An Example of Instability in XCP

Lachlan L. H. Andrew, Bartek P. Wydrowski, Steven H. Low

***Abstract*— We present a simple network in which XCP is locally stable but globally unstable in the presence of latency. In this network, the ratio of maximum round trip time (RTT) to mean RTT is large. It can be stabilised by setting the control gain inversely proportional to the maximum RTT, rather than the mean RTT as in the original XCP.**

I. Introduction

TCP congestion control [1] has prevented severe congestion while the Internet underwent explosive growth during the last decade. However, the algorithm has shown serious difficulties as the network continues to scale in size and capacity [2], [3]. This has motivated several recent enhancements [4–9]. (See [6] for extensive references.) Of these, XCP [9] has received much attention for grid computing networks, where its need for explicit communication between the traffic sources and the network is less of a deployment barrier than in the current Internet. Unlike most proposals, which set the flow rates according to the *sum* of congestion measures at the links of their paths, XCP sets them according to the *minimum* “available capacity” in their paths. This has the same flavor as MaxNet [10–12] which sets flow rates according to the *maximum* of congestion measures in their paths. XCP has been shown [9] to be stable when all round trip times (RTTs) are equal. Zhang and Henderson [13] have independently implemented and tested XCP, and highlighted several deployment challenges, such as the sensitivity of XCP to rounding errors in it calculations.

In this paper, we present an example network with distinct RTTs, in which XCP seems to be stable locally, but unstable globally. We also discuss a way to stabilize this example.

II. Instability: an example

Consider XCP running over the network in Figure 1 with $N = 10$ flows sharing a bottleneck link bandwidth of 1 Gbps. Flow 1 traverses both links and has an RTT of 220 ms, while the remaining nine flows traverse only link L1, and have an RTT of 20 ms. Link L2 has a bandwidth of 2 Gbps, and is not a bottleneck. This network is simulated in ns2 using the default parameters of XCP in ns2.

A. Nonlinear instability

We start the network in a state far from equilibrium, namely the state in which all flows start with their default initial window sizes. The throughput of flow 1 is shown in Figure 2, and is highly erratic. This experiment shows that XCP is not globally stable in the presence of latency.

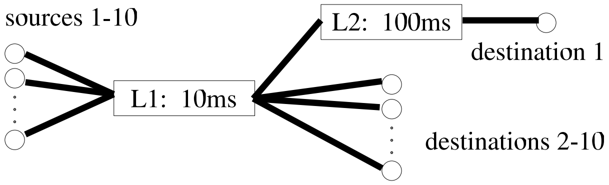

Fig. 1. Topology causing instability.

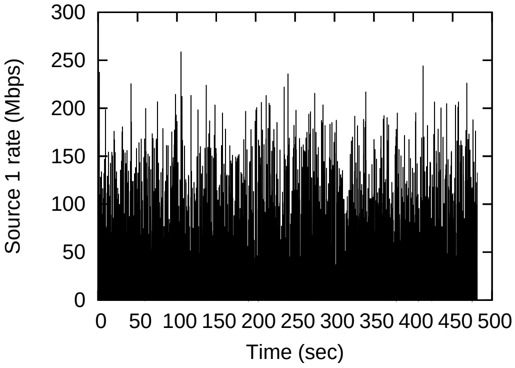

Fig. 2. Throughput of flow 1 with standard XCP for a flow with an RTT much longer than average.

B. Restoring stability

XCP’s instability is due largely to the fact that the loop gain is too high. The loop gain is determined by a parameter called the update interval, d, and the frequency which which control variables are updated. Standard XCP sets d to be the mean RTT observed in the previous update interval.

For this example network, the instability observed can be avoided by setting d to be the *maximum* RTT observed during the estimation interval, instead of the mean RTT. Figure 3 shows the rate of flow 1 with this definition of d. Although it takes many seconds, it does eventually converge, unlike standard XCP.

C. Local stability

Figure 2 demonstrates that XCP is not globally stable. To investigate its local stability, it was simulated starting near its equilibrium state. This was achieved by running the stabilised version of XCP (with d set to the maximum RTT) until the system settled partially (near equilibrium), and then continuing with the standard XCP (setting d to the mean RTT).

From Figure 3, the stabilized version of XCP converges to equilibrium at around 50 s. We started the ns2 simulation using the stabilized version of XCP until time 10 s, and then switched over to standard XCP. Figure 4 shows the resulting rate of flow 1. Clearly, the network had not converged yet at time 10 s, but XCP was able to converge almost to the

L. Andrew is with the ARC Special Research Centre for Ultra-Broadband Information Networks, University of Melbourne, an affiliated programme of the National ICT Australia. B. Wydrowski and S. H. Low are with the CS and EE departments, California Institute of Technology.

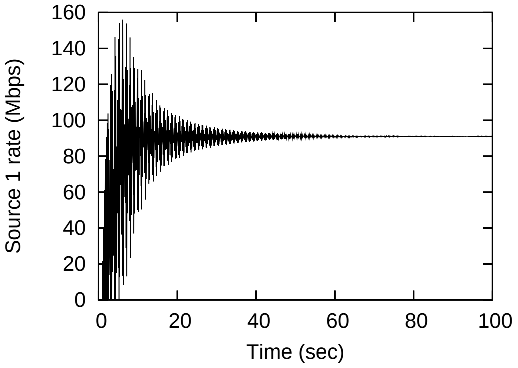

Fig. 3. Throughput when setting d to maximum RTT.

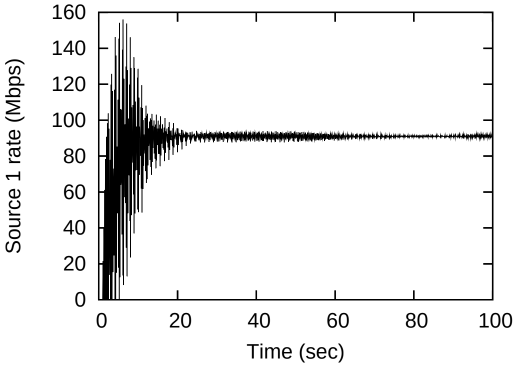

Fig. 4. Throughput when setting d to maximum RTT until 10 s, and using standard XCP thereafter.

equilibrium. This demonstrates that the instability in Figure 2 is due to the nonlinearity encountered when the rate is far from its equilibrium value. The local convergence is faster than that of the stabilised XCP, due to the higher control gain: the convergence time in Figure 3 is roughly 50 s and that in Figure 4 roughly 25 s.

However, standard XCP does permit some random fluctuations around the equilibrium to build up. This can be seen in Figure 5, in which the stabilised XCP is run for 100 s before standard XCP takes over. The stabilised form of XCP brings the system very close to equilibrium, but under standard XCP it then drifts away to a state with small but sustained fluctuations. These are probably due to the discrete increments of the window size, another non-linear effect.

III. Discussion

XCP achieves stability by reducing the amount by which it adjusts flows' rates when the RTT are long. It is well known [14] that a network will be unstable if the product of the feedback delay (RTT) and the control loop gain becomes too large. XCP reduces its control gain by a factor of d; if all flows have an RTT of d then this gives the correct scaling behavior.

However, when the RTTs are significantly disparate, it is possible for a flow to have an RTT much larger than the average RTT, d. For example, if a link is shared by $N-1$ flows with RTT 1 [unit], and one flow with RTT $N+1$ [units], then the average RTT is $d = 2$ [units]. In Figure 1 with $N = 10$ the ratio of peak to mean RTT is 5.5.

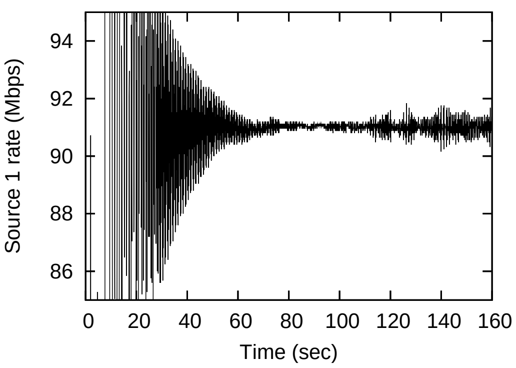

Fig. 5. Throughput when setting d to maximum RTT until 60 s, and using standard XCP thereafter.

Simulation results were presented in Figures 8 and 16 of [9] for networks with flows having RTTs differing by an order of magnitude. However, instability was not observed in those cases. That may be because the ratio of maximum to mean RTT was not sufficiently high: in each case, the ratio was less than 2, compared with a ratio of 5 in our examples.

To understand the source of instability, consider the following dynamic model of XCP (neglecting feedback delay) developed in [15]. The model is for a network with L links shared by N flows. Let R be the $L \times N$ routing matrix: $R_{li} = 1$ if flow i uses link l and 0 otherwise. Let $L(i) := \{l | R_{li} = 1\}$ be the set of links in the path of flow i. For each flow i, define the following variables:

- $w_i(t)$: window size at time t, in packets.
- τ_i: round-trip propagation (and fixed processing) delay.
- $T_i(t)$: round-trip time (RTT) at time t.
- $x_i(t)$: flow rate at time t.

For each link l, define the following variables:

- c_l: capacity, in packets/sec.
- $b_l(t)$: backlog at time t, in packets.
- $y_l(t) := \sum_i R_{li} x_i(t)$: aggregate input rate at link l at time t.

Then

$$\dot{w}_i(t) = \frac{w_i(t)}{d^2} \min_{l \in L(i)} F_{li}(t) \tag{1a}$$

$$\dot{b}_l(t) = \begin{cases} y_l(t) - c_l & \text{if } b_l(t) > 0 \\ \max(y_l(t) - c_l, 0) & \text{if } b_l(t) = 0 \end{cases} \tag{1b}$$

where

$$F_{li}(t) = \frac{h_l(t) + \phi_l^+(t)}{N_l x_i(t)} - \frac{h_l(t) + \phi_l^-(t)}{y_l(t)} \tag{2a}$$

$$\phi_l(t) = \alpha d(c_l - y_l(t)) - \beta b_l(t) \tag{2b}$$

$$h_l(t) = \max(\gamma d y_l(t) - |\phi_l(t)|, 0) \tag{2c}$$

$$x_i(t) = \frac{w_i(t)}{T_i(t)} \tag{2d}$$

$$T_i(t) = \tau_i + \sum_l R_{li} \frac{b_l(t)}{c_l} \tag{2e}$$

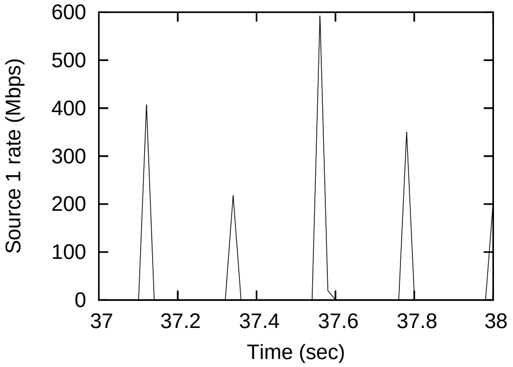

Fig. 6. Magnified view of throughputs with standard XCP.

Here, $\alpha = 0.4$, $\beta = 0.226$ and $\gamma = 0.1$, and $\phi_l^+(t) = \max(\phi_l(t), 0)$, $\phi_l^-(t) = \max(-\phi_l(t), 0)$.

The cause of the erratic behavior in Figure 2 can be understood by looking in more detail at the behavior of this flow. The enlarged view of the rate of flow 1 in Figure 6 shows that the transmission rate is highly peaked.

The source of this peakedness is the rapid changes of window size. The transmit rate is the rate at which the right hand side of the transmit window advances, which consists of the rate at which acknowledgements are received plus the rate at which the window size changes. Whilst (2d) gives the average transmit rate over one round trip time, the instantaneous transmit rate may differ significantly if the window changes significantly during the round trip time.

While the instantenous rate, $x_i(t)$, is positive, it can be expressed as

$$x_i(t) = x_i(t - T_i(t)) + \dot{w}_i(t). \tag{3}$$

If the right hand side drops below 0, the window size drops below the number of packets outstanding. Thus $x_i(t)$ becomes zero and stays zero until the number of packets outstanding drops below the window size, which occurs some time after the right hand side again becomes positive. The periods of zero transmission rate observed in Figure 6 correspond not to $w_i(t) = 0$ as suggested by (2d), but to times when the window size has dropped below the number of packets already in the network.

This peaky behavior is self-sustaining. When flow 1 observes a low rate, the $Nx_i(t)$ in the denominator of (2a) causes any increase in window size to be highly magnified. This results in a large peak in the queue size, causing (2a) to subsequently become very negative. This can be avoided by making XCP respond less rapidly to periods of high or low throughput, such as by responding to the maximum RTT, rather than the mean.

Although simply setting d to the maximum RTT may stabilize XCP, it makes the system sensitive to “outliers”, flows which contribute little traffic but report long round trip times. These flows may actually have very long round trip times, overestimate their RTT due to operating system jitter, or maliciously overstate their RTT. If the dynamics of the flow control for the entire network is slowed down to accommodate a few flows, then it will respond sluggishly to transients. This may lead to long periods of underutilisation, alternating with periods of overload causing excessive delay and packet loss. Hence it is desirable for each flow to adjust its loop gain in response to its own RTT estimate. How to do this in XCP, where control is computed at the routers, is an interesting open problem.

IV. Conclusion

We have presented a simple example in which XCP is stable locally but unstable globally, when the maximum round trip time of a flow is much larger than the mean round trip time. This instability is removed by setting the estimation interval to be the maximum observed RTT, rather than the mean RTT. However, that makes the system vulnerable to erroneous RTT advertisements.

V. Acknowledgement

This work is supported by NSF, Caltech, ARO, AFOSR, and Cisco as part of the FAST Project, and supported by the Australian Research Council.

References

[1] V. Jacobson, “Congestion avoidance and control,” *Proceedings of SIGCOMM’88, ACM*, August 1988. An updated version is available via ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z.

[2] C. Hollot, V. Misra, D. Towsley, and W. Gong, “Analysis and design of controllers for AQM routers supporting TCP flows,” *IEEE Transactions on Automatic Control*, vol. 47, no. 6, pp. 945–959, 2002.

[3] S. H. Low, F. Paganini, J. Wang, and J. C. Doyle, “Linear stability of TCP/RED and a scalable control,” *Computer Networks Journal*, vol. 43, no. 5, pp. 633–647, 2003. http://netlab.caltech.edu.

[4] C. Casetti, M. Gerla, S. Mascolo, M. Sansadidi, and R. Wang, “TCP Westwood: end-to-end congestion control for wired/wireless networks,” *Wireless Networks Journal*, vol. 8, pp. 467–479, 2002.

[5] S. Floyd, “HighSpeed TCP for large congestion windows.” Internet draft draft-floyd-tcp-highspeed-02.txt, work in progress, http://www.icir.org/floyd/hstcp.html, February 2003.

[6] C. Jin, D. X. Wei, and S. H. Low, “FAST TCP: motivation, architecture, algorithms, performance,” in *Proceedings of IEEE Infocom*, pp. 2490–2501, March 2004. http://netlab.caltech.edu.

[7] L. Xu, K. Harfoush, and I. Rhee, “Binary increase congestion control for fast long-distance networks,” in *Proc. IEEE Infocom*, pp. 2514–2524, 2004.

[8] T. Kelly, “Scalable TCP: Improving performance in highspeed wide area networks,” *ACM Comp. Commun. Review*, vol. 33, pp. 83–91, Apr. 2003.

[9] D. Katabi, M. Handley, and C. Rohrs, “Congestion control for high-bandwidth delay product networks,” in *Proc. ACM Sigcomm*, Aug. 2002.

[10] B. Wydrowski and M. Zukerman, “MaxNet: A congestion control architecture for maxmin fairness,” *IEEE Communications Letters*, vol. 6, pp. 512–514, November 2002.

[11] B. Wydrowski, L. L. H. Andrew, and M. Zukerman, “MaxNet: A congestion control architecture for scalable networks,” *IEEE Communications Letters*, vol. 7, pp. 511–513, Oct. 2003.

[12] B. Wydrowski, L. L. H. Andrew, and I. M. Y. Mareels, “MaxNet: Faster flow control convergence,” in *Proc. Networking 2004. Springer Lecture Notes in Computer Science, LNCS 3042*, (Greece), pp. 588–599, 2004.

[13] Y. Zhang and T. R. Henderson, “An implementation and experimental study of the explicit control protocol (xcp),” in *Proc. IEEE Infocom*, Mar. 2005.

[14] F. Paganini, J. C. Doyle, and S. H. Low, “Scalable laws for stable network congestion control,” in *Proc. IEEE Conf. Decision Contr. (CDC)*, (Orlando, FL), pp. 185–90, 2001.

[15] S. Low, L. Andrew, and B. Wydrowski, “Understanding XCP: Equilibrium and fairness,” in *Proc. IEEE INFOCOM*, (Miami, FL), 2005.