Title: XCP (eXplicit Congestion control Protocol)

Contacts: Dina Katabi (dk@mit.edu)

URLs / RFCs / Papers:

- "Congestion Control for High Bandwidth-Delay Product Networks," Dina Katabi, Mark Handley and Chalrie Rohrs, Proceedings on ACM Sigcomm 2002.
- http://www.ana.lcs.mit.edu/dina/XCP/

Principle / Description of Operation

XCP generalizes the Explicit Congestion Notification (ECN) proposal. Instead of one bit congestion indication used by ECN, it proposes using precise congestion signaling, where the network explicitly tells the sender the state of congestion and how to react to it.

Like TCP, XCP is a window based congestion control protocol intended for best effort traffic. Senders maintain their congestion window (cwnd) and RTT and communicate this to routers via a congestion header (shown in figure 1) in every packet. Sender uses the feedback field in the congestion header to request its desired window increase. Routers monitor the input traffic rates to each of their output queues. Based on the difference between the link bandwidth and its input traffic, router tells the flows sharing that link to increase or decrease their congestion window. It does this by annotating the congestion headers of data packets. Feedback is divided between flows based on their congestion window and RTTs so that the system converges to fairness. A more congested router later in the path can further reduce the feedback in the congestion header by overwriting it. Ultimately the packet will contain the feedback from the bottleneck along the path. When the feedback reaches the receiver, it is returned to the sender in an acknowledgment packet, and the sender updates its cwnd accordingly.

Sender's current cwnd (filled by sender and remains unmodified)
Sender's RTT estimate (filled by sender and remains unmodified)
Feedback (initialized to sender's demands; can be modified by the routers)

Figure 1: Congestion header

Whenever a new acknowledgment arrives, positive feedback increases the sender's cwnd and negative feedback reduces it. An XCP receiver is similar to TCP receiver except when acknowledging a packet it copies the congestion header from the data packet to its acknowledgment.

XCP also introduces the concept of decoupling utilization control from fairness control. A router has both an efficiency controller and fairness controller. The purpose of efficiency controller is to maximize link utilization while minimizing drop rate and persistent queues. It only looks at aggregate traffic and need not care about fairness issues. It computes aggregate feedback at every interval (average RTT of all the flows sharing the link). The aggregate feedback is proportional to both spare bandwidth and persistent queue size. How exactly this aggregate feedback is divided among the packets is the job of the fairness controller. The fairness controller uses the same principle TCP uses (AIMD) to converge to fairness. If the aggregate feedback is positive, allocate it so that the increase in throughput of all flows is the same and if it is negative, allocate it so that the decrease in throughput of a flow is proportional to its current throughput.

Supported operation mode:
Memory to memory (general transport)

Authentication: No

Implementations / API:
XCP implementation in the NS simulator is available at
http://www.ana.lcs.mit.edu/dina/XCP/

Congestion Control Algorithms:
XCP is a congestion control algorithm.

Fairness:
Demonstrates a fairness mechanism and shows how to use it to implement both min-max fairness and the shadow prices model.

TCP Friendly:
Describes a mechanism that allows XCP to compete fairly with TCP but it involves additional work in the routers. Simulation results have been used to demonstrate TCP friendliness of the proposed mechanism.

Predictable Performance Model:
Theoretical analysis on the stability of the protocol and its convergence to fairness can be found in the paper. It is shown to be stable for any link capacity, feedback delay or number of sources.

Results:
The simulations were conducted using the packet level simulator ns-2. The simulations cover link capacities in the range 1.5 Mbps to 4 Gbps, RTTs between 10 ms to 3 sec and

number of sources in the range between 1 and 1000. Further, they simulate 2 way traffic and dynamic environments with arrivals and departures of short web like flows. Most of their simulations use the topology in figure 2. Simulations over the topology in figure 3 are used to show that their results generalize to large and more complex topologies.

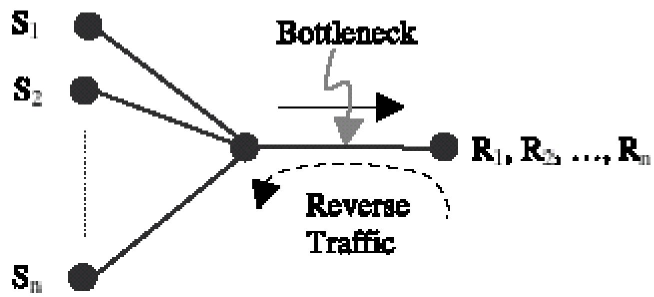

Figure 2: Single bottleneck topology

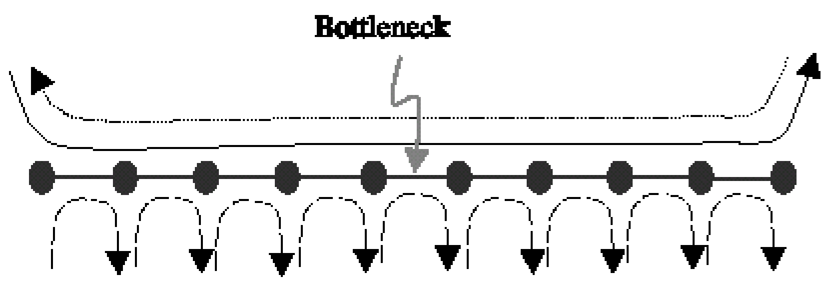

Figure 3: Parking lot topology

They compare XCP with TCP Reno over various Active Queue Management schemes such as RED (Random Early Detection), REM (Random Exponential Marking), AVQ (Adaptive Virtual Queue) and CSFQ (Core Stateless Fair Queuing). The results show that XCP significantly outperforms TCP (with all queuing schemes) in high bandwidth environments as well as in high delay environments. They also show that XCP is efficient in environments with arrivals and departures of short web-like flows. In an environment where the RTTs of the flows that share the bottleneck link vary widely from one another, XCP provides a significantly fairer bandwidth allocation than TCP. They also show how XCP can be used to provide differentiated services to the users based on the price they pay and how XCP can be deployed and how it can gracefully co-exist with TCP.

Target Usage Scenario:
Though XCP is intended to solve TCP's limitation in high-bandwidth large-delay environments, simulation results show that it performs well in conventional environments too.