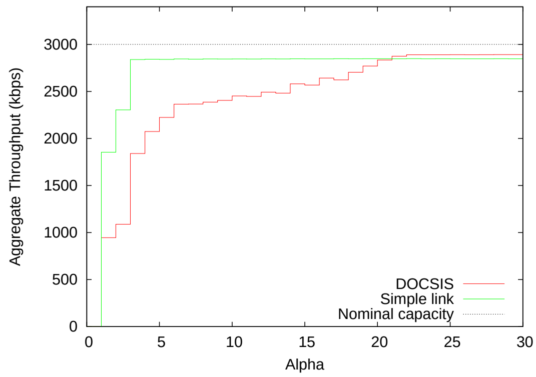

Fig. 4. Throughput versus α for two FAST flows for DS=3 Mbps, US=512 kbps, for simple and DOCSIS links [1].

and inline with the analysis conducted in [1], our aim is to investigate the cause for needing an increased alpha in the DOCSIS system.

A. Single Cable Modem System

We now extend the above mentioned study to consider multiple FAST flows sharing a single bottleneck link in DOCSIS cable modem system. Specifically, we considered a system of four, six, eight and ten TCP connections, respectively, which were generated and run as concurrent iperf sessions from the TCP server to the receiver. For this set of experiments, all flows shared the same cable modem (and receiver). The α parameter was set equal for all TCP connections and varied from 1 to 30. Each experiment was run 10 times for statistical accuracy, resulting in a total of 1200 tests for each considered type of access network (simple low-speed links were also considered).

Figure 5 shows the aggregate throughput obtained by four FAST flows as a function of α for both a simple link with downlink/uplink speed of 3 Mbps/512 Kbps and a DOCSIS link of the same speed. This demonstrates

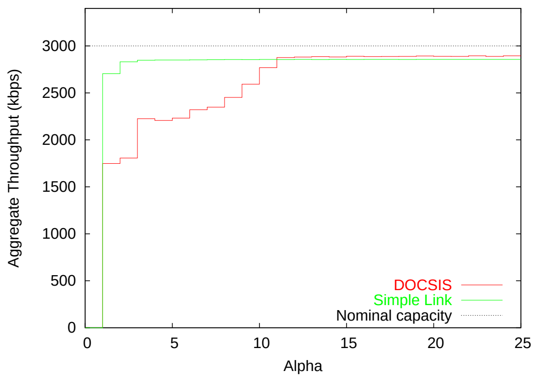

Fig. 5. Throughput versus α for four FAST flows for DS=3 Mbps, US=512 kbps, for simple and DOCSIS links.

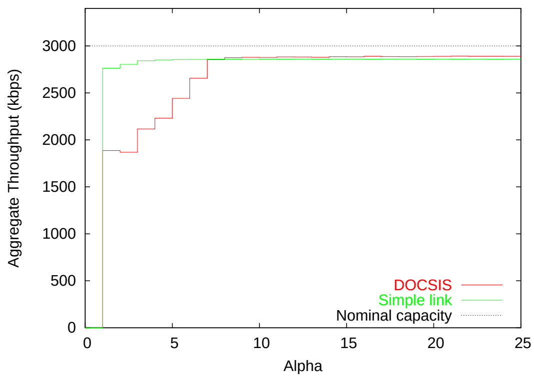

Fig. 7. Throughput versus α for six FAST flows for DS=3 Mbps, US=512 kbps, for simple and DOCSIS links.

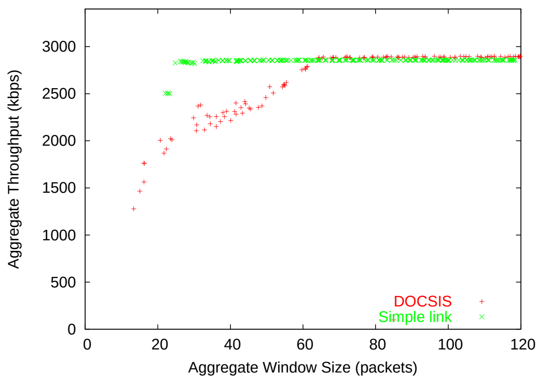

Fig. 6. Throughput versus window size for four FAST flows for DS=3Mbps, US=512Kbps, for simple and DOCSIS links.

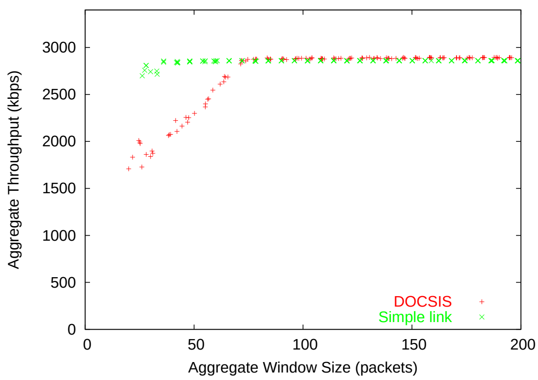

Fig. 8. Throughput versus window size for six FAST flows for DS=3Mbps, US=512Kbps, for simple and DOCSIS links.

that $\alpha = 3$ is sufficient for full utilisation on a simple 3 Mbps link, but that a much larger value, $\alpha = 12$ is required on a 3 Mbps DOCSIS link[1]. Consistent with the results from [1], the throughput achieved by a FAST flow for a given α is much less in a DOCSIS access system than an equal-rate system not running DOCSIS. At 3 Mbps with 1500-byte packets, $\alpha = 3$ corresponds to a delay of 12 ms and $\alpha = 12$ corresponds to a delay of 48 ms, which in both instances is much larger queueing delay than what is necessary to obtain accurate timing estimates. The reason for needing such a large queueing is discussed in the following, starting with the simple link case.

If the reason for needing $\alpha = 4$ in the single flow case (Fig. 3) were simply to allow for delay jitter, we would expect in the case of four flows to achieve full bandwidth utilisation with $\alpha = 1$ for each flow, as opposed to the required $\alpha = 3$. This is not too unexpected, however, since we showed in [1] that it is reasonable to need $\alpha \geq 3$ for reasons other than needing queueing delay. Specifically, we showed that there are two major effects at work causing low utilization for $\alpha <= 3$. The first of these is caused by the *burstiness due to delayed acknowledgements*, while the second is caused by the *integer arithmetic* of the rule that FAST uses to update its window (Eq. II). The delayed ACK mechanism causes two back-to-back packets to be transmitted at once, which even at low utilization of the link, results in the mean queueing delay to be overestimated by an entire packet time. To account for this effect, FAST needs at least $\alpha = 1$ to achieve full utilisation. The second effect, caused by the integer arithmetic, can require that α be increased by at most two (for details see [1]). Note, however, that the impact of the integer arithmetic depends on the amount of rounding at the particular equilibrium point, which means that combined the two effects one can observe instances where full utilisation

[1]The slight reduction from full capacity is mostly accounted for by the 2.5% overhead of TCP and IP headers (20+20 bytes out of 1500).

is achieved with $\alpha = 1$ (when the amount of rounding from the integer arithmetic operation is zero). Equally, the requirement of $\alpha > 3$ in some instances may be due to some additional burstiness, other than the unavoidable burstiness from the delayed ACKs. In our case, the experimental results for the simple link case match well the expectations from theory.

For the discrepancy in the DOCSIS case, (i.e., at 3 Mbps $\alpha = 12$ is needed for full utilisation, compared to $\alpha = 3$ on a simple link) there could be several possible reasons, as discussed in [1]. First possibility is that the random delays introduced by the MAC protocol of the DOCSIS system interfere with FASTs estimate of the queueing in the network, resulting in the congestion window being too low. A second possibility is that the actual window size required to achieve a full utilization in a DOCSIS system is larger than the bandwidth-delay product as a result of the additional queueing that the MAC protocol introduces.

To investigate the second possibility, the throughput achieved is plotted against the aggregate window size in Figure 6 for the 300 experiments conducted with four FAST flows using α values from 1 to 30. From theory we know that a bottleneck link carrying a number of flows in a purely deterministic network will be fully utilized if the flowss aggregate window size is at least the "bandwidth delay product", d times the link capacity. For a 100 ms (or 115 ms) path with a bottleneck link of 3 Mbps, this is 25 (or 28) packets of 1500 bytes. For smaller windows, the throughput reduces in proportion to the window size. From Fig. 6 it can be seen that the expected behavior is observed for a simple link (i.e., full utilization is achieved for a total window size equal to the bandwidth delay product of 25 packets). However, the DOCSIS system consistently yields lower utilization than predicted and the total windows size required for full utilisation is significantly greater than the bandwidth delay product (28 packets) i.e., full utilisation is achieved for a total window size of 68 packets. Thus, even if FAST correctly sets the window size to the bandwidth delay product plus α, full utilisation will not be achieved unless $\alpha \geq (6828)/4 = 10$ packets. As discussed previously, the integer arithmetic of the rule which FAST uses to update its window can require α to be increased by 2, yielding a requirement of $\alpha = 10 + 2 = 12$ packets for full utilization. This is precisely what we observed in Fig. 5, suggesting that FAST's ability to estimate the queueing in the network is not affected by the delay fluctuations introduced by DOCSIS. This indicates, however, that DOCSIS is not work conserving i.e., it may buffer packets even when the link is idle (due to the Request-and-Grant scheme it employs) and, therefore, α needs to be set large enough to allow for the additional packets stored in the cable modem link. Due to this additional queueing delay on the cable modem link, occurring before the link is fully utilised, the total target queueing delay increases with the increase of the number of flows i.e., total queueing delay of 192 ms for four flows, as opposed to 176 ms for two flows and 52 ms for the single flow. Thus, contrary to the expectation from theory, the total queueing delay does change (constantly increases) and the target queue size α does not scale inversely with the increase of the number of flows.

TABLE I
TARGET ALPHA AS A FUNCTION OF NUMBER OF FLOWS IN SINGLE CM SYSTEM

Number of flows - N	*Total alpha required - α_T*	*Total queueing delay (msec)*	*Per-flow alpha α*
1	13	52	13
2	44	176	22
4	48	192	12
6	54	216	9
8	72	288	9
10	80	320	8

Let us now consider the case of six flows sharing a single bottleneck link. The results for this set of experiments are summarised in Fig. 7 and Fig. 8, which show the aggregate throughput as a function of α and the aggregate window, respectively, for both a simple link with downlink/uplink speed of 3 Mbps/512 Kbps and a DOCSIS link of the same speed. It can be seen that FAST requires $\alpha = 9$ to obtain full utilisation using DOCSIS, compared with $\alpha = 3$ on a simple link. As previously observed for the case of two and four flows, we can see that the α value does not scale inversely with the number of flows, and that the total queuing delay has increased further with the presence of more flows to 216 ms. This is not as big a jump as from one to two flows, but we can still see the undesired effect of an increase in buffering requirements with more flows. By analyzing the results from Fig. 8, it can be seen that full utilisation on a simple link is achieved for a total congestion window equal to the bandwidth delay product of 25 packets, whereas on a DOCSIS link much larger congestion window is required i.e., 70 packets. Again, assuming that FAST correctly sets the window size to the bandwidth delay product plus α, indicates that full utilistaion can only be achieved if the target queue size for each flow is at least $\alpha = (7028)/6 = 7$. When the effect from the integer arithmetic is accounted for, the target α increases to 9 and this precisely matches the obtained results (Fig. 7). We extended this analysis to the case of 8, and 10 flows, summarised in Table I, and the results consistently confirm the previous observation that FAST window size is not adversely affected by the randomness of the delay at this operating point. The continuous increase in total target queueing delay and non-inverse scale of alpha with the number of flows is attributed to the additional delays and packet buffering on the cable modem link, occurring due to the non-work-conserving nature of the Request-and-Grant Cycle mechanism that DOCSIS employs for controlling data transmission on the DS and US channels.

B. Multiple Cable Modem System

We have extended the experiments further to consider multiple FAST flows sharing a single bottleneck link (DS=3 Mbps and US=512 Kbps) in a DOCSIS system, comprised of two and three cable modems, respectively. The experiments consisted of running multiple iperf sessions from the TCP server to all receivers concurrently, starting with a single flow per cable modem and gradually incrementing the number of flows per cable modem to 2, 3, 4, 5 and 10. All flows had the same RTT of 100 ms and started and terminated at the same times. The α parameter was set equal for all TCP flows and varied from 1 to 20. Each experiment was run 10 times for statistical accuracy.

We plot the aggregate throughput versus alpha for this test suite and the results for 2 CMs and 3 CMs are summarised in Figure 9 - Figure 10 and Table II and Table III, respectively. When including multiple modems in the access network, the same property of increasing target queueing requirements for more flows continues to show. Interestingly enough, however, the required value of alpha for maximum throughput is actually lower than what was required in single cable modem system and with the increase in the number of flows tends to asymptote towards the "simple-link" target alpha requirement of $\alpha = 3$.

.

V. Fairness Analysis

As discussed in Section II, FAST regulates a source's transmission rate by adapting its window size in response to the measured queueing delay. The queueing delay is estimated as the difference between the mean RTT D, and the round-trip propagation delay, which is in turn estimated as the minimum RTT observed by any packet d (also called *baseRTT*). Inherent problem with delay-based congestion control algorithms (this also affects TCP Vegas) is that, if the actual round-trip propagation

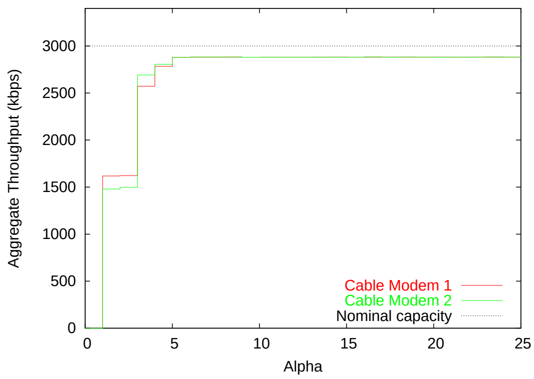

Fig. 9. Throughput versus α for 2 FAST flows in 2 CMs (1 flow per CM) for DS=3 Mbps, US=512 kbps DOCSIS link.

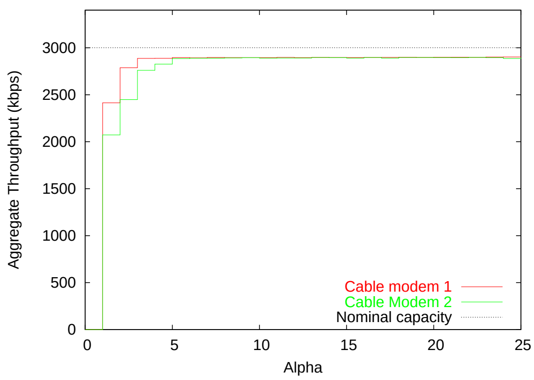

Fig. 10. Throughput versus window size for 6 FAST flows in 2 CMs (3 flows per CM) for DS=3Mbps, US=512Kbps DOCSIS link.

TABLE II
TARGET ALPHA AS A FUNCTION OF NUMBER OF FLOWS IN TWO CMS SYSTEM

Number of flows per CM	*Total alpha required - α_T*	*Total queueing delay (msec)*	*Per-flow alpha α*
1	12	48	6
2	16	64	4
3	30	120	5
4	32	128	4
5	30	120	3
10	60	240	3

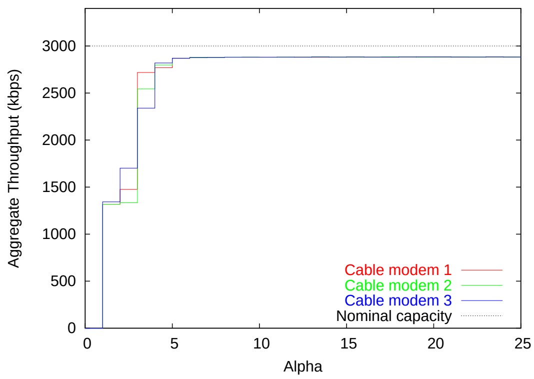

Fig. 11. Throughput versus α for 3 FAST flows in 3 CMs (1 flow per CM) for DS=3 Mbps, US=512 kbps DOCSIS link.

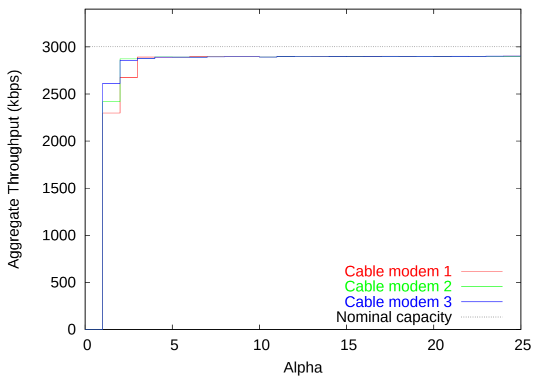

Fig. 12. Throughput versus window size for 9 FAST flows in 3 CMs (3 flows per CM) for DS=3Mbps, US=512Kbps DOCSIS link.

TABLE III
TARGET ALPHA AS A FUNCTION OF NUMBER OF FLOWS IN THREE CMS SYSTEM

Number of flows per CM	*Total alpha required -* α_T	*Total queueing delay (msec)*	*Per-flow alpha* α
1	18	72	6
2	30	120	5
3	36	144	4
4	48	192	4
5	45	180	3
10	90	360	3

delay is inaccurately estimated by *baseRTT*, this will results in unfairness [15]. Note that, this is a realistic occurrence in operational networks as router's queues are never completely empty.

In the following, we evaluate the fairness of FAST and, specifically, investigate how the possible inaccurate estimation of the *baseRTT* would affect FAST. For this analysis, we considered a *persistent congestion* test scenario which demonstrates how a situation may arise where certain flows underestimate their queueing delay relative to other concurrent flows, which subsequently results in unfair share of the resources.

The following tests were performed for both the DOCSIS and the simple link, respectively. For a given run, persistent sources from the same host (TCP server) were added gradually i.e., starting with one flow, every 60 seconds an iperf flow was added, up to 10 flows in total. For this set of experiments, all flows were set with same α parameter, which was initially set to $\alpha = 1$, and all flows shared the same cable modem (receiver). The bottleneck link was configured with DS and US bandwidth of 3 Mbps and 512 kbps, respectively, and with round trip propagation delay of 100 ms (i.e., RTT = 100 ms applied to all flows). We repeated the same experiment with different alpha values for the flows i.e., $1 \geq \alpha \geq 25$.

Interesting questions that we set to investigate are"how many flows are required to observe unfairness" and "how does the fairness compare with the theory". From theory, we expect the first few flows to observe the correct *baseRTT*, and, therefore, to be treated fairly. As the number of flows (and hence mean queue size) increases, we would expect flows to observe higher *baseRTTs*, and start observing unfairness due to persistent congestion. For larger alpha, we would also expect the number of flows required before observing unfairness to decrease.

Figure 13, Figure 14, Figure 15 and Figure 16 plot the observed throughput versus time for all 10 flows using four different FAST control rules: $\alpha = 1$, $\alpha = 2$, $\alpha = 5$ and $\alpha = 10$, respectively. The results show the predicted general behaviour. Each time a new flow is introduced, we see that it achieves the highest throughput at that time. As explained before, this is because a new flow sees a larger *baseRTT* and, hence, estimates a lower queueing delay, which consequently results in getting a higher rate for the flow. In other words, later joining FAST flows underestimate their queueing delay relative to early joining flows and this results in unfair share of the resources. Note, however, that all flows see the same total delay.

It can be further observed, as expected, that for small alpha (i.e., $\alpha = 1$) the number of flows required to observe unfairness is larger compared to higher alpha values. For $\alpha = 1$, the first three flows are treated fairly and only when the 4-th flow is introduced, unfairness is observed, which becomes even more pronounced with

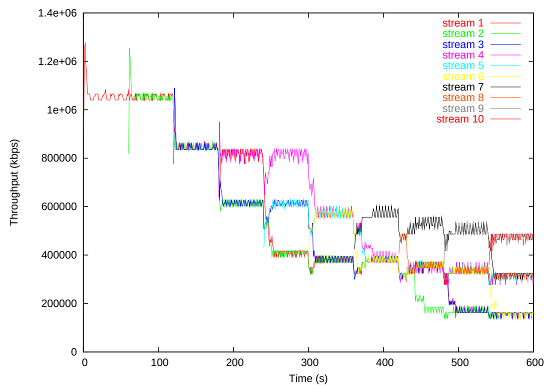

Fig. 13. Throughput versus time for 10 FAST flows set with $\alpha = 1$ for DS=3 Mbps, US=512 kbps simple link.

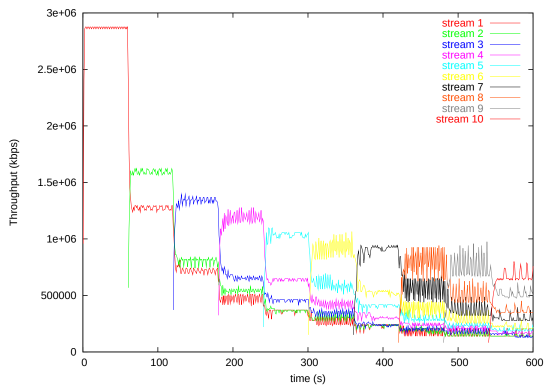

Fig. 15. Throughput versus time for 10 FAST flows set with $\alpha = 5$ for DS=3 Mbps, US=512 kbps simple link.

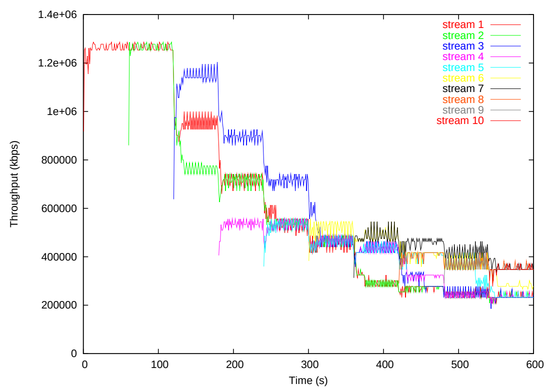

Fig. 14. Throughput versus time for 10 FAST flows set with $\alpha = 2$ for DS=3 Mbps, US=512 kbps simple link.

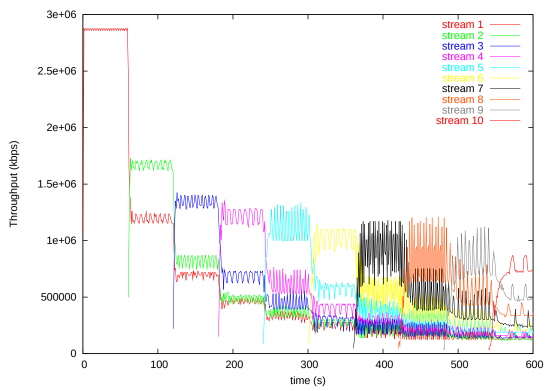

Fig. 16. Throughput versus time for 10 FAST flows set with $\alpha = 10$ for DS=3 Mbps, US=512 kbps simple link.

the further increase in the number of flows. When the FAST control rule is set to $\alpha \geq 1$, the number of flows before unfairness starts to occur rapidly decreases e.g., for $\alpha = 2$ the number of flows is 3 (as shown on Figure 14) and for $\alpha \geq 3$ ($\alpha = 5$ and α=10 are shown in Figure 15 and Figure 16, respectively) the number of flows is 2. Note that, for a single FAST flow, $\alpha \geq 3$ was required to achieve full utilisation on a 3 Mbps/512 Kbps simple link (see Section IV) suggesting that in ideal operating conditions unfairness will be experienced by every flow except for the first one.

In the case of the DOCSIS link, because of the extra randomness in the queue size, it can be seen that the unfairness is even more pronounced with the increase in the number of flows (see Figure 17). We summarise the DOCSIS results in Figure 18, by plotting the ratio of the throughout achieved by the last and the second last flow for all flows and various alpha values (1, 3, 5, 10, and 15). For example, $N = 2$ represents the ratio between the throughput of the second added flow and the very first flow on the link, averaged during the time interval in which the second flow is introduced and, similarly, $N = 5$ is the ratio between the throughput of the fifth and the fourth added flow, respectively. The results are averaged across 10 runs. From theory, we would expect for small alpha values this throughput ratio to be close to 1 and it should be slightly increasing with the increase of N. In general, as expected, we observe that the larger alpha the more unfairly the old flows are treated, which have an accurate estimate of their propagation delay. However, due to the additional randomness in the queueing that DOCSIS introduces there is slight oscillation around the expected values, especially, for small alpha values.

Although, the presented experimental results are conclusive in that they match well the expectation from theory, for future work it would be interesting to investigate the fairness of FAST with different RTTs for the persistent sources, configured with same alpha parameters. For example, one can consider dynamic scenario, where new flows are added with RTTs much

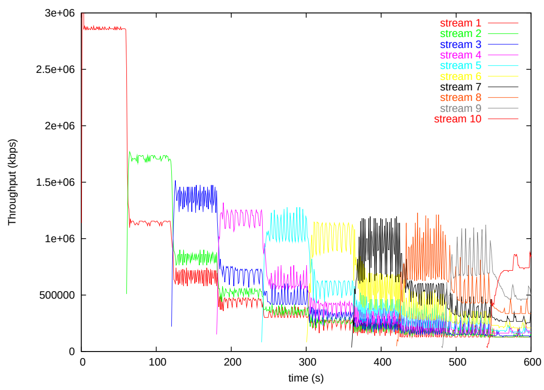

Fig. 17. Throughput versus time for 10 FAST flows set with $\alpha = 10$ for DS=3 Mbps, US=512 kbps DOCSIS link.

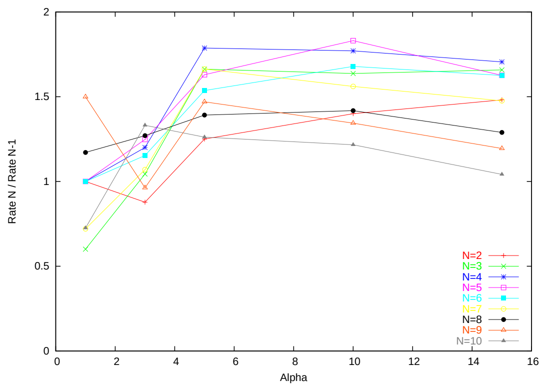

Fig. 18. Throughput versus time for 10 FAST flows set with $\alpha = 10$ for DS=3 Mbps, US=512 kbps simple link.

larger than the old flows, and vice versa, and investigate if the number of sources required to observe persistent queueing increases, decreases or is unchanged (in both DOCSIS and simple link).

VI. Conclusions

We have investigated the performance of FAST over simple, as well as DOCSIS-based low-speed links, and in particular extended the study of [1] to consider test scenarios with multiple FAST flows sharing a single bottleneck link. Consistent with our previous study, FAST achieves full utilisation over a low-speed link if its target queue size (alpha) is at least three packets. In the case of DOCSIS, we observed that the trend of increasing target queue size, which was quite striking in the single cable modem system, does not seem to continue when multiple cable modems are included in the access network. The target alpha value was consistently below FAST's current default value of $\alpha = 20$, and also tend to asymptote to the "simple link" target value ($\alpha = 3$) with the increase in the number of flows, suggesting that FAST, with its current default settings, is able to achieve full utilisation in realistic low speed access networks.

We also considered the issue of unfairness associated with FAST TCP operation due to inaccurate estimation of the round-trip propagation delay. Using a "persistent congestion" test scenario we demonstrated how certain flows, specifically, later joining FAST flows may consistently underestimate their queueing delay relative to early joining flows, which results in unfair share of the resources. The extensive results showed that fairness is adversely affected by the increase in the number of flows. Also the larger alpha the more unfairly the old flows are treated, which have an accurate estimate of their propagation delay. Finally, in the case of cable modem links, the unfairness is even more pronounced, as a result of the additional delay that the MAC layer mechanism (i.e., Request and Grant Cycle) of DOCSIS introduces.

Acknowledgment

We thank Cisco Systems Australia for donating equipment, and L. Stewart of CAIA for assistance setting up the network.

References

[1] L. H., Andrew, I. Atov, D. Kennedy, and B. Wydrowski, "Evaluation of FAST TCP in Low-Speed DOCSIS-based Access Networks", in *Proc. of IEEE TENCON 2005*, pp. 1680-1685, Melbourne, Australia, November 2005.

[2] D. Kennedy and I. Atov, "Implementing a Testbed for the Evaluation of FAST TCP in DOCSIS-based Access Networks", Technical Report 060119A, Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, January 2006.

[3] J. Postel, *Transmission Control Protocol*, STD 7, IETF RFC 793, September 1981. Available at http://www.ietf.org/rfc/rfc793.txt.

[4] M. Allman, V., Paxson, and W. Stevens, *TCP Congestion Control*, IETF RFC 2581, April 1999. Available at http://www.ietf.org/rfc/rfc2581.txt.

[5] L. Xu, K., Harfoush, and I. Rhee, "Binary Increase Congestion Control for Fast Long-Distance Networks", in *Proc. of IEEE INFOCOM 2004*, pp. 2514-2524, Hong Kong, March 2004.

[6] T. Kelly, "Scalable TCP: Improving Performance in Highspeed Wide Area Networks", *ACM SIGCOMM Computer Communication Review*, Vol.33, No. 2, pp. 83-91, 2003.

[7] C. Jin, D. X., Wei, and S. H. Low, "FAST TCP: Motivation, Architecture, Algorithms, Performance", in *Proc. of IEEE INFOCOM 2004*, pp. 2490-2501, Hong Kong, March 2004.

[8] J. Wang, D. X., Wei, and S. H. Low, "Modeling and stability of FAST TCP", in *Proc. of IEEE INFOCOM 2005*, Miami, FL, March 2005. Available at http://netlab.caltech.edu/pub/papers/FASTstability-infocom05.pdf.

[9] S. Floyd, *HighSpeed TCP for Large Congestion Windows*, IETF Internet Draft, draft-floyd-tcp-highspeed-02.txt, 2002.

[10] L. S. Brakmo and L. L. Peterson, "Vegas: End-to-End Congestion Avoidance on a Global Internet", *Journal on Select Areas in Communications*, Vol.13, No. 8, pp. 1465-1480, 1995.

[11] H. Bullot, R. L. Cottrel, and R. Highes-Jones, "Evaluation of Advanced TCP Stacks on Fast Long-Distance Production Networks", *Journal on Grid Computing*, Vol.1, No. 4, pp. 345-359, Dec 2003.
[12] O. Elloumi, et. al., "A Simulation-based Study of TCP Dynamics over HFC Networks", Computer Networks, Vol. 32, No. 3, pp. 301-317, 2000.
[13] J. Martin, "The Interaction Between the DOCSIS 1.1/2.0 MAC Protocol and TCP Application Performance", HET-NET'04.
[14] J. Martin, "The Impact of DOCSIS 1.1/2.0 MAC Protocol on TCP", in *Proc. of IEEE Consumer Communications and Networking Conference*, Las vegas, January 2005.
[15] L.L.H. Andrew, L. Tan, T. Cui, and M. Zukerman, "Fairness Analysis of FAST TCP and TCP Vegas", in *Proc. of International Teletraffic Congress ICT-19*, Beijing, China, 2005.
[16] CableLabs, *Data-Over-Cable Service Interface Specifications Radio Frequency Interface Specification SP -RFIv1.1-I01-990311*,1999.
[17] Broadband Access Research Testbed, Centre for Advanced Internet Architectures, Swinburne University of Technology, http://caia.swin.edu.au/bart.
[18] T. T. T., Nguyen and G. J. Armitage, "Experimentally derived interactions between TCP traffic and service quality over DOCSIS cable links", in *Proc. of IEEE GLOBCOM 2004*, Texas, USA, November 2004.
[19] S. Hegde, D. Lapsley, B. Wydrowski, J. Lindheim, D. Wei, C. Jin, S. Low, H. Newman, "FAST TCP in High-Speed Networs: An Experimental Study, *Proc. of GridNets*, San Jose, CA, October 2004.
[20] C. Jin, D. Wei, and S. H. Low, "FAST TCP for High-Speed Long-Distance Networks," internet draft draft-jwl-tcp-fast-01.txt, [Online]. Available http://netlab.caltech.edu/pub/papers/draft-jwl-tcp-fast-01.txt.
[21] (2004) Understanding Data Throughput in a DOCSIS World. [Online]. Available: http://www.cisco.com/en/US/tech/tk86/tk168/ technologies_tech_note09186a0080094545.shtml
[22] Cisco uBR7100 Series Software Configuration Guide. [Online]. Available: http://www.cisco.com/en/US/products/hw/cable/ps2211/products_configuration_guide_chapter09186a00801b355a.html#wp1021916.
[23] L. Rizzo, "Dummynet", [Online]. Available: http://info.iet.unipi.it/∼luigi/ip_dummynet/.
[24] Iperf - The TCP/UDP Bandwidth Measurement Tool [Online]. Available: http://dast.nlanr.net/Projects/Iperf/.
[25] tcpdump [Online]. Available: http://www.tcpdump.org/.