ETH zürich

ECN Beyond the Web: Measurements from BitTorrent DHT

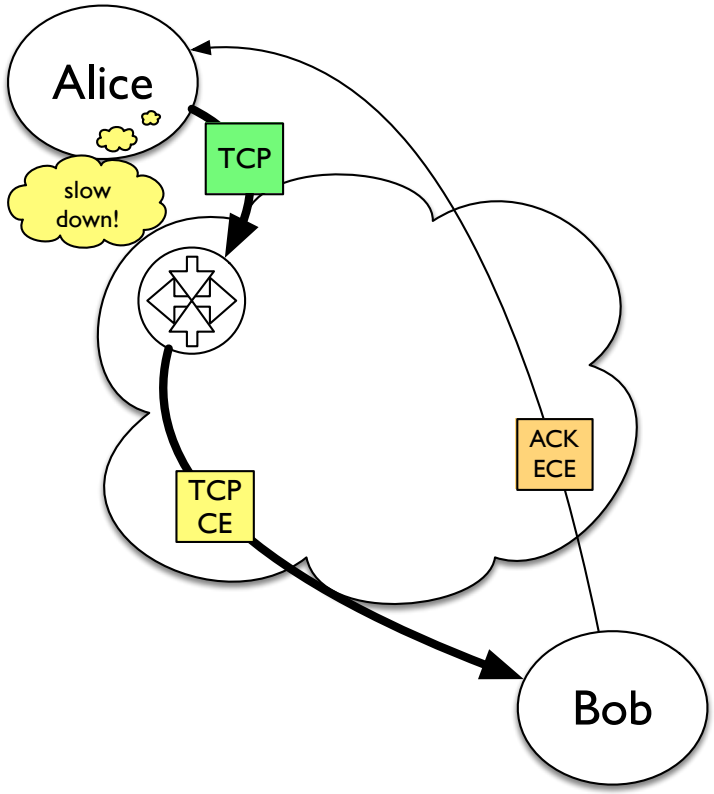
Brian Trammell, Mirja Kühlewind, Elio Gubser

Internet Congestion Control Research Group
IETF 93 Prague, Wednesday 22 July 2015

mPlane

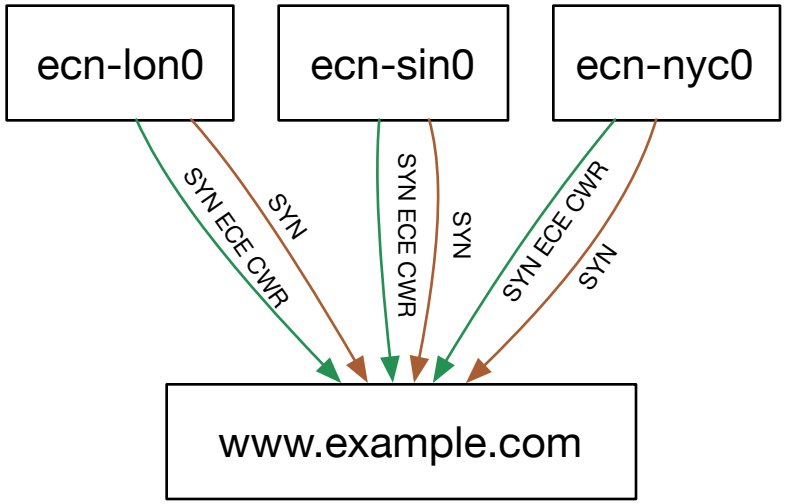
Review: The Problem

- Explicit Congestion Notification (ECN) defined in RFC 3168
  - 15 years ago!
- Idea: routers mark packets to signal congestion
- Deployment largely failed
  - Rebooting routers
  - Broken middleboxes
  - Overprovisioning “fixed” the problem in the meantime
- Changing network environment means that ECN is relevant again



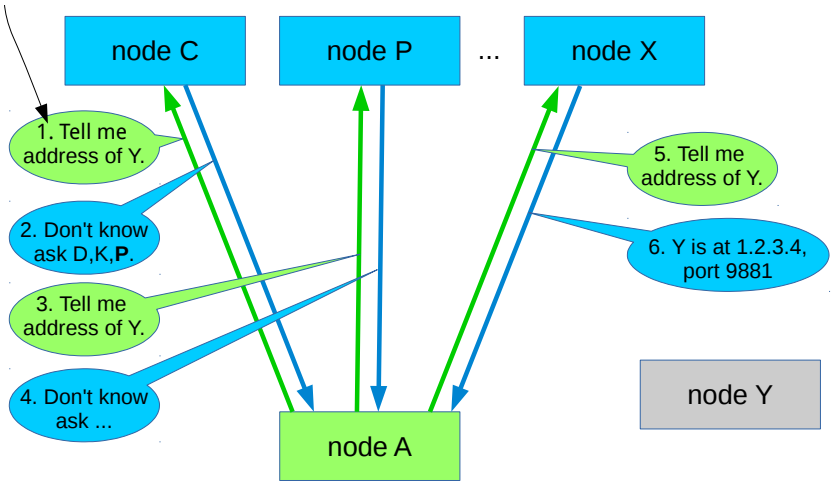
Review: Measuring ECN safety on the web

- Last time, we probed the Alexa top million from three vantage points.
- > 99.5% of the time, ECN leads to no connectivity issue, almost all handled by simple fallback.
- > 65% (mostly Linux) will negotiate.
- Comment: "the web is the most difficult environment"
  - Lots of boxes in front of web servers that "add value".
  - "Go look at something else."



Harvesting BitTorrent Endpoints from the DHT

- Problem: We need lots of non-web-servers to try and connect to, with reasonably legitimate traffic.
  - SMTP? "Spammer!"
- BitTorrent uses a distributed hash table to point to content.
- Walking the DHT with *random* torrent IDs allows one to very quickly and cheaply get valid {IP,port} tuples of hosts legitimately listening on TCP.
  - (1k endpoints/sec, 400kbps)



Findings: unsurprising.

- 687,089 IPv4 hosts probed
- **0.21%** of them (1441) will connect without ECN but not with (compared with 0.38% of Web servers)
  - Much more transient failure (about 6%): these processes have much more variable lifetimes and connectivity.
- **No significant difference** in connectivity risk between web and (directly-connected) BitTorrent hosts.
- 68% of Linux and 1% of Windows hosts will negotiate.
- Can we safely leverage client-side defaults to drive ECN deployment?
  - ***Still yes.***