Controlling high volume aggregates using Pushback:

Joint work w/ Steve Bellovin, Sally Floyd, and Ratul Mahajan.

Core idea: router signals its upstream peers to restrict a given *aggregate* to a given *transmission rate*.

Router detects aggregate overwhelming it by using packet drops as samples of the traffic flowing through it (alá RED).

Aggregate might be coarse (**dst 192.0.0.0/12**) or fine
(**src www.victoriasecret.com and dst 128.32/16**).

Controlling high volume aggregates using Pushback, con't:

Upon receipt of a Pushback request, upstream router
constructs a *pre-queue* to rate-limit that traffic.

If traffic arrives below rate, fine, it sails through.

If traffic arrives above rate, it is dropped down to the rate.
In addition, router samples *that* drop process and *recursively* sends Push-backs upstream to *its* peers.

Pushback potentially propagates all the way to the source;
but at least to a provider's edge, and quite possibly beyond.

Pushback details:

Pushback requests are *topologically validated*: must arrive with TTL=255.

Upstream routers send *reports* to the destination summarizing how many packets they've dropped and any *narrowing* they've done.

Pushback requests are *soft state*. Congestion router refreshes (or changes) request periodically, or allows it to die out.

We're working on an "aggregate detection" algorithm based on fast routing lookups.

Pushback summary:

Pushback is a *general* mechanism for controlling high-bandwidth aggregates, including legit traffic such as “flash crowds.”

Could evolve into a general traffic management technique . . .?

Open question #1:

what are the actual time constants you get in practice?

Open question #2:

how diffuse a DDOS attack can it defend against?

Open question #3:

what sort of traffic mgt. services does it subsume?