XCP Performance in the Presence of Malicious Flows

Dina Katabi
Computer Science and Artificial Intelligence Laboratory @ MIT
Cambridge, Massachusetts, 02139
dk@mit.edu

I. Problem & Motivation

This work studies the impact of misbehaving users on XCP's efficiency and fairness. XCP [2] is a recently proposed congestion control protocol. It outperforms TCP both in traditional environments and large bandwidth-delay product networks. XCP uses explicit feedback from the network. It requires the senders to report their current estimates of the throughput[1] and round trip time (RTT) in a special congestion header attached to every packet [1]. The routers use this information to apportion the congestion feedback between the senders.

The XCP Sigcomm paper [2] assumes that most senders are well-behaved and comply with the rules of the protocol. When senders are malicious, the paper proposes edge monitoring or statistical sampling of traffic to detect misbehaving sources and provide flow protection. Although flow monitoring solves the problem of misbehaving senders, in some cases monitoring could be too expensive or simply unavailable.

In this paper, we examine the impact of misbehaving on XCP networks when flow protection is unavailable (i.e., no flow monitoring). We address questions such as: What are the different ways an XCP source can misbehave? How do they affect efficiency and fairness? How many sources must be involved in the attack before it has noticeable impact on the performance?

II. Methodology

We can classify misbehaving XCP sources as liars and abusers. Liars lie about the information in the congestion header, reporting an incorrect throughput or round trip delay. We study both cases of declaring values larger and smaller than the true one.

In contrast to liars, abusers report their correct throughput and delay, but they do not appropriately react to the feedback they receive. Some abusers may completely ignore the routers' feedback, in which case we call them unresponsive abusers. Less abusive are responsive abusers, which decrease in time of congestion and increase when there is spare bandwidth, yet they increase more and decrease less than they are told by the network. We note that in practice a misbehaving user might be both a liar and an abuser. Our classification separates these issues to help disentangle the effects of various attacks. Table I shows a taxonomy of the studied attacks.

Liars		**Abusers**	
Lie about throughput	Lie about RTT	Limited reaction to congestion feedback	Ignore feedback

TABLE I
Taxonomy of XCP Source Misbehavior

We analyze the impact of the above attacks by closely examining the XCP feedback equations. We also use extensive ns [3] simulations to support our analysis.

III. Summary of Results

We found that lying about one's throughput causes unfairness but does not affect network efficiency, whereas lying about the RTT has little impact on fairness but may degrade the efficiency. In our simulations, the degradation is negligible when the declared RTT is within one order of magnitude of the true RTT, or when the number of liars is small (i.e., less than 50% of the flows). But the degradation becomes significant when most of the flows significantly lie about their RTTs.

As for abusers, we found that when sharing the link with a completely unresponsive flow (i.e., a CBR), the XCP flows adapt to fill up the bandwidth unused by the unresponsive flow. On the other hand, if the abusive flow is partially reacting to the feedback but is too aggressive in grabbing the bandwidth and too lenient in releasing it, then it can obtain more bandwidth than other flows. However, our simulations show that this unfairness is limited because the more the abusive flow sends the larger the negative feedback it receives.

References

[1] D. Katabi. *Decoupling Congestion Control From Bandiwdth Allocation Policy and its Application in High Bandwidth-Delay Product Networks.* PhD thesis, Massachusetts Institute of Technology, 2003.

[2] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high bandwidth-delay product networks. In *Proc. of ACM SIGCOMM '02*, Aug. 2002.

[3] The network simulator ns-2. http://www.isi.edu/nsnam/ns.

[1] In the XCP Sigcomm paper [2], the senders declare their congestion window, cwnd, and round trip time, RTT, in the congestion header, and adjust cwnd according to the received feedback. In contrast, in Katabi's thesis [1], the senders declare their throughput and RTT and adjust their throughput according to the feedback. In principle, there is no difference between the two because the throughput is the congestion window divided by the RTT. Here, we assume the senders declare their throughput and RTT.