SWINBURNE UNIVERSITY OF TECHNOLOGY

Empirical TCP Research

Some Thoughts, Tools & Results

Lawrence Stewart

lastewart@swin.edu.au

Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology

Outline

Detailed outline (section 1 of 4)

1 TCP Recap

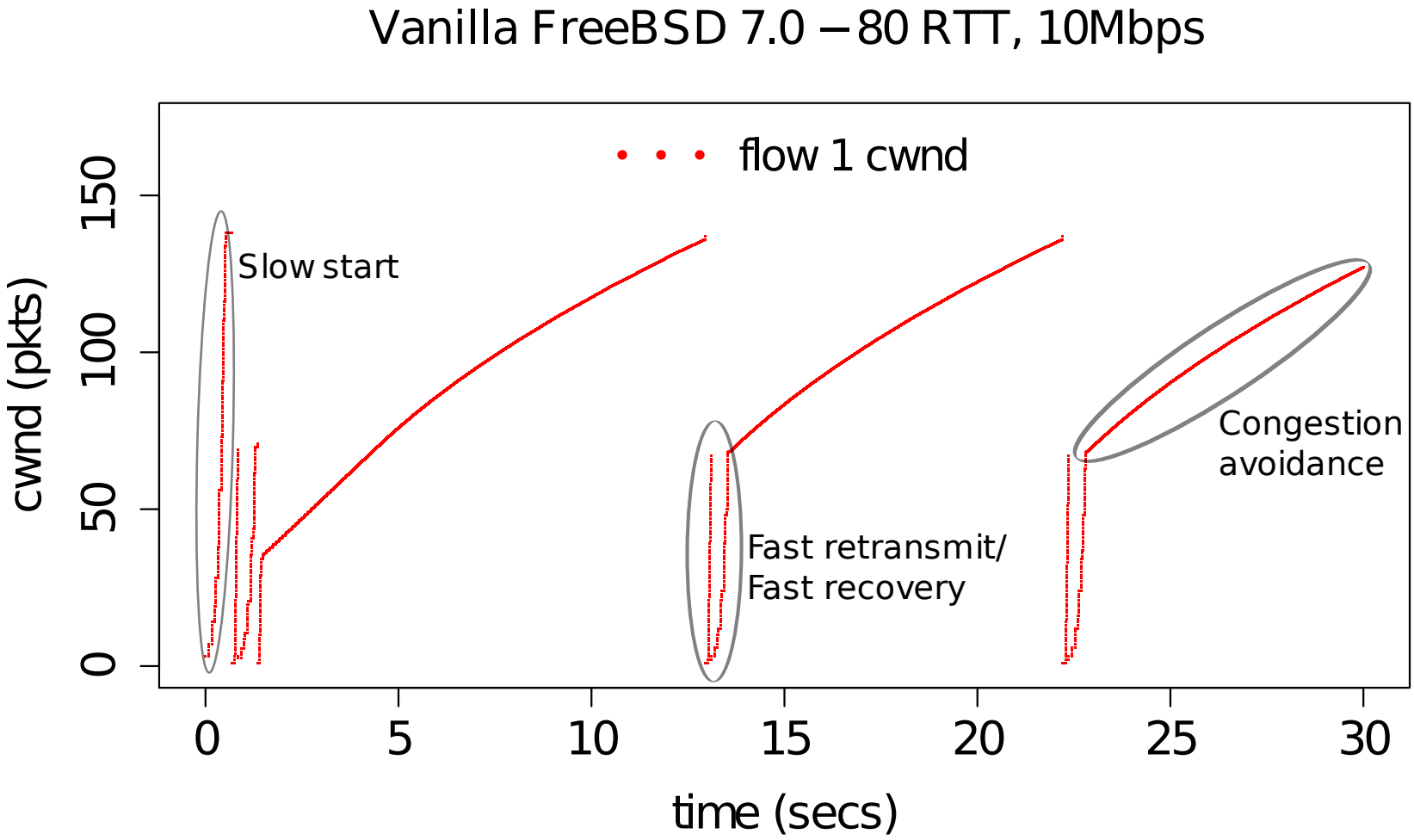
- Key Facts
- Where are we today
- Open issues

Key Facts

- Core TCP modes of operation [1]
  - Slow start
  - Congestion avoidance
  - Fast retransmit
  - Fast recovery
- Many protocol tweaks and additions along the way
  - SACK, ABC, ECN, window scaling, timestamps, etc.
- RFC 4614 provides a good summary of TCP related RFCs

[1] See RFC2001

Key Facts


Vanilla FreeBSD 7.0 −80 RTT, 10Mbps
flow 1 cwnd
Slow start
Fast retransmit/
Fast recovery
Congestion
avoidance
cwnd (pkts)
0
50
100
150
0
5
10
15
20
25
30
time (secs)

Where are we today

- Many incremental (partially implemented) improvements
- State of the CC union
  - NewReno is defacto standard with warts (LFN, wireless)
  - Many new proposals
  - BSD still uses NewReno
  - Linux uses CUBIC
  - Windows Vista uses CTCP
- TCP/IP stack enhancements e.g.
  - CSO/TSO/LRO/TOE
  - Various locking/caching tricks
  - Socket buffer autotuning

Open issues

- High-speed CC algorithms [2]
  - FAST, HS-TCP, H-TCP, CTCP, CUBIC, etc.
- Delay based CC algorithms
- How do we compare and evaluate TCPs?
- Multipath
- CSO/TSO/LRO/TOE obscure behaviours
- Testing/verification of TCP/IP stack behaviour

[2] Nice summary:
http://kb.pert.geant2.net/PERTKB/TcpHighSpeedVariants

Detailed outline (section 2 of 4)

2 Tools

- Modular congestion control
- Web100/SIFTR
- Dummynet
- TCP Testbed

Modular congestion control

- Facilitates:
  - TCP CC research e.g. ICCRG, TMRG, PFLDnet
  - Standardisation process i.e. RG -> WG, I-D -> RFC
- Catering to specialised applications
  - Select most appropriate CC algorithm for the task
- Available in FreeBSD [3], NetBSD, Linux
- Ultimately a better Internet (hopefully!)

[3] Sponsored by Cisco Systems

Web100/SIFTR

- Gather in-kernel TCP endpoint connection data as CSV
  - ssthresh, cwnd, RTT, etc.
- Web100
  - Linux patchset
  - Poll based
- Statistical Information For TCP Research (SIFTR)
  - FreeBSD [6,7,8] kernel module [4]
  - Event based

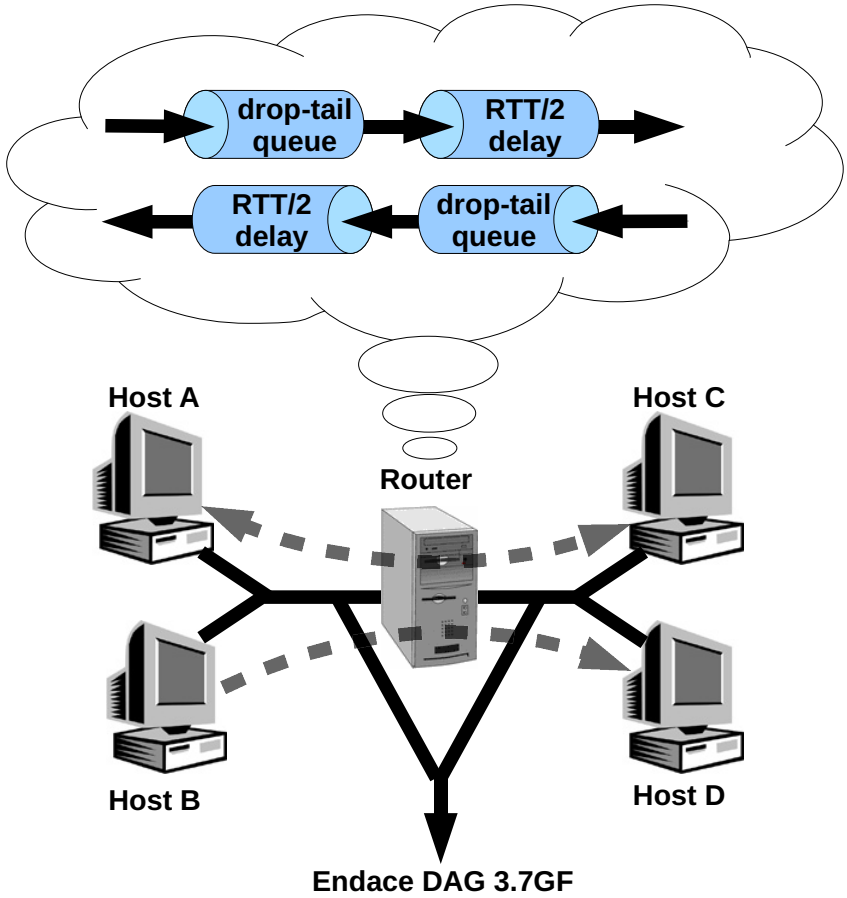
[4] Sponsored by Cisco Systems and the FreeBSD Foundation

Dummynet

- Deterministic Packet Discard (DPD)
  - Patch against FreeBSD 8.x IPFW/Dummynet
  - Useful for protocol (not just TCP!) verification and testing
  - Adds 'pls' (packet loss set) option for dummynet pipes
  - e.g. ipfw pipe 1 config pls 1,5-10,30 would drop packets 1, 5-10 inclusive and 30
- Forensic logging support
  - Patch against FreeBSD 8.x Dummynet
  - Log queue state on each packet event

Testbed

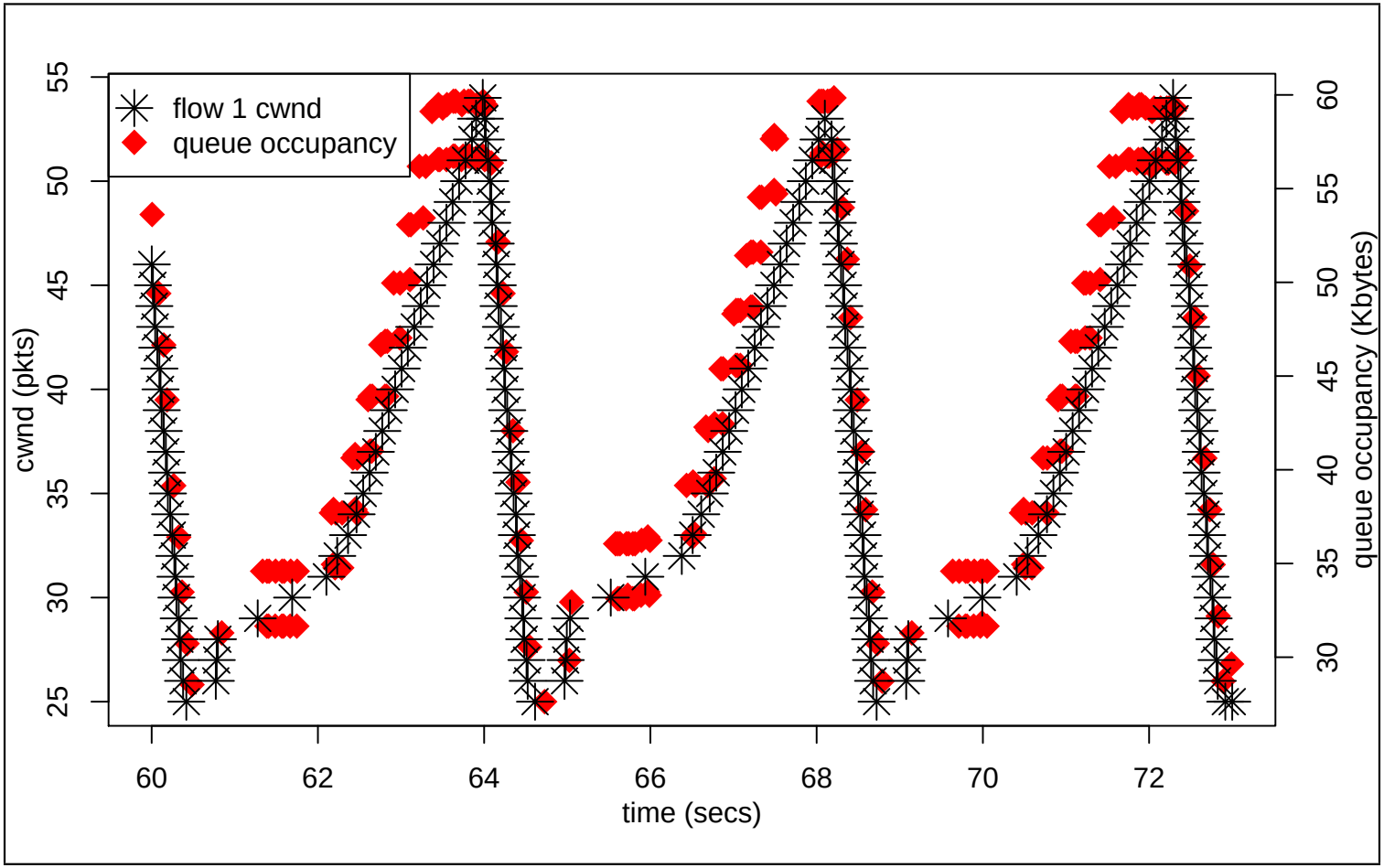
- Linux/FreeBSD hosts
- Modular congestion control
- Web100/SIFTR for Linux/FreeBSD testing
- Iperf/Tcpreplay for traffic generation
- FreeBSD dummynet router
- Endace DAG 3.7GF capture card
- Characterise your kit!



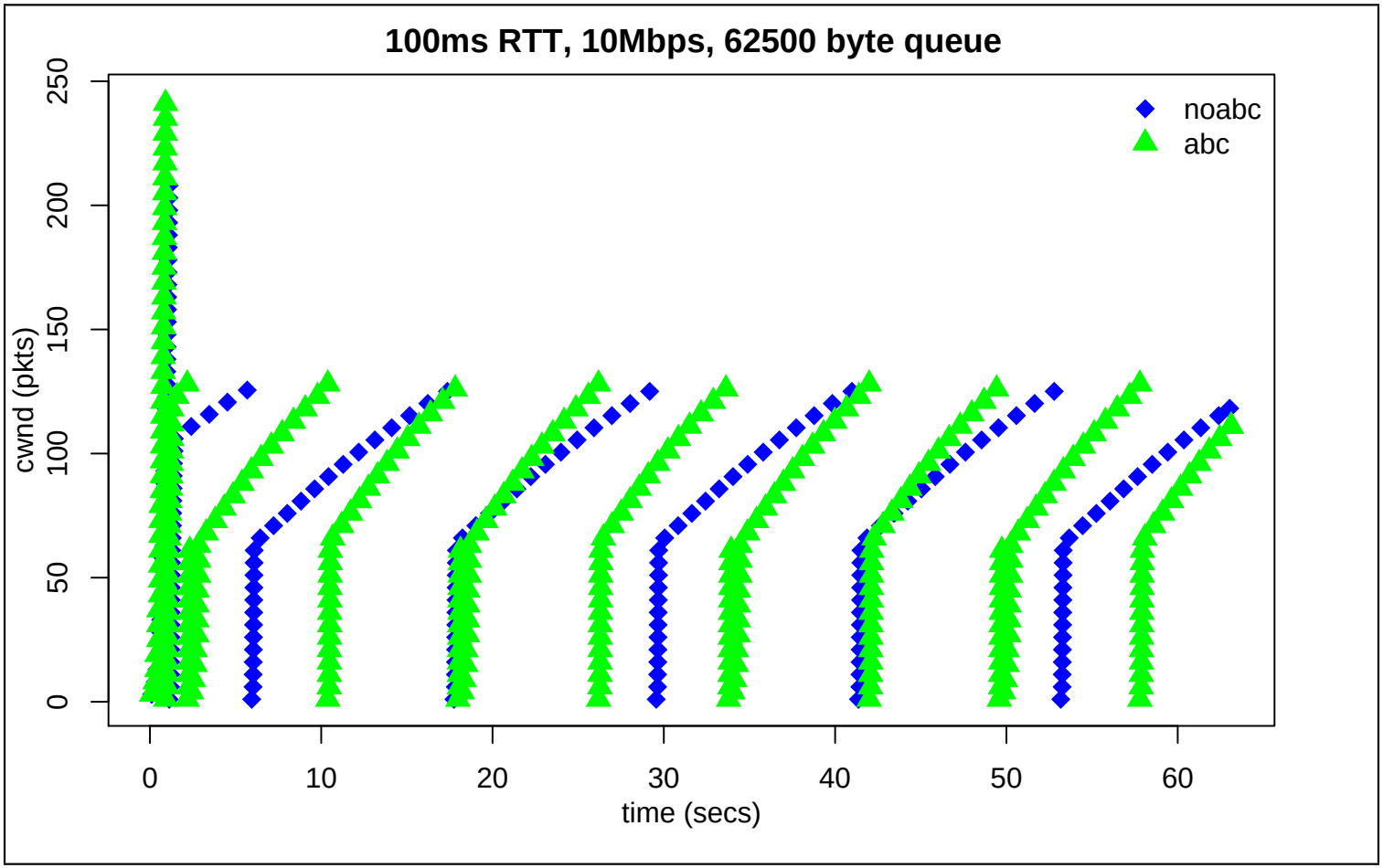
Detailed outline (section 3 of 4)

Connection Dynamics

- 1 TCP flow, H-TCP, 100ms RTT, 1Mbps, 60000 byte queue



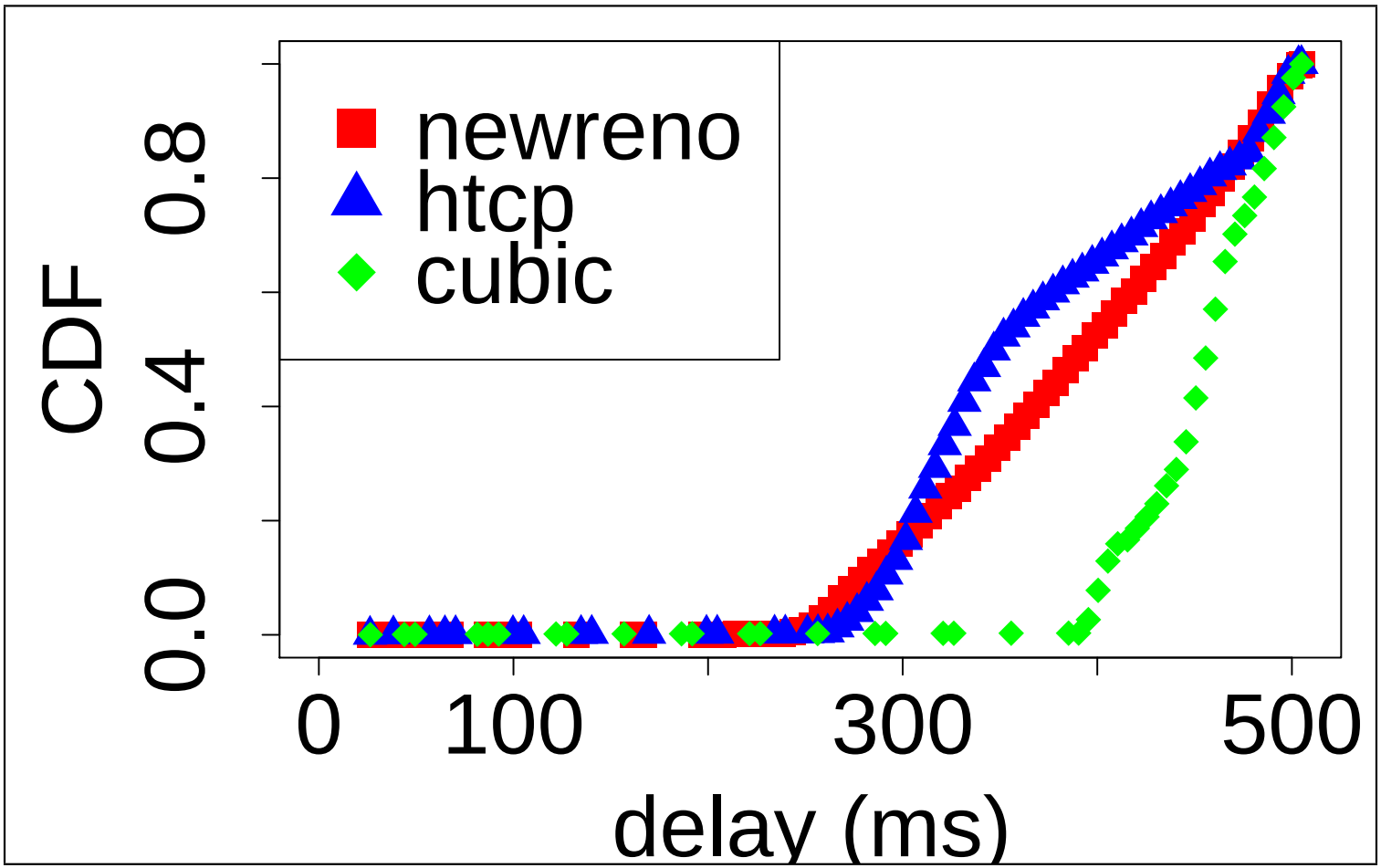
Appropriate Byte Counting (ABC)



- Sponsored by the FreeBSD Foundation

Collateral Damage

- Induced delay: 1 TCP flow, 50ms RTT, 1Mbps, 60000 byte queue



Detailed outline (section 4 of 4)

4 Wrapping Up

- Acknowledgements
- Questions

Acknowledgements

- Cisco Systems

- The FreeBSD Foundation

- University of Cambridge, Computer Laboratory

Fin

Questions?