CENTRE FOR ADVANCED INTERNET ARCHITECTURES

Evaluation of FAST TCP in Low-Speed DOCSIS-based Access Networks

by Irena Atov

Joint work with:
L.H. Andrew, D. Kennedy and B. Wydrowski

Presentation Outline

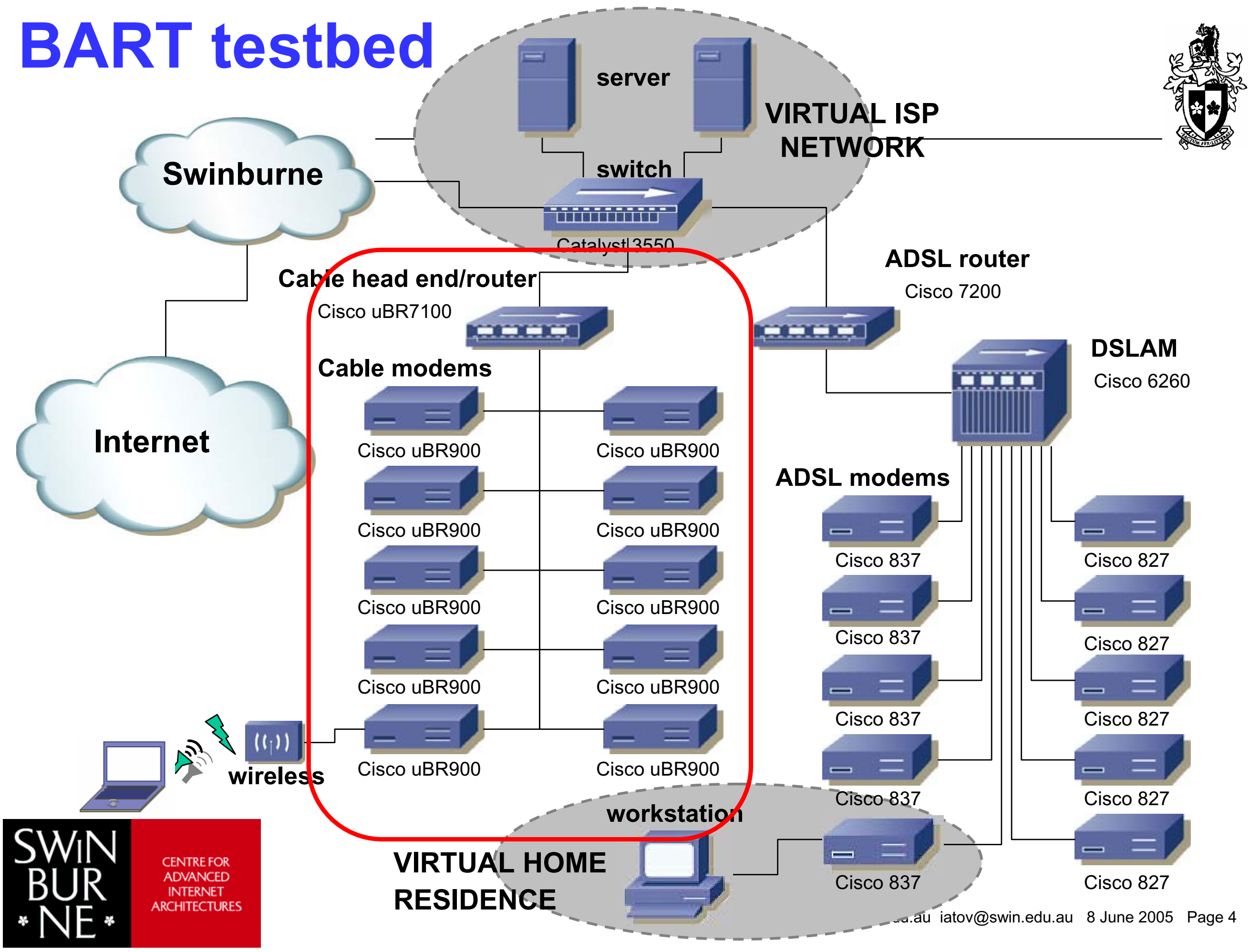
- Motivation
- Overview of FAST and DOCSIS protocols
- Experimental setup to study the behaviour of FAST in two different low-speed environments
- Results and analysis
- Conclusions and future work

Motivation

- It is evident that the efficiency of the Internet is limited by its existing TCP congestion control system (TCP Reno)
- A replacement, called FAST TCP, is being designed at Caltech to improve performance in high-speed networks
- For its standardization and deployment, it must be tested in a wide variety of environments
- To date, FAST has been tested by Caltech and independent groups such as SLAC and CERN in a wide range of ***high speed environments***.
- This project aims to experimentally evaluate the performance of FAST in ***low speed environments*** i.e., over a typical access link, with bandwidths of around 0.5 – 3 Mbps
- Specifically, to investigate links both using the DOCSIS cable modem medium access control (MAC) protocol and simple low rate links
- The understanding gained will allow optimal parameter settings to be determined for a range of conditions.

SWINBURNE CENTRE FOR ADVANCED INTERNET ARCHITECTURES

BART testbed



SWINBURNE CENTRE FOR ADVANCED INTERNET ARCHITECTURES

FAST overview

- TCP regulates source sending rate by adapting window size according to some congestion signal:
  - Packet-loss-based: Most TCP algorithms since TCP Tahoe regulate source sending rate by adapting window size according to the packet loss rate ← Reno, HSTCP, BIC
  - Delay-based: Flow rates are adjusted in response to the measured delay. These algorithms attempt to maintain for a flow a constant number of packets, α, queued in nodes along its path ← FAST, Vegas
- FAST updates the window size according to:

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

D – average RTT
d – base RTT

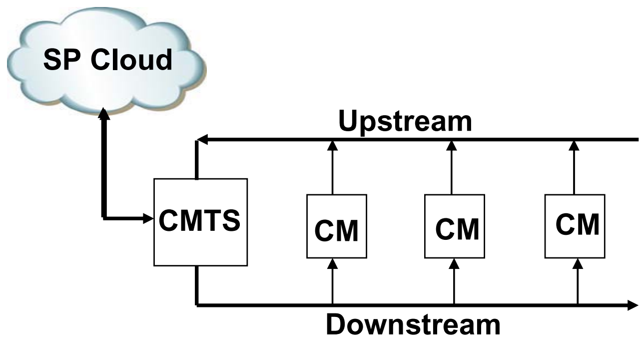
SWIN BUR NE CENTRE FOR ADVANCED INTERNET ARCHITECTURES

Focus of our work

- The focus of our work is how to set the alpha parameter in two different low-speed environments
- High-speed: For high-speed links, it has been recommended that α be set to cause a given small queueing delay, such as ~2ms i.e., α = 2C, q = 2C/C
- Low-speed: In this work we show that this rule of thumb gives insufficient queueing for low-speed networks, especially when DOCSIS is used

SWIN BUR NE CENTRE FOR ADVANCED INTERNET ARCHITECTURES

Overview of DOCSIS

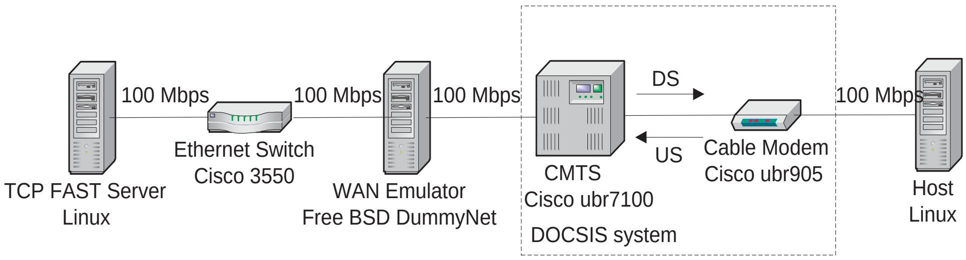


Logical topology of DOCSIS network

DOCSIS networks operate on a reservation scheme (known as *Request-and-Grant* cycle) where the modems request a time to transmit and the CMTS grants time slots based on availability.

- It has been shown that the DOCSIS system introduces latency fluctuations based on the offered load in the system (T. T. T., Nguyen and G. J. Armitage 2004)
- Our goal is to investigate how factors such as the Request-and-Grant control of transmission cycles between the CMTS and the CM can affect the performance of a FAST TCP application over DOCSIS network and to identify factors that need to be considered in achieving optimal performance over a DOCSIS cable network

Experimental Setup

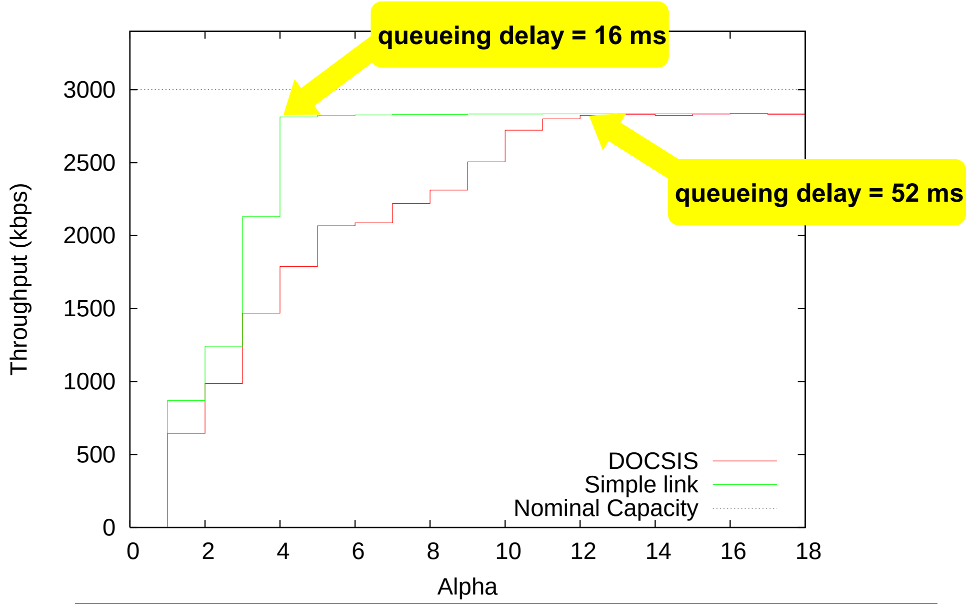


DS	US
256 Kbps	128 Kbps
512 Kbps	256 Kbps
1.54Mbps	512 Kbps
2 Mbps	512 Kbps
2.5 Mbps	512 Kbps
3 Mbps	512 Kbps

- We have experimentally evaluated the performance of FAST over two different access networks, each with a single bottleneck link:
  - DOCSIS link, and
  - simple rate-limited link.
- Static scenarios were considered i.e, the bottleneck link carried one or two FAST flows, and no other traffic (iperf flows with 1500-byte in the downlink)
- DOCSIS link
  - Dummynet: RTT=100ms, no bandwidth limitations US & DS, buffer size of 2048 Kbytes
  - Maximum buffer size at CMTS set to the default Cisco value of 512ms
- Simple link - DOCSIS system was bypassed. Dummynet emulated system with equivalent US and DS capacities and buffering. Also, RTT=100ms.

Single Flow Results

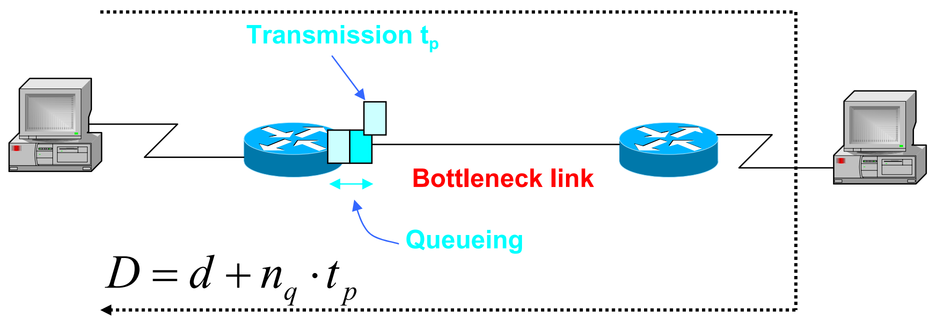
Throughput vs. α parameter for DS = 3 Mbps and US = 512 Kbps (results are averaged over 100 runs)



Much larger queueing is needed than what is necessary to obtain accurate timing estimates!

FAST on a simple slow link

- There are two major effects at work causing utilisation to be low for $\alpha \leq 3$:
  - Burstiness – causes the average queue size observed by the packets to be greater than the true mean queue size. FAST attempts to minimise burstiness, but is hindered by TCP delayed acknowledgements (RFC 2581).
  - The integer arithmetic of FAST control mechanism



$$D = d + n_q \cdot t_p$$

- Even for very low utilisation, delayed ACKs alone can allow D to be up to $d+t_p$ (tp is the packet delay).
- Thus, FAST requires $\alpha \geq 1$ to achieve full utilisation

FAST on a simple slow link – cont.

- The second reason for requiring large α is the floor operation in (1)

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

- Without the $\lfloor \cdot \rfloor$ operation, the update rule satisfies the equilibrium relationship

$$w(t+1) = \frac{1}{2}\left(w(t) + \frac{d}{D} w(t) + \alpha \right) \quad (2) \quad \Rightarrow \quad w = \alpha \frac{D}{D-d}$$

- However, at equilibrium of (1) we know:

$$w \le \frac{1}{2}\left(w + \frac{d}{D} w + \alpha \right) \le w + 1 \quad \Rightarrow \quad (\alpha - 2)\frac{D}{D-d} \le w \le \alpha \frac{D}{D-d}$$

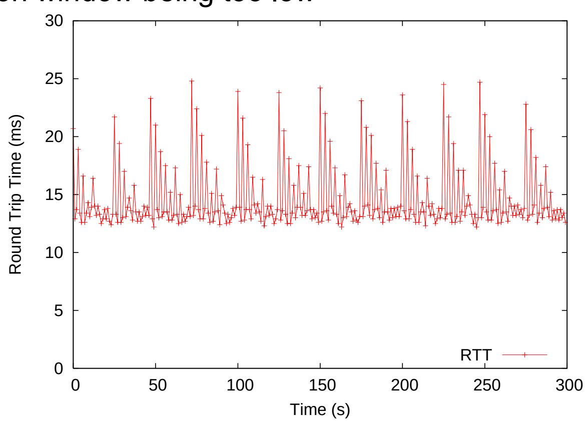
- So that equilibrium window size can be as small as that predicted by (2) under the substitution $\alpha' = \alpha - 2$

Thus, we expect that $\alpha \ge 3$ for full utilisation

SWINBURNE CENTRE FOR ADVANCED INTERNET ARCHITECTURES

FAST on a DOCSIS link

- FAST achieves much lower throughput in a DOCSIS system (e.g, at 3Mbps it needed α=13 for full utilisation, compared to α=4 on a simple link)
- There are several possible reasons for this discrepancy:
  - The latency fluctuations introduced by the DOCSIS system could possible interfere with FAST estimates of the queueing in the network, resulting in the congestion window being too low



Selected packets experience up to 13ms delay

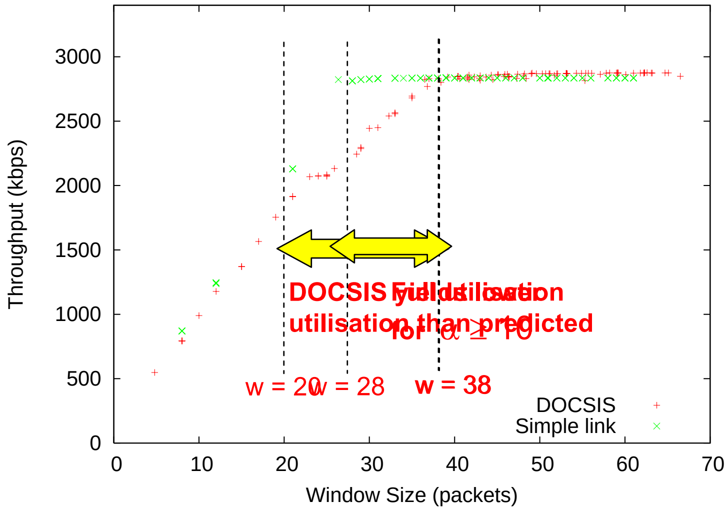
  - Another possibility is that the actual window size required for full utilisation in DOCSIS is larger than the B-D product

SWINBURNE CENTRE FOR ADVANCED INTERNET ARCHITECTURES

FAST on a DOCSIS link – cont.

Throughput vs. window size for DS = 3 Mbps and US = 512 Kbps for 140 experiments using α values from 1 to 35

- A bottleneck link carrying a single flow in a purely deterministic network will be fully utilised if the flow's window size is at least the ``bandwidth delay product''
- 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

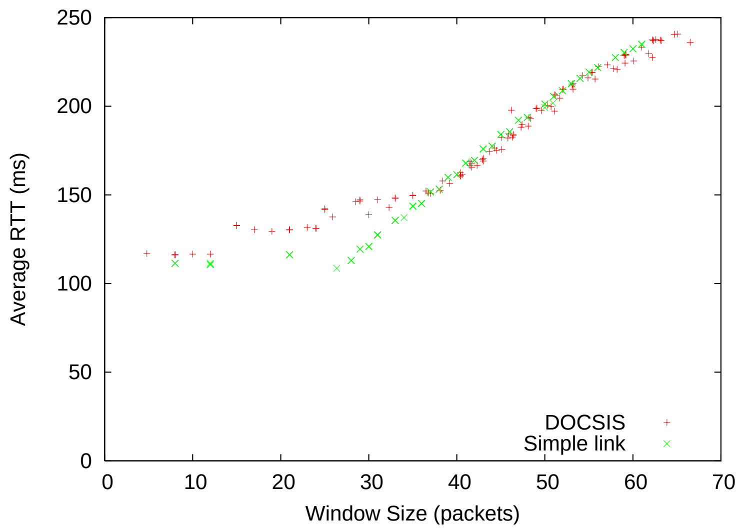


This suggests that FAST window size is not adversely affected by the randomness of the delay

FAST on a DOCSIS link – cont.

Average RTT vs. window size for DS = 3 Mbps and US = 512 Kbps DOCSIS and simple link

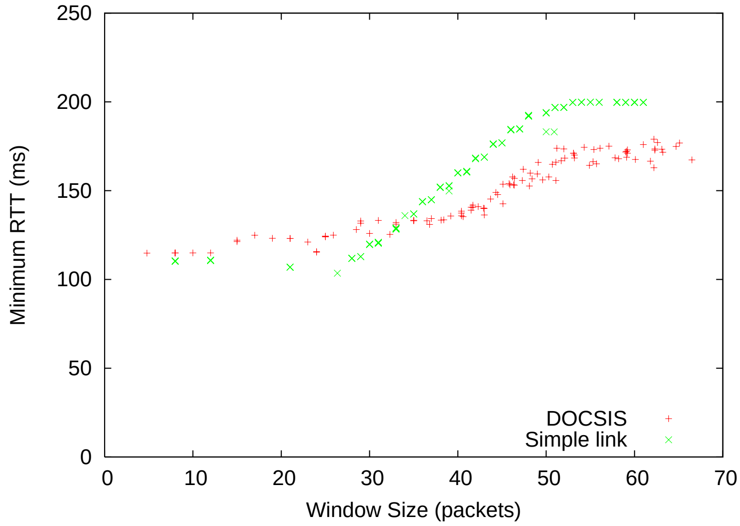
- For underutilised link, the queueing delay should be zero
- For fully utilised link the queueing delay should be proportional to the difference between the window size and the bandwidth delay product
- This is observed for a simple link, but DOCSIS shows a higher delay for windows between about 13 and 38 packets, with a value of approximately 145 ms between about 20 and 38 packets



FAST on a DOCSIS link – cont.

Average minimum RTT vs. window size for DS = 3 Mbps and US = 512 Kbps DOCSIS and simple link

- The "phantom delay" cannot be attributed to burstiness, as it would be expected that at least some packets every round trip time would observe approximately the true propagation delay.
- Thus, it is not simply FAST's estimate of the RTT that has increased, but rather the actual RTT



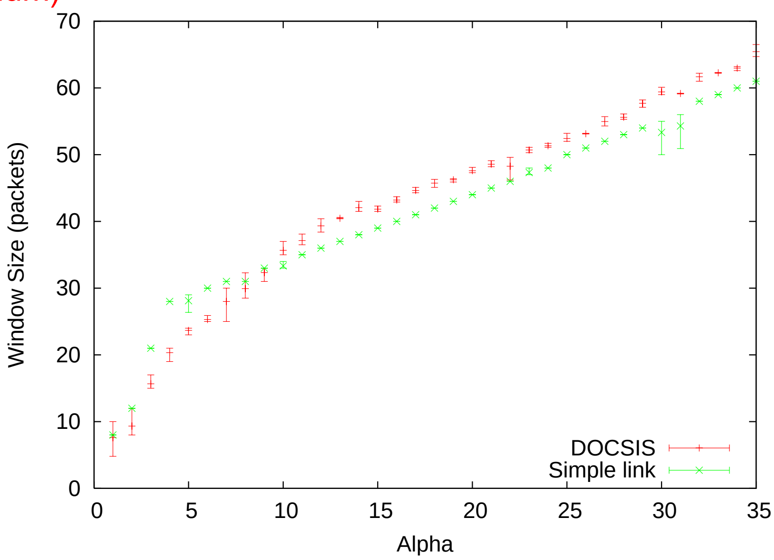
This indicates that DOCSIS may buffer packets even when the link is idle (due to the Request-and-Grant scheme it employs)

SWiNBURNE CENTRE FOR ADVANCED INTERNET ARCHITECTURES

FAST on a DOCSIS link – cont.

Window size calculated by FAST as a function α for DS = 3 Mbps and US = 512 Kbps, DOCSIS and simple link (error bars indicating the maximum and minimum)

- Other possible reason for requiring large α is that the DOCSIS delay causes FAST to set the window size too small
- For small α, when the link is not fully utilised, FAST underestimates w using DOCSIS, reflecting the phantom delay
- Once the link reaches full utilisation, the phantom delay is ``absorbed'' into the queueing delay, and the window size is no longer too small

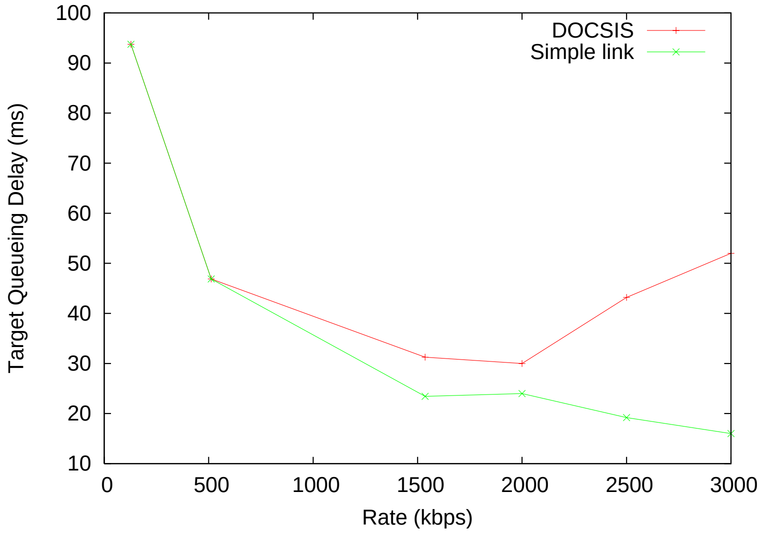


SWiNBURNE CENTRE FOR ADVANCED INTERNET ARCHITECTURES

FAST on a DOCSIS link – cont.

Summary of results for a single flow - Total target queueing delay required by FAST as a function of link capacity

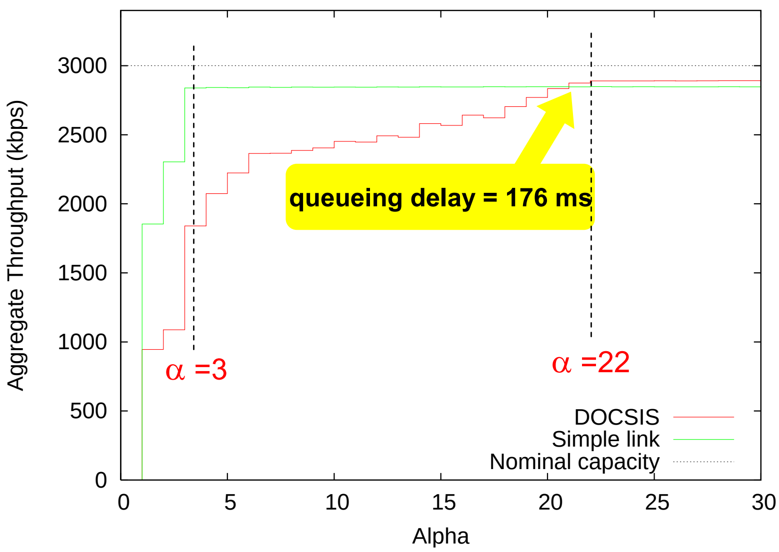
- For very low rates, full utilisation is achieved with α =1 or 2
- As the rate increases, the queueing required on a simple link decreases monotonically, and will asymptote to a value which gives a queueing delay just large enough to be reliably detected
- In contrast, the queueing delay required over DOCSIS increases again as the rate increases



Two Flow Results

Aggregate throughput vs. α parameter for DS = 3 Mbps and US = 512 Kbps for 2 flows - α was set equal for the 2 flows (α = 1, 2…,35) and results are averaged over 10 runs

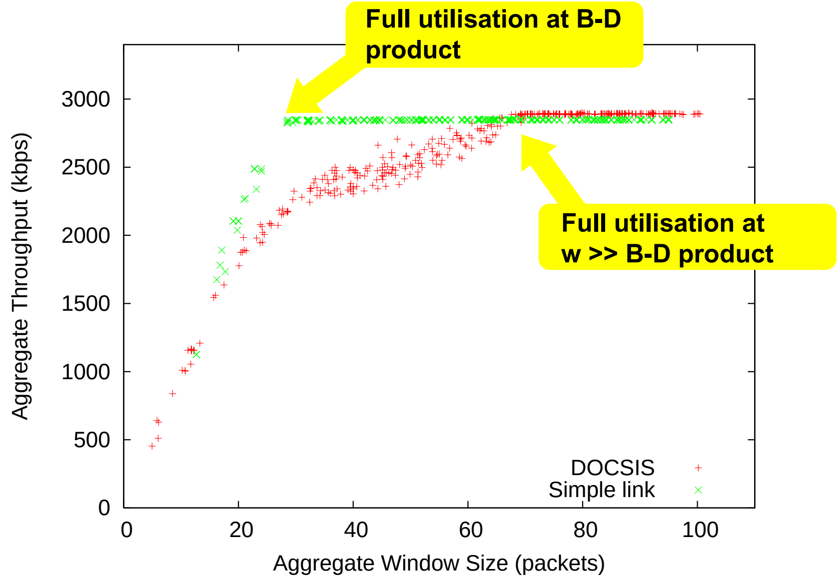
- When *n* flows share a single bottleneck link, the total queueing at the link is ideally $n\alpha$
- Thus, we would expect the required α value to scale inversely with *n* i.e, for 2 two flows, we would expect α to halve
- E.g., for DOCSIS link and 3M/512K (we needed α=13 and q=52ms for single flow) and we expect each flow to need α=7, which would again give a total queuing delay of 52ms



For DOCSIS, the required α actually increased to 22! That corresponds to a total target queue size of 44 packets or a delay of 176 ms.

Two Flow Results – cont.

Aggregate throughput vs. aggregate window for DS = 3 Mbps and US = 512 Kbps



- The trend of superlinear buffer requirements is concerning, as the CMTS had a default ``traffic shaping'' buffer with maximum delay 512 ms, which can be increased to at most 1028 ms
  - the default buffer could support fewer than 512/(176/2) ≈ 6 flows at full utilisation!

SWINBURNE CENTRE FOR ADVANCED INTERNET ARCHITECTURES

Conclusions and Future work

- The performance of FAST TCP was investigated over low speed links, and in particular links running DOCSIS.
- Main findings:
  - FAST is able to achieve almost full utilisation over a low speed link if its target queue size (alpha) is at least three packets.
  - This corresponds to a decreasing queueing delay as the data rate increases.
  - In contrast, DOCSIS introduces additional latency, which requires both alpha and the target queueing delay to increase as the link capacity increases beyond 1 Mbps, up to alpha=12 for 3Mbps.
  - Although DOCSIS also introduces unpredictable delays, these do not appear to interfere with FAST's ability to estimate the queueing delay.
- Future work:
  - Study the mechanisms by which DOCSIS introduces the additional delays
  - Derivation of an analytic model of the interaction between DOCSIS and FAST

SWINBURNE CENTRE FOR ADVANCED INTERNET ARCHITECTURES

Thank You

…any questions?