An Implementation and Experimental Study of the eXplicit Control Protocol (XCP)

Yongguang Zhang and Tom Henderson
INFOCOMM 2005

WPI

Presenter - Bob Kinicki

Outline

- Overview of XCP Study
- XCP Implementation Details
- XCP Experimental Evaluation
- XCP Sensitivity Study
- Conclusions and Critique

Overview

- The XCP paper involved control theory laws/proofs and ns-2 simulation results.
- This paper involves implementing XCP in Linux and conducting a testbed evaluation.
- While initial validation results match reported simulation results, this paper exposes: implementation challenges, precision and range issues for XCP, and environmental factors that adversely affect XCP performance.

Implementation Details

- XCP implemented as a TCP option.
- The use of XCP negotiated on connection setup.
- XCP congestion header encoded as a TCP header option and attached to the outgoing TCP header.

XCP Option Format

Forward direction:

opt	optsize	
14	8	H_feedback
H_rtt		H_cwnd

Return direction:

opt	optsize	
15	4	H_feedback

Fig. 1. XCP option formats in both directions

- Each of the three XCP congestion header fields are 16 bits.
- H_cwnd and H_feedback measured in packets and represented as split mantissa-exponent.
- H_rtt measured in millisec. and stored as unsigned integer.

XCP functionality from TCP

- XCP implementation retained fast retransmit and cwnd reset to one on timeout.
- TCP slow start and congestion avoidance disabled.
- Advertised window modified to grow by feedback_rcv.
- To address wireless packet losses, XCP implemented with two SACK blocks.

XCP Router

- XCP Router function divided into:
  - Kernel support and interface
  - XCP congestion control (CC) engine
- Linux kernel support
  - XCP Router function implemented as a kernel loadable module.
  - Linux network device layer includes generic packet queueing and scheduling mechanism *Qdisc.*
  - Linux *tc* command used to manage XCP Qdisc.

XCP Router

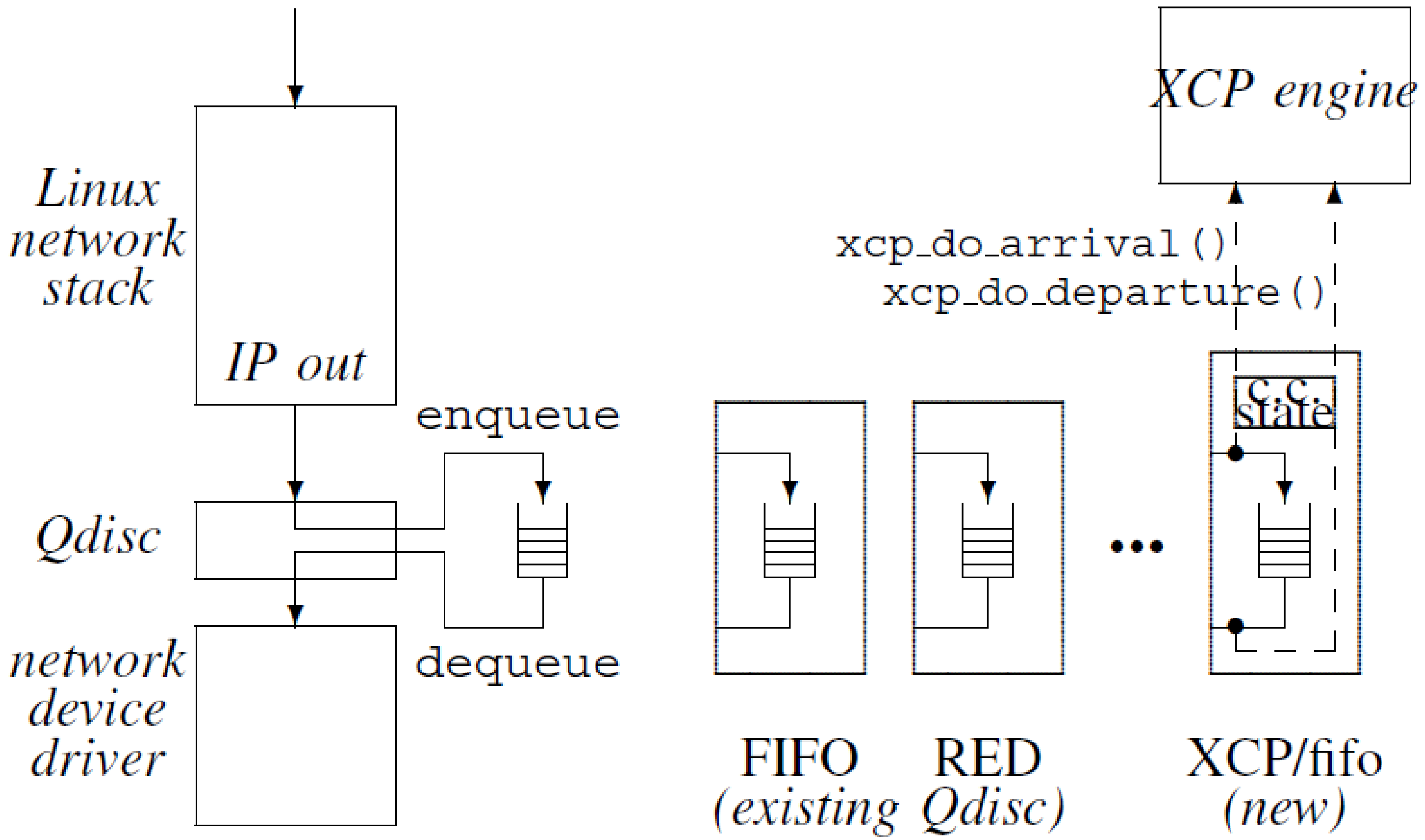

Fig. 2. Structure of XCP router module w.r.t. networking stack in Linux kernel

XCP CC Engine

- No per-flow state in XCP router.
- Per-link congestion control info stored at c.c. state in XCP Qdisc.
- xcp_do_arrival updates running traffic statistics.
- xcp_do_departure calculates individual packet feedback.
- XCP router engine must do per interval (d) processing.

XCP Router

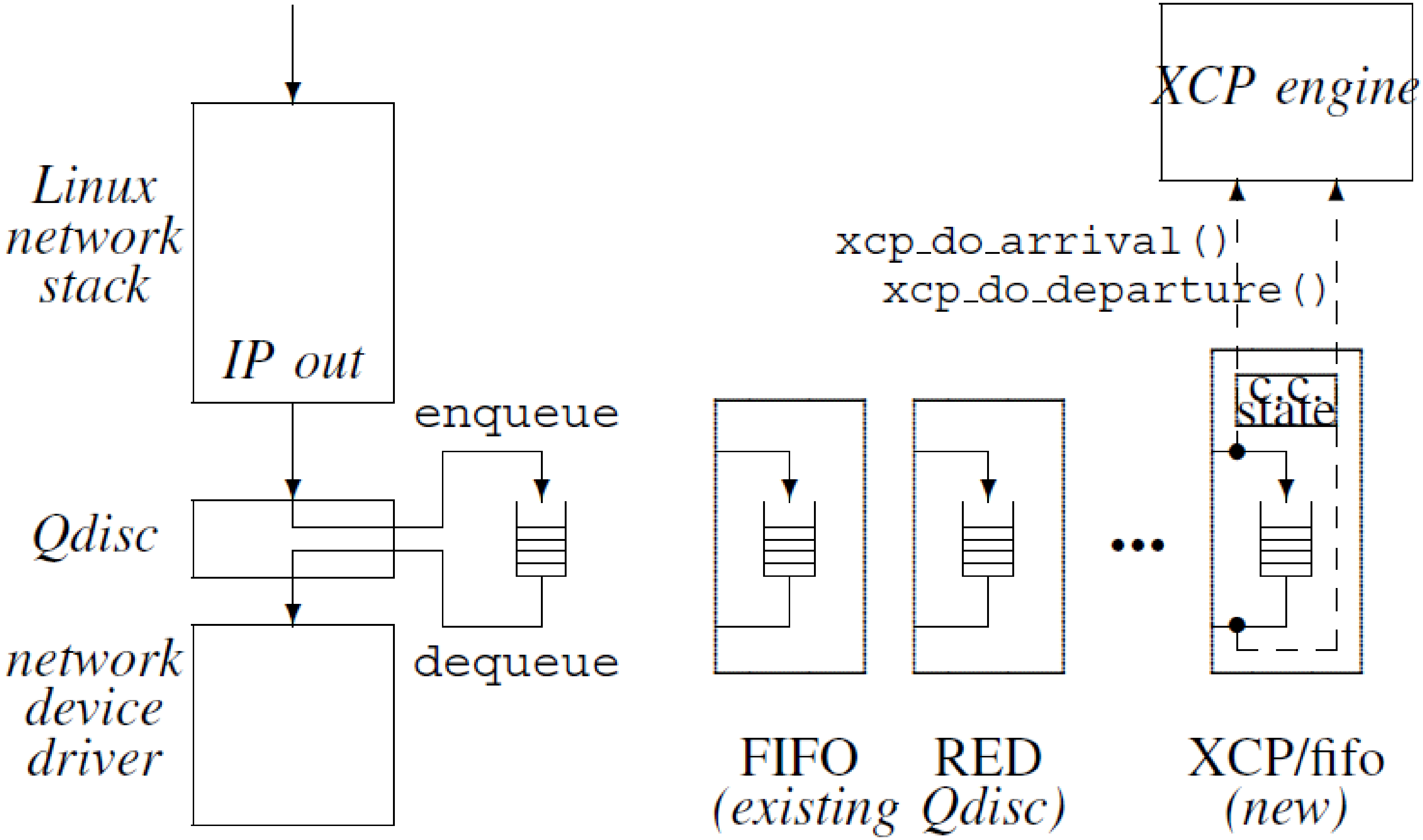

Fig. 2. Structure of XCP router module w.r.t. networking stack in Linux kernel

XCP Implementation Challenges

- Lack of Linux support for double long and floating point in kernel forced careful scaling analysis of variables and XCP parameters to support high-speed networks.
- Changes in bandwidth (capacity) can be fractional ➔ individual packet feedback will be lost to rounding if H_feedback represented as integer number of segments.

Using Mantissa-Exponent Format

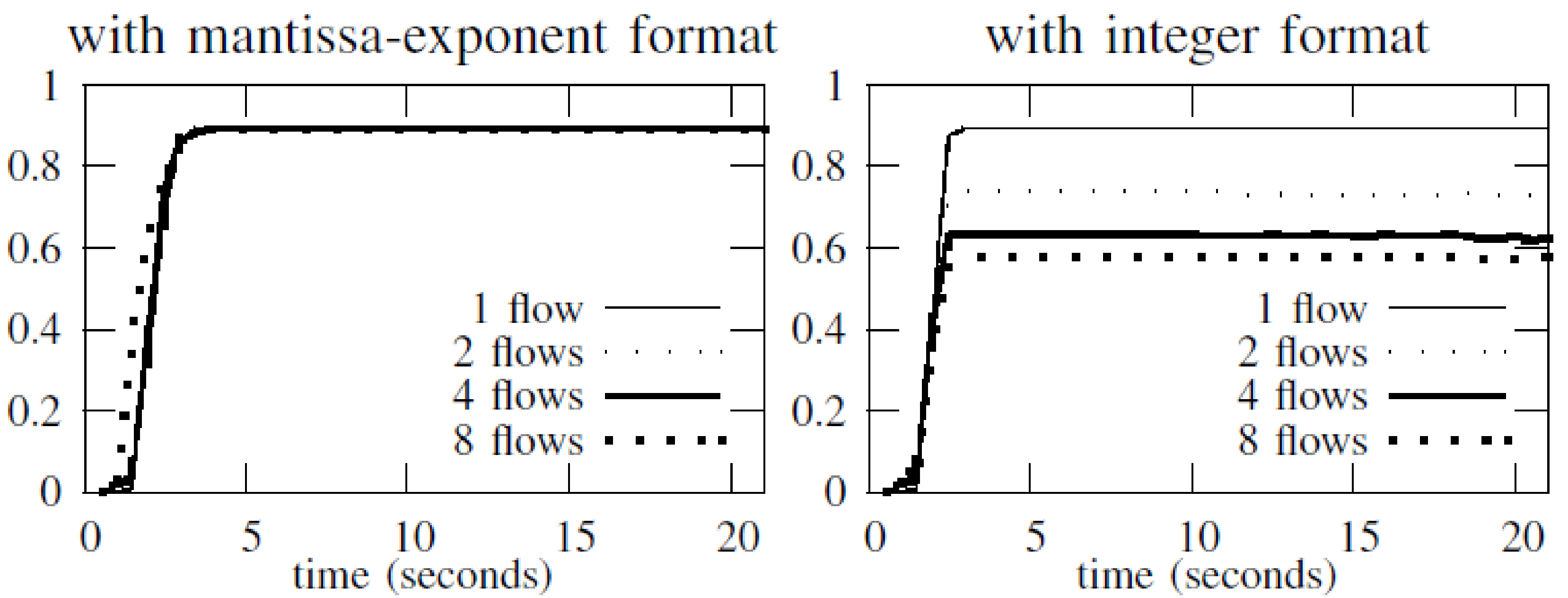

Fig. 3. Bandwidth utilization (total goodput of all flows as a ratio of raw bandwidth) comparison when 1, 2, 4, or 8 flows share the same bottleneck.

Implementation details can strongly impact performance!!

XCP Experimental Network

RTT = 500 ms; router queue size = 2*BDP

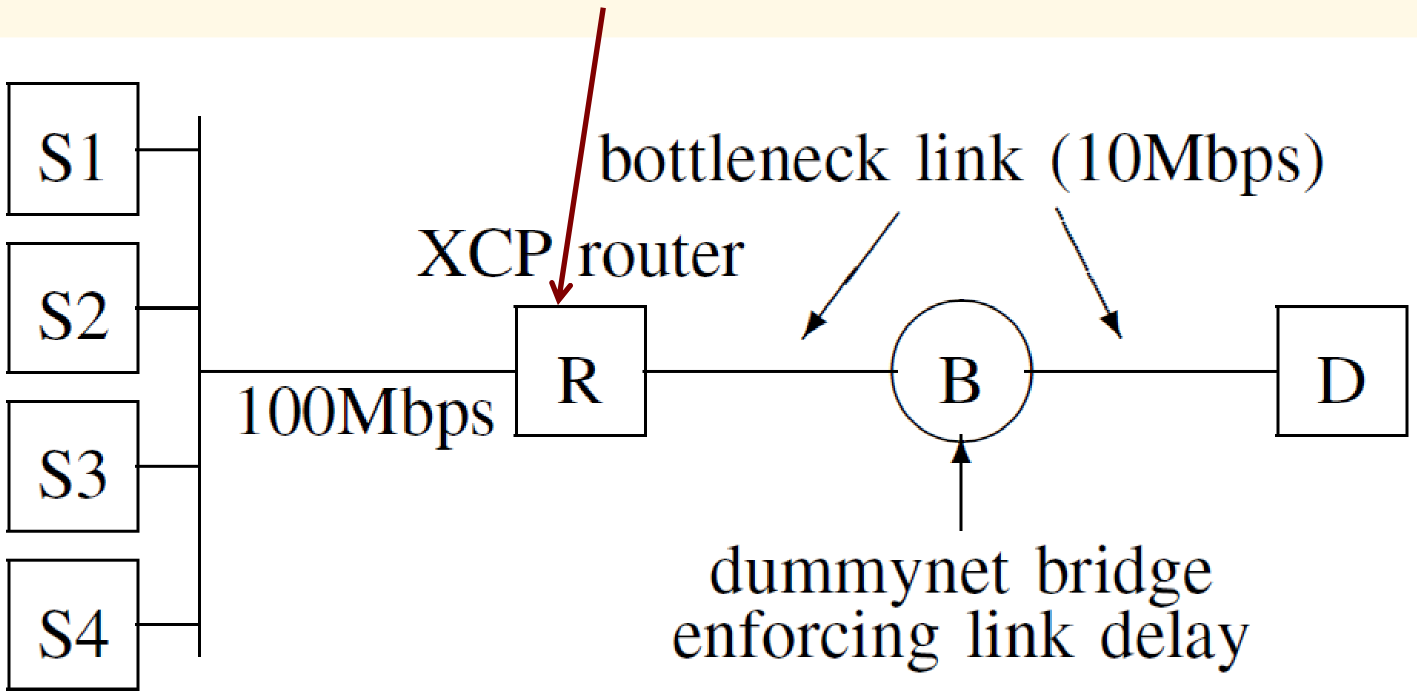

Fig. 4. Experimental network configuration

Figure 5 (top three graphs)

Validation Experiments

Run four flows of one type.
Each flow sent from S_i.
All flows started at the same time.

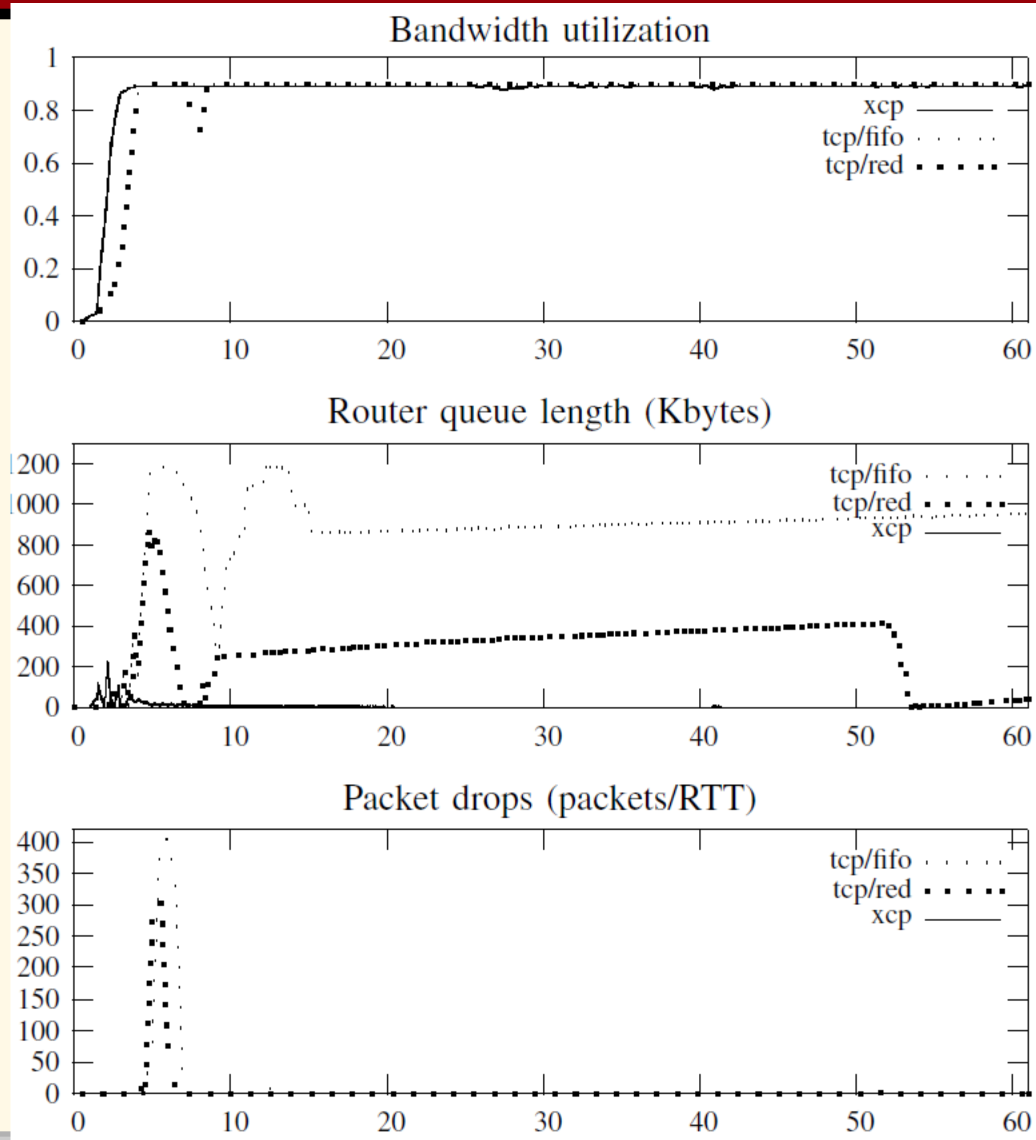

Figure 5 (bottom two graphs)

XCP flows share capacity equally and fairly.

TCP flows have significant fluctuations.

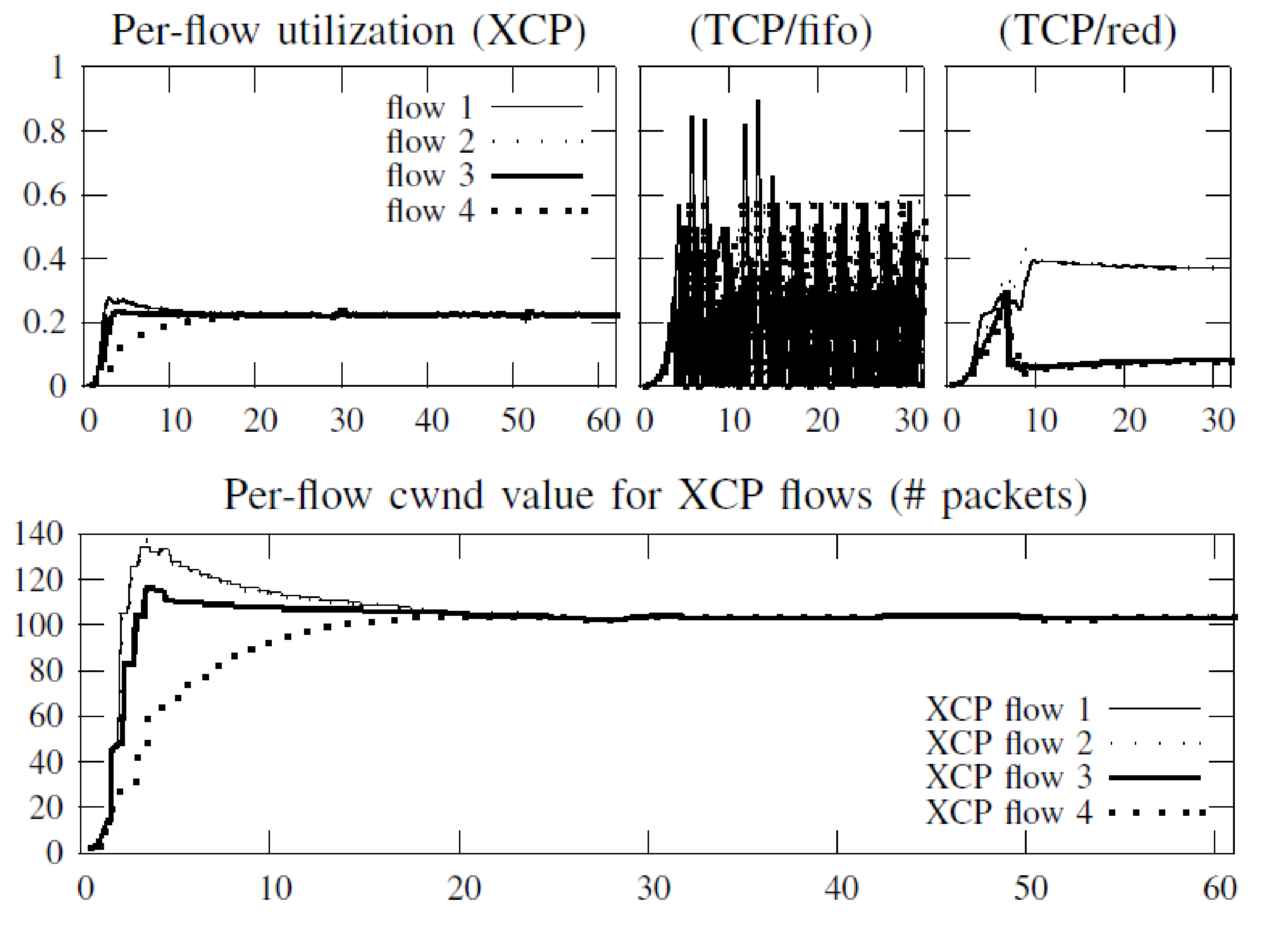

Fig. 5. Performance comparison between XCP and TCP.

Figure 6: XCP vs TCP Per-Flow Fairness

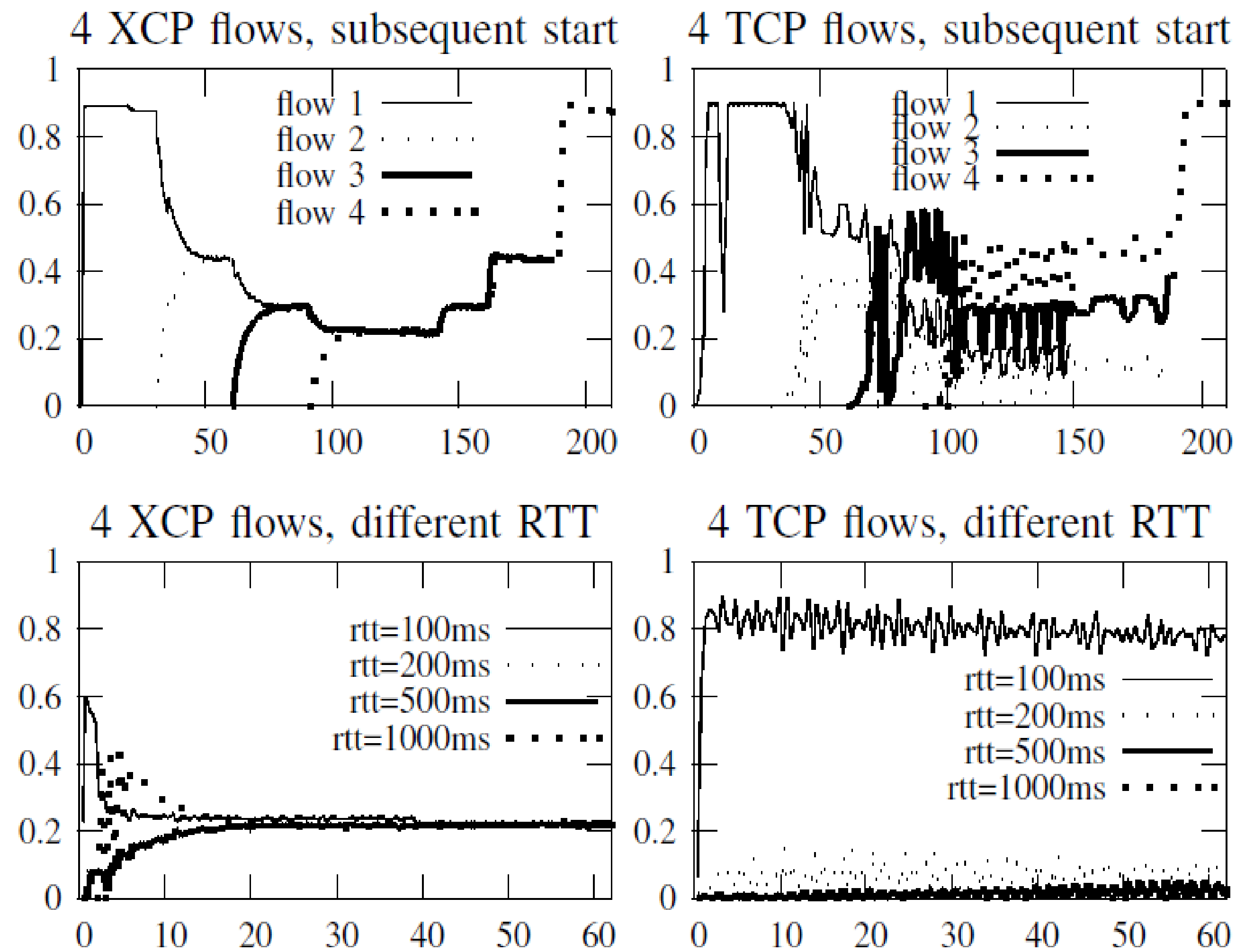

Fig. 6. Per-flow utilizations showing fairness between XCP flows (but not between TCP flows) with different start times or different RTTs.

XCP Sensitivity Study

- XCP sensitive to 'environmental conditions', i.e., OS configuration and network conditions.
- When receiver buffer too small, difference between actual sent and advertised XCP cwnd treated as spare capacity.
- XCP weakness is determining available capacity.
- XCP has deployment problems with non-XCP queues.

Figure 9

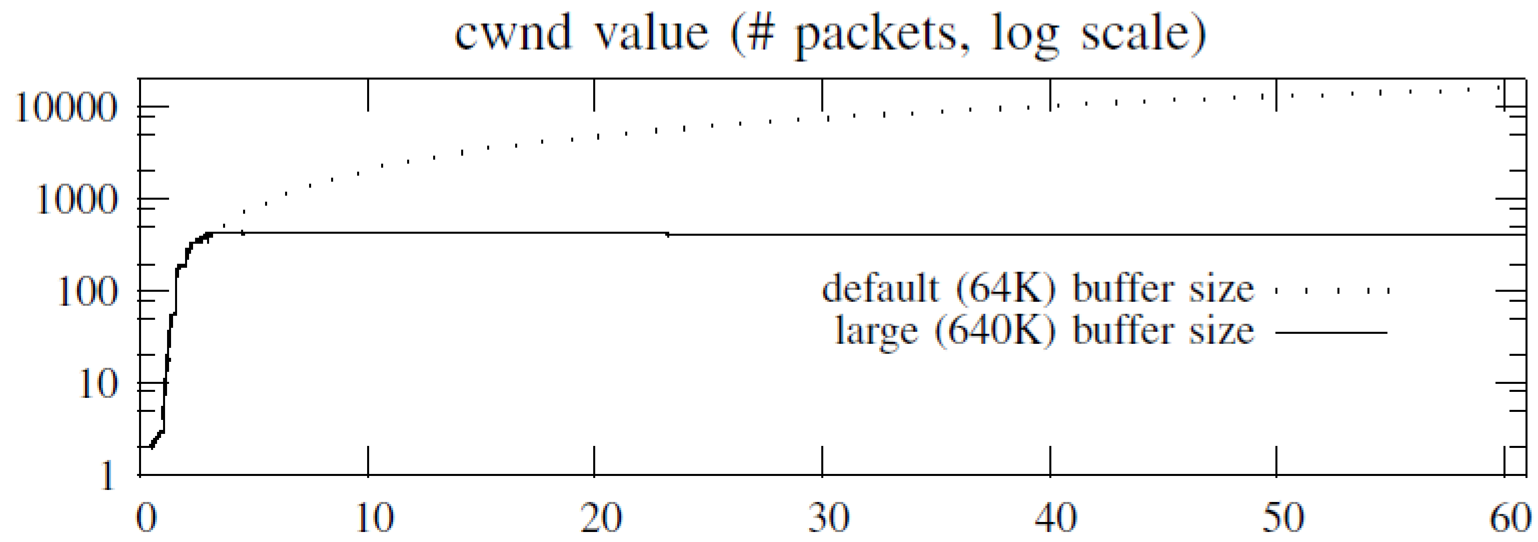

Fig. 9. XCP flow cwnd convergence under different parameter tuning.

The Efficiency Controller (EC)

$$\Phi = \alpha * d * S - \beta * Q$$

- α: 0.4 based on stability analysis
- d: average RTT (feedback delay)
- S: spare capacity (input traffic rate – link capacity)
- β: 0.226 based on stability analysis
- Q: persistent queue size

Link capacity estimates affected by low layer overhead (see Figure 7), crossing traffic and media access contending traffic (see Figure 10) and non-congestion wireless packet losses.

Wireless losses are induced 'pre' and 'post' XCP router and along reverse 'ack' path.

Emulating WLAN Packet Loss

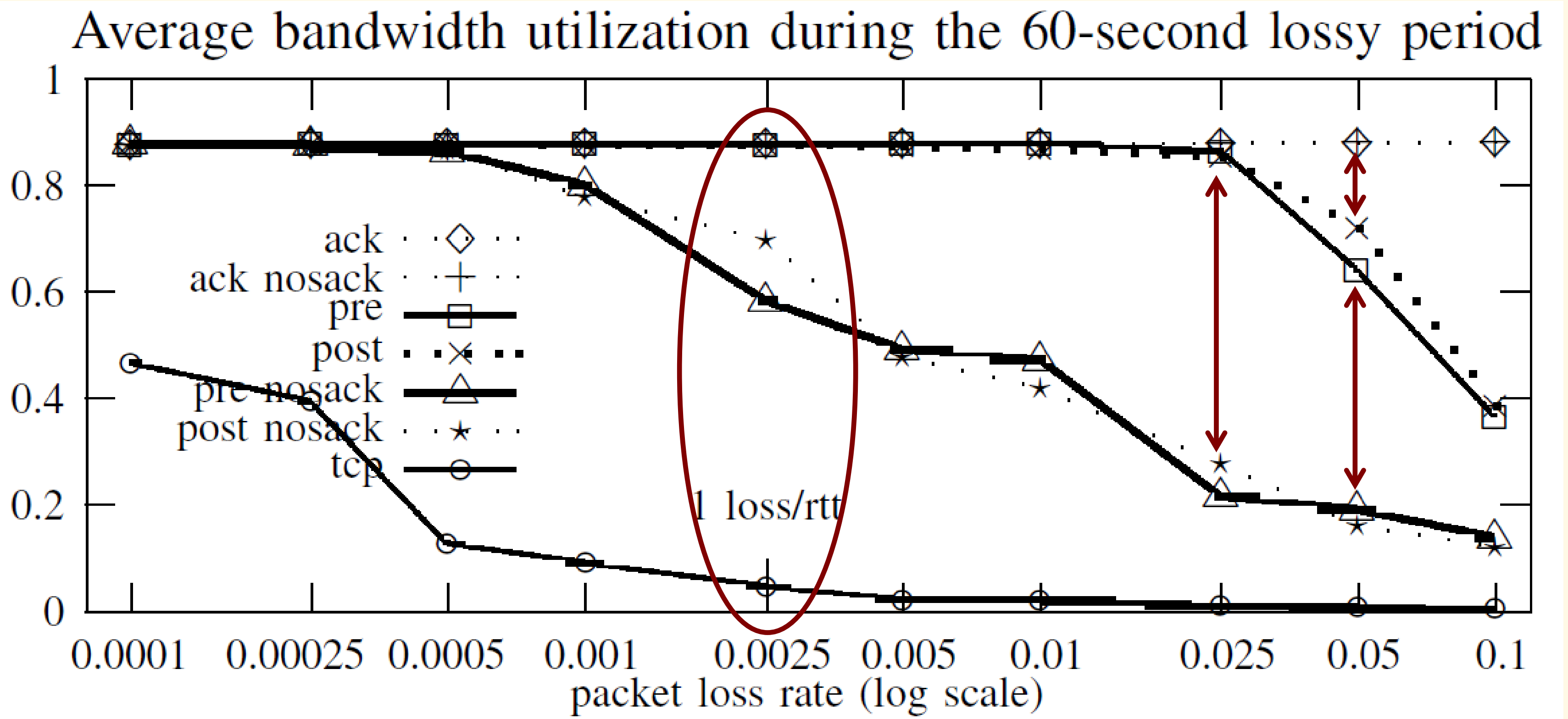

Fig. 11. XCP performance under non-congestion loss

XCP Deployment

- If the bottleneck router is non-XCP (e.g. DropTail or RED), XCP has no mechanism to react to this congestion.
- Experiments conducted with 'tighter' non-XCP router before (case 2) and after (case 1) XCP router.

Hybrid Network Cases

Non-XCP router has FIFO queue with 100 packet buffer and 5 Mbps capacity.

Since XCP assumes all losses due to link impairments, XCP does not respond to FIFO packet drops.

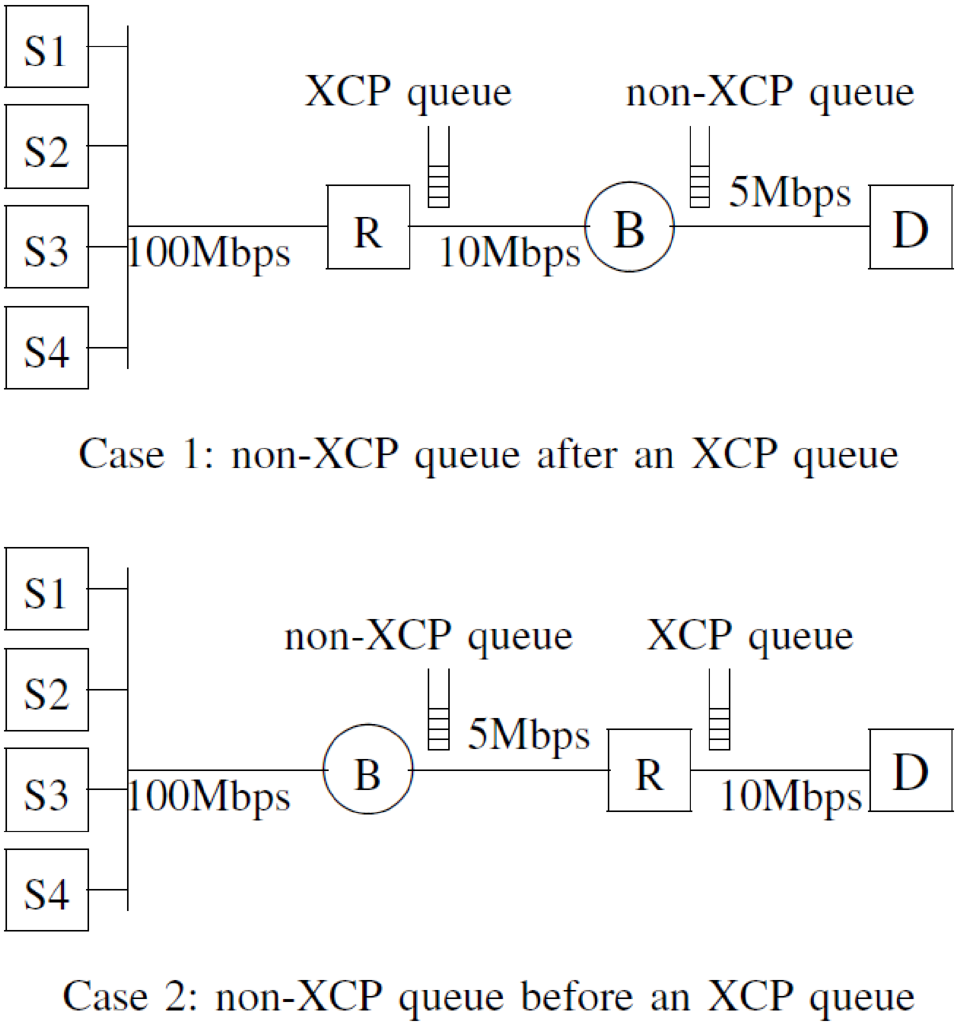

Fig. 12. Network configuration for hybrid network experiments

Case 1 Hybrid Network

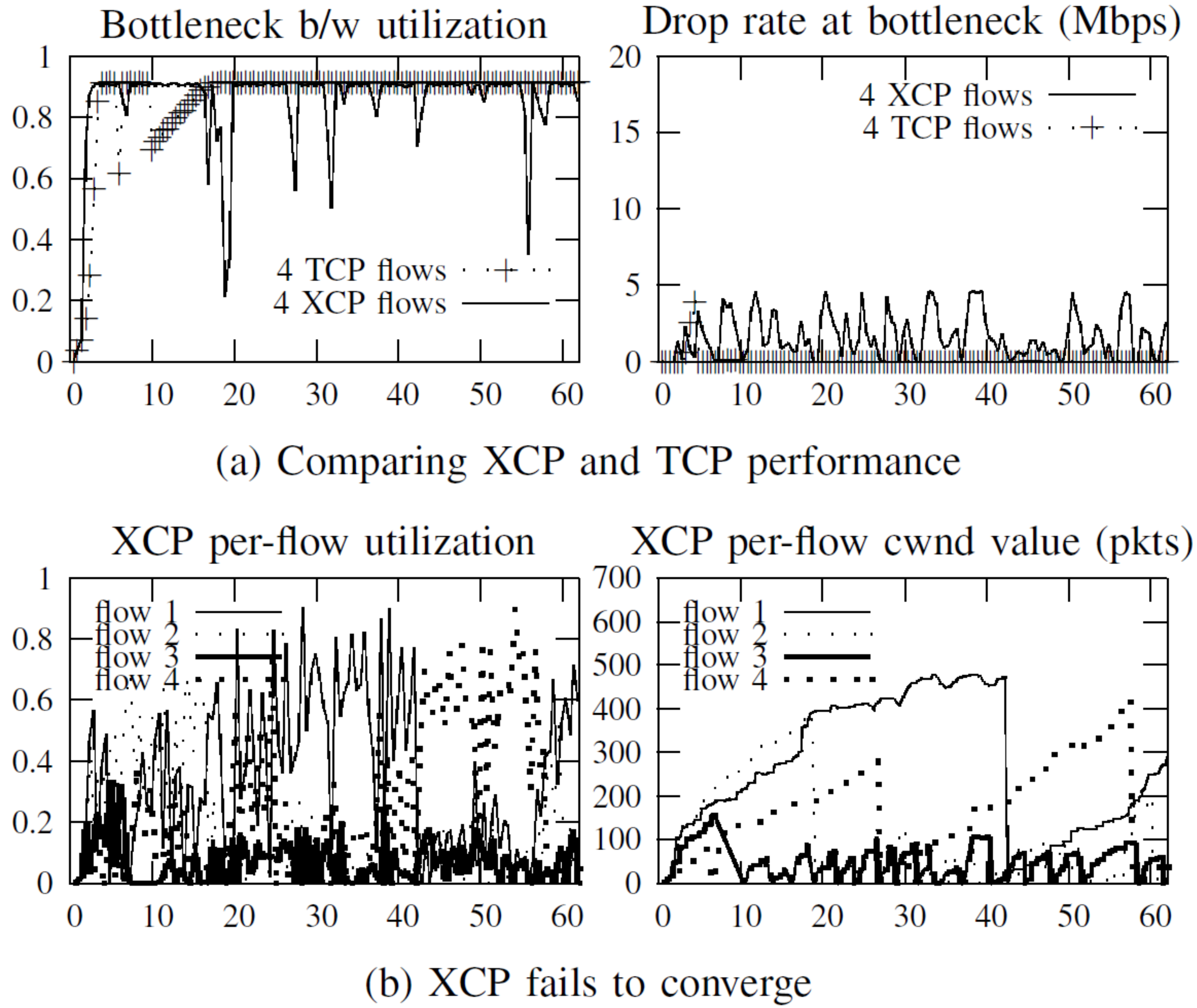

Fig. 13. Results of the non-XCP queue experiment (case 1)

Control Theory Discussion

- Fluid flow model expresses XCP feedback as a set of differential equations:

$$\dot{q}(t) = y(t) - c \qquad (1)$$

$$\dot{y}(t) = -\frac{\alpha}{d}(y(t-d) - c) - \frac{\beta}{d^2}q(t-d) \qquad (2)$$

where α and β are two control parameter constants.

Control Theory Discussion

- Fluid flow model has 'oversight' in meaning of c.
  - c in equation 1 is attainable capacity
  - c in equation 2 is estimated capacity (a configuration parameter).

- Other deficiency is model assumes q(t) has no bound (i.e., assumes no queue limit).

Control Theory Discussion

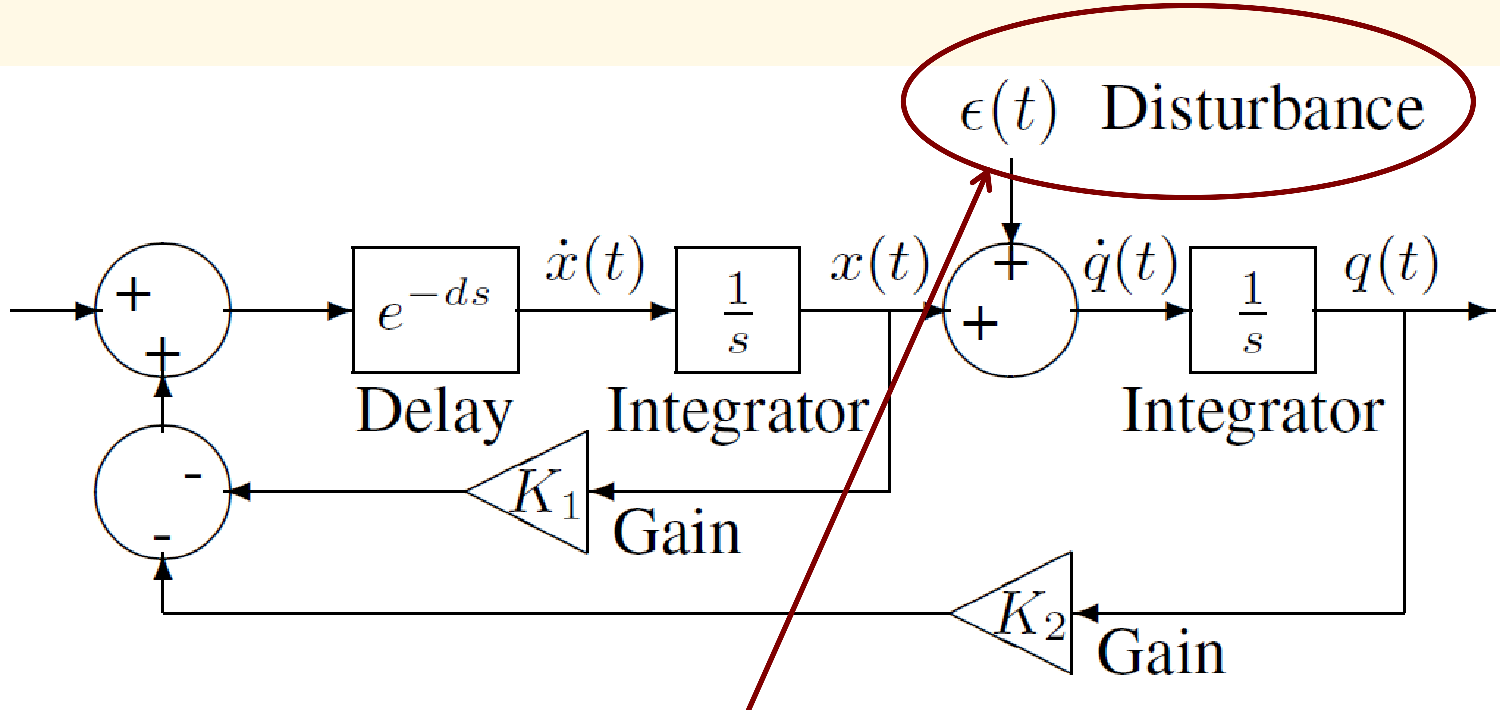

Fig. 15. XCP feedback loop with capacity estimation error

Non-zero disturbance signal causes this linear system to fluctuate and be 'unstable'.

Conclusions and Critique

- This research implements XCP as a TCP option and validates XCP simulation results.
- Implementation challenges were encountered due to lack of support in Linux kernel for precision arithmetic and floating point data types.
- XCP has difficulty when available capacity is not fixed.
- With wireless packet losses, XCP performs better with SACK option.

Conclusions and Critique

- When deployed in mixed networks, FIFO bottleneck routers can cause XCP to perform poorly.
- Paper does not really propose anything new or possible solutions. However, experimental performance studies are often this way.
- The value of the research comes from careful experimental methodology that shows clearly performance strengths and weaknesses.

Thanks!

Questions??