Aggregate Congestion Control

Ratul Mahajan, Steven M. Bellovin, Sally Floyd,
John Ioannidis, Vern Paxson, and Scott Shenker*

`http://www.aciri.org/pushback`

There is a vast body of research dealing with congestion control inside the network. However, all of it deals with congestion at the granularity of a flow (for some definition of a flow). When congestion is caused by events like DoS attacks or flash crowds, which contain lots of flows, flow-based congestion control (FCC) is of little or no use. To deal with such events, we propose aggregate-based congestion control (ACC), which works at a different granularity - that of an aggregate. An aggregate is a collection of packets sharing a common property. Examples of an aggregate are all packets with a given source prefix, and all ICMP ECHO packets destined for a particular address.

FCC fundamentally differs from ACC because of the following reasons: a) there is no definition of an aggregate to start with; the aggregate specification would come into existence when congestion occurs and the aggregate responsible for it is found. b) there is no well-defined fairness goals for aggregates, like max-min fairness is for flows.

The primary goal of ACC is to protect the network and the rest of the traffic from severe congestion caused by high-bandwidth aggregates. When the high-bandwidth aggregate is malicious, ACC should also try to protect the innocent traffic within the aggregate because not all traffic going to a server under attack is malicious.

A router implementing ACC monitors its level of congestion. On discovering sustained severe congestion, the router tries to identify the aggregate(s) responsible for it, using either drop history or random samples. Properties like source and destination prefixes are considered while looking for the responsible aggregate. The identified aggregates are rate-limited to a level that is dynamically decided based on the arrival rate of non-rate-limited aggregates, and the congestion level at the router. This is done such that the aggregate is not punished too harshly, while significantly reducing the drop rate at the congested router.

While rate-limiting an aggregate, a class of packets within the aggregate, that is more likely to be the culprit, can be dropped more heavily. For example, given the aggregate specification as packets with destination prefix D, TCP SYNs can be dropped more heavily if they account for a large fraction of packets going to D.

Described above is a scheme that is purely local to the congested router, and hence called local ACC (LACC) It helps in protecting the rest of the traffic from the high drop rates caused by the high-bandwidth aggregate(s). LACC can be supplemented at the routers with a cooperative ACC mechanism called pushback. Using pushback, the congested router can request its upstream routers to rate-limit the aggregate on it behalf. Pushback can be recursively propagated further upstream.

Pushback has two advantages in addition to those of LACC. First, by taking rate-limiting upstream, pushback reduces bandwidth consumption of packets that would eventually be dropped downstream. Second, and more important, pushback can help focus rate-limiting on traffic coming from directions that are more likely to be pumping in malicious traffic. This can be achieved by intelligently computing the rate-limits sent upstream (can be different for different upstream routers), and would protect the innocent traffic in the aggregate specification.

We have implemented pushback in the *ns*[1] network simulator. Initial simulation results[2] are very encouraging. Currently, we are investigating both LACC and pushback in more detail, looking at issues like implementation complexity (a FreeBSD prototype implementation is also in progress), incremental deployment of pushback, policy issues, attack topologies (utility of pushback depends on it), and finer time-scale effects of LACC.

*Ratul Mahajan is from University of Washington (work done while at ACIRI); Steven M. Bellovin and John Ioannidis are from AT&T Labs Research; Sally Floyd, Vern Paxson and Scott Shenker are from ACIRI. The email addresses are ratul@cs.washington.edu, smb@research.att.com, floyd@aciri.org, ji@research.att.com, vern@aciri.org, and shenker@aciri.org

References

[1] *ns* Web Page. `http://www.isi.edu/nsnam`.

[2] Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker. Controlling High Bandwidth Aggregates in the Network. `http://www.aciri.org/pushback`. July 2001.