Active Queue Management, ECN, and Beyond

Sally Floyd
May 1, 2001

Juniper brown bag lunch

Topics:

- First, the intro about end-to-end congestion control.
- Active Queue Management.
- Explicit Congestion Notification.
- Controlling misbehaving or high-bandwidth flows.
- Controlling congestion from flash crowds or Denial-of-Service attacks.

Why do we need end-to-end congestion control?

- As a tool for the application to better achieve its own goals:
E.g., minimizing loss and delay, maximizing throughput.

- To avoid congestion collapse.
  – Congestion collapse occurs when the network is increasingly busy, but little useful work is getting done.
  – E.g., congested links could be busy sending packets that will be dropped before reaching their destination.
  – Tragedy of the commons is avoided in part because the “players” are not individual users, but vendors of operating systems and other software packages.

- Fairness (in the absence of per-flow scheduling).

TCP congestion control:

- Packet drops as the indications of congestion (so far).

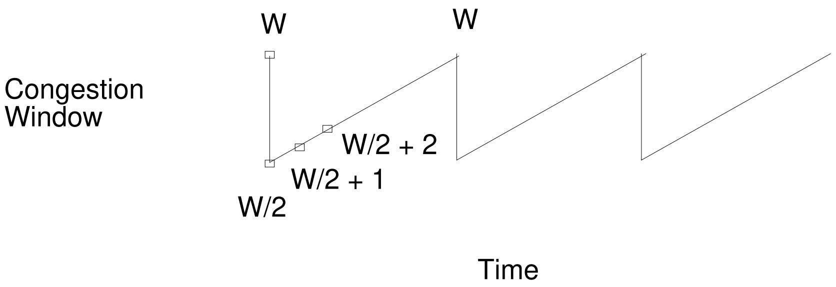
- TCP uses Additive Increase Multiplicative Decrease (AIMD) [Jacobson 1988].
  - Halve congestion window after a loss event.
  - Otherwise, increase congestion window each RTT by one packet.

- In heavy congestion, when a retransmitted packet is itself dropped, use exponential backoff of the retransmit timer.

- Slow-start: start by doubling the congestion window every roundtrip time.

The “steady-state model” of TCP:

- The model: Fixed packet size B in bytes.
  - Fixed roundtrip time R in seconds, no queue.
  - A packet is dropped each time the window reaches W packets.
  - TCP’s congestion window: W, $\frac{W}{2}$, $\frac{W}{2}$ + 1, ..., $W - 1$, W, $\frac{W}{2}$, ...

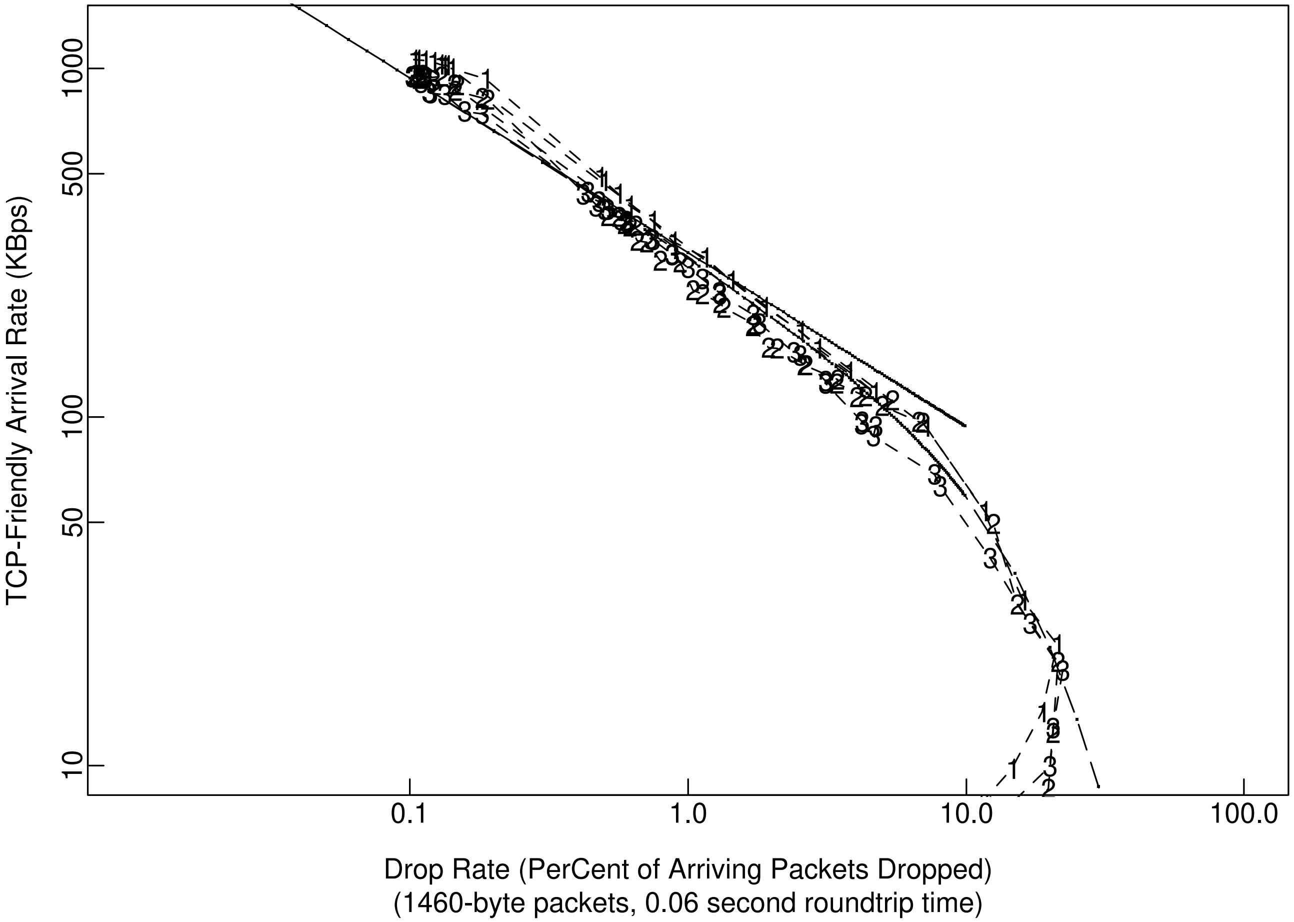


- The maximum sending rate in packets per roundtrip time: W
  - The maximum sending rate in byes per second: WB/R
  - The average sending rate T: $T = (3/4)WB/R$

- The packet drop rate p: $p = \frac{1}{(3/8)W^2}$

- The average sending rate T in bytes/sec: $T = \frac{\sqrt{1.5}B}{R\sqrt{p}}$

Verifying the “steady-state model” of TCP:



Solid line: the simple equation characterizing TCP

Numbered lines: simulation results

Topics:

-
- Active Queue Management.
-
-
-

Goals of Active Queue Management:

- The primary goal: Controlling average queueing delay,
while still maintaining high link utilization.

Secondary goals:

- Improving fairness
(e.g., by reducing biases against bursty low-bandwidth flows).

- Reducing unnecessary packet drops.

- Reducing global synchronization
(i.e., for environments with small-scale statistical multiplexing).

- Accommodating transient congestion
(lasting less than a round-trip time).

Non-goals of Active Queue Management:

- Preventing oscillations in the queue size, or in the average queue size.
- Eliminating buffer overflow.
- Providing max-min fairness between flows, or any other precise control over fairness.

RED queue management, roughly:

```
for each packet arrival
   calculate the new average queue size avg
   if min_th ≤ avg < max_th
      calculate probability p_a
      with probability p_a:
         mark/drop the arriving packet
   else if max_th < avg
      drop the arriving packet
```

Variables:

avg: average queue size

p_a: packet marking/dropping probability

Parameters:

min_{th}: minimum threshold for queue

max_{th}: maximum threshold for queue

The argument for using the *average* queue size in AQM:

- To be robust against transient bursts:

  – When there is a transient burst, to drop just enough packets for end-to-end congestion control to come into play.

  – To avoid biases against bursty low-bandwidth flows.

  – To avoid unnecessary packet drops from the transient burst of a TCP connection slow-starting.

Topics:

-
-
- Explicit Congestion Notification.
-
-

- The old document:

A Proposal to add Explicit Congestion Notification (ECN) to IP, Ramakrishnan, K.K., and Floyd, S., RFC 2481, Experimental, January 1999.

- The new document:

The Addition of Explicit Congestion Notification (ECN) to IP,
draft-ietf-tsvwg-ecn-03.txt
K. K. Ramakrishnan, Sally Floyd, and David Black

This has finished its second IESG Last Call, and should be considered by the IESG on Thursday for Proposed Standard.

The most recent change in the ECN draft:

defining the fourth codepoint in the IP header:

```
+-----+-----+
| ECN FIELD |
+-----+-----+
  ECT   CE         The ECT and CE bits defined in RFC 2481.
   0     0         Not-ECT
   0     1         ECT(1)     * THIS IS THE NEW CODEPOINT *
   1     0         ECT(0)
   1     1         CE

The ECN Field in the IP Header.
ECT: ECN-Capable Transport
CE: Congestion Experienced.
```

The current deployment problem: (broken) web servers that block ECN-capable TCP connections

- The problem is that some Internet hosts are not reachable from an ECN-Capable TCP client.

- For more information:

  – The ECN web page:
http://www.aciri.org/floyd/ecn.html

  – The ECN-under-Linux Unofficial Vendor Support Page:
http://gtf.org/garzik/ecn/

  – The TBIT (TCP Behavior Inference Tool) web page:
http://www.aciri.org/tbit/

Topics:

-
-
-
- Controlling misbehaving or high-bandwidth flows.
-

Questions about congestion in the Internet:

- How often do routers have periods of unusually-high packet drop rates?

- Which routers? (E.g., access routers? last-mile routers? routers for transoceanic links?)

- For periods of high packet drop rates, how often is it due to:
  - A few flows not using end-to-end congestion control?
  - Legitimate flash crowds?
  - DOS attacks?
  - Network problems (e.g., routing failures)?
  - Diffuse general congestion?

Misbehaving or high-bandwidth flows:

- Flow: defined by source/destination IP addresses and port numbers.
  - Example: a single TCP connection.

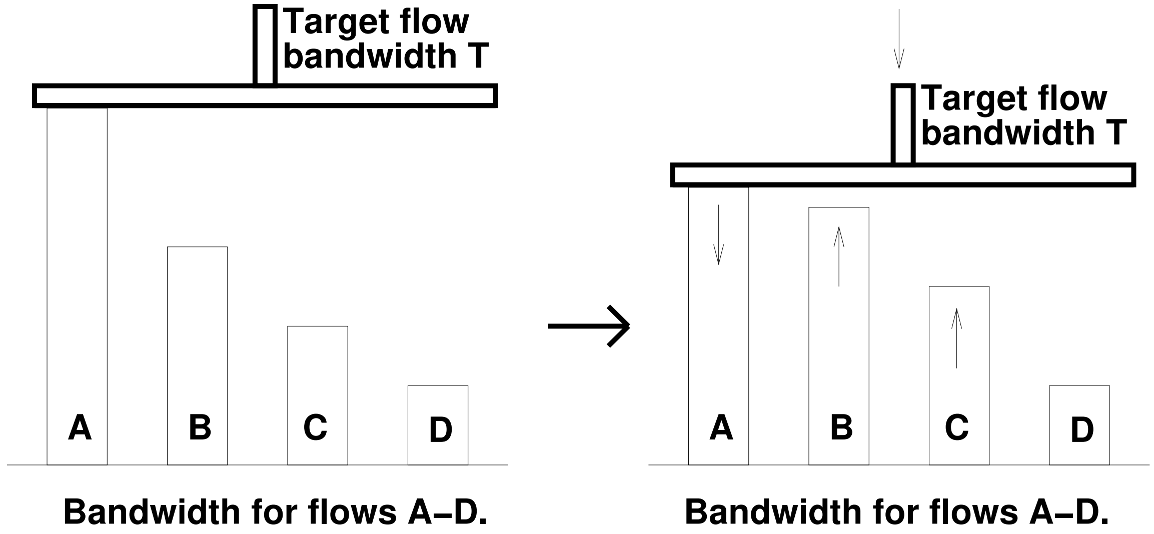
- Problem: Preventing congestion collapse from congested links carrying undelivered packets.

- The answer: Either end-to-end congestion control, or a guarantee that packets that enter the network will be delivered to the receiver.

- The concrete incentive to users: Provide mechanisms in routers that, in times of high congestion, police high-bandwidth flows contributing to that congestion.

Controlling High-Bandwidth Flows at the Congested Router

- Max-min fairness is an acceptable policy for flows.
  - Per-flow scheduling gives max-min fairness.



- Implementation issues:
  - detecting high-bandwidth flows;
  - deciding the bandwidth limit for rate-limiting those flows.

Controlling High-Bandwidth Flows: RED-PD

RED with Preferential Dropping

- Use the packet drop history at the router to detect high-bandwidth flows.

- The target bandwidth in pkts/sec from the TCP throughput equation is $\frac{\sqrt{1.5}}{R\sqrt{p}}$, for:

  R: a configured round-trip time

  p: the current packet drop rate

.

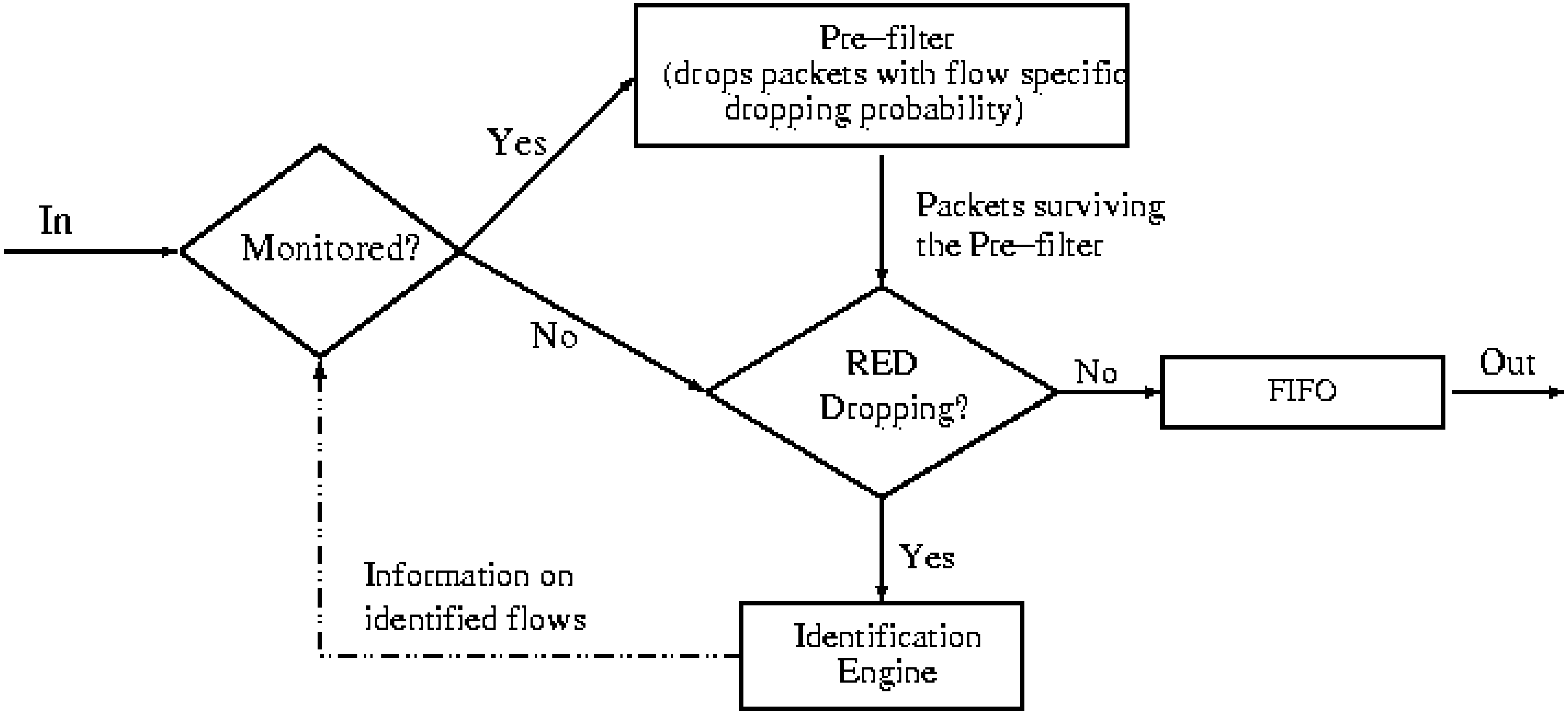
- Monitored flows are rate-limited before the output queue.

- Monitored flows could be misbehaving flows (e.g., not using end-to-end congestion control) or conformant flows with small round-trip times.

- Identifying which monitored flows are *misbehaving* would be a separate step.

– Mahajan and Floyd, Controlling High-Bandwidth Flows at the Congested Router, November, 2000.

Architecture of RED-PD



Topics:

-
-
-
-
- Controlling congestion from flash crowds or Denial-of-Service attacks.

Aggregate-based Congestion Control: Congestion from Flash Crowds

- Example: The Starr Report, September 11, 1998:
"Nothing in recent times has caused a spike quite like that: not the Olympics (Nagano or Atlanta); not the beginning or end of the World Cup."

- Example: The Victoria's Secret Internet fashion show, May 18, 2000.

- Example: The Slashdot Effect:
  – "The spontaneous high hit rate upon a web server due to an announcement on a high volume news web site."

- Problem: Protecting other traffic on congested links.

Aggregate-based Congestion Control: Denial of Service Attacks

- Example: Denial of Service attacks, February 7 and 8, 2000:
  - Attacks on a large number of web sites across the U.S.
  - "It's completely clear that the entire Internet had higher packet loss and far lower reachability for several hours." - John Quarterman.
- Problem: Limiting the damage to the legitimate traffic at the site.
- Problem: Protecting the rest of the Internet.

The Mechanisms of Aggregate-based Congestion Control:

• Detect sustained congestion, as characterized by a persistent, high packet drop rate.

• Look at the packet drop history:

– See if some aggregate is heavily represented in the packet drop history.

– An aggregate is defined by destination address prefix, source address prefix, etc.

• If an aggregate is found:

– Preferentially drop packets from the aggregate before they are put in the output queue, to rate-limit aggregate to some specified bandwidth limit.

– Mahajan, Bellovin, Floyd, Ioannidis, Paxson, and Shenker, Controlling High Bandwidth Aggregates in the Network, February 2001.

Traffic Aggregates are Different from Flows:

• Similarities between the mechanisms for controlling aggregates and flows:

– Both use the packet drop history for identification.

– Both use rate-limiting before the output queue.

• Differences:

– Per-flow scheduling does not control aggregates.

– There is no simple fairness goal for aggregates, as for flows.

– Control of aggregates is heavily affected by policy, customer relationships, differentiated services, etc.

– A single flow could be in several different aggregates:

  – E.g., destination 192.0.0.0/12, or source www.victoriasecret.com.

– Aggregate-based congestion control (ACC) should only be invoked for extreme congestion.

A Thought Experiment of Aggregate-based Congestion Control (ACC):

- Under normal conditions, with no flash crowd:
  – N aggregates A_1-A_n share link with background traffic.
  – Packet drop rate p (e.g., $p = 0.01$).

- During flash crowd i from aggregate A_i, with no ACC at the router:
  – The drop rate is p_i (e.g., $p_i = 0.2$).
  – The throughput for A_j, for $j \neq i$, is roughly $\frac{1}{\sqrt{p_i/p}}$ of its value without the flash crowd (e.g., 1/5-th of its old value).

- During flash crowd i, with ACC at the router:
  – Assume that during the flash crowd, A_i is restricted to at most half the link bandwidth:
  – A_i's throughput is at worst halved, compared to the flash crowd with no ACC.
  – All other traffic has its throughput at worst halved, compared to times with no flash crowd (and its packet drop rate at most quadrupled).

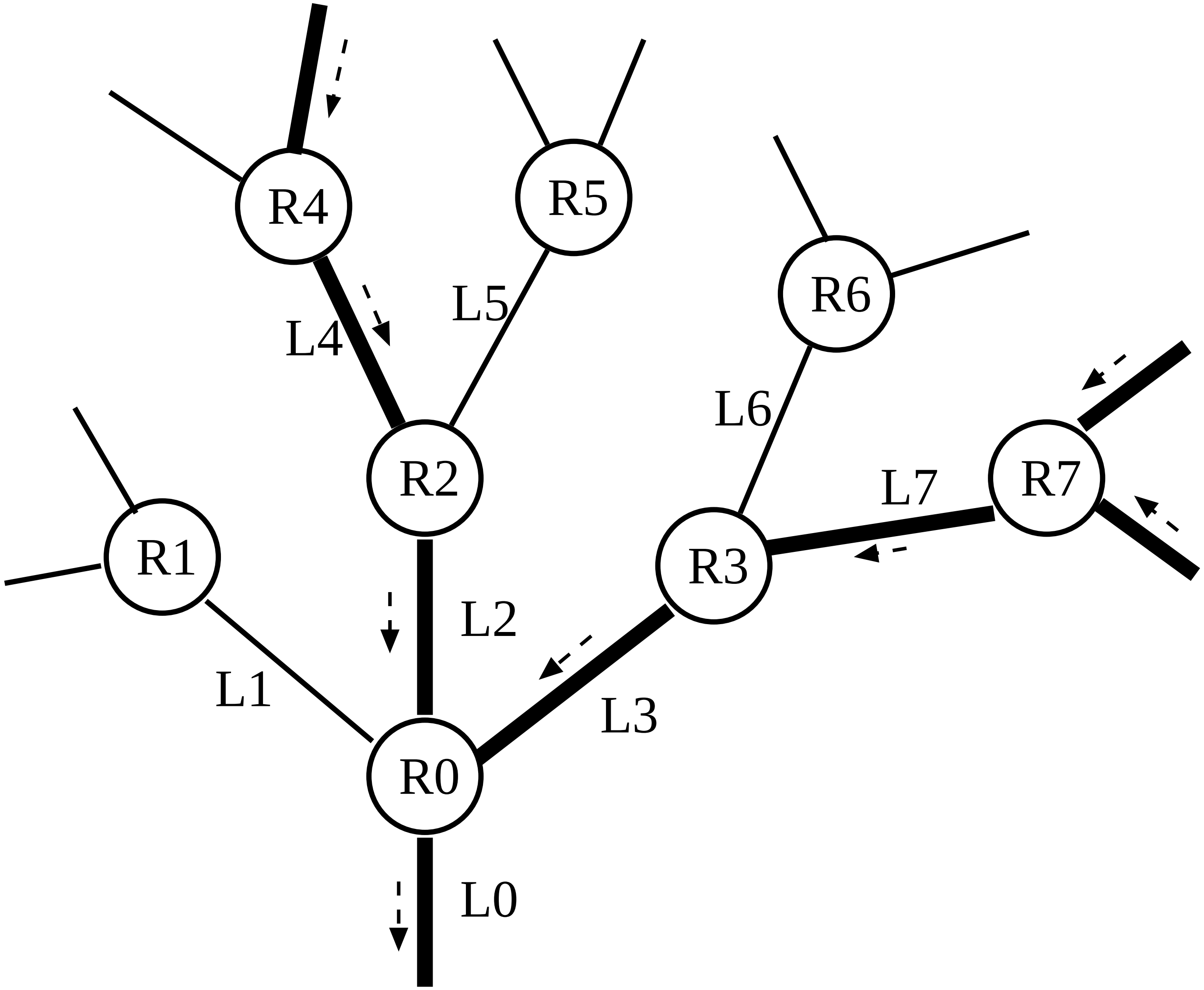
Now consider a Denial of Service (DOS) Attack:

- If an aggregate causing congestion is from a DOS attack, then the aggregate will contain both malicious traffic and legitimate, “good” traffic.

- We can not necessarily trust the IP source addresses.

- “Pushing-back” some of the rate-limiting of the aggregate to neighboring, upstream routers:
  - Limits the damage from the DoS attack, reducing wasted bandwidth upstream.
  - In some cases, allows rate-limiting to be concentrated more on the malicious traffic, and less on the good traffic within the aggregate.
  - Does not assume valid IP source addresses.

Illustration of pushback.



Questions about Aggregate-based Congestion Control?

- ACC helps traffic not in the aggregate, but why should we restrict the bandwidth given to a single aggregate in the first place?

- When does ACC with Pushback help an attacker to deny service to legitimate traffic within the aggregate?

-

-

Extra viewgraphs:

Pushback, Traceback, and Source Filtering:

- With Pushback, a router rate-limiting packets from aggregate A might ask upstream routers to rate-limit that aggregate on the upstream link.

- Pushback is orthogonal to ”traceback”, which tries to trace back an attack to the source.
  - Traceback allows legal steps to be taken against the attacker.
  - Traceback by itself does not protect the other traffic in the network.

- Pushback is orthogonal to source filtering, which limits the ability to spoof IP source addresses.
  - Source filtering is important in any case.
  - Pushback can be useful even when source addresses can be trusted.

The “steady-state model” of TCP: an improved version.

$$T = \frac{B}{RTT\sqrt{\frac{2p}{3}} + (2RTT)(3\sqrt{\frac{3p}{8}})p(1+32p^2)} \tag{1}$$

T: sending rate in bytes/sec

B: packet size in bytes

p: packet drop rate

– J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, Modeling TCP Throughput: A Simple Model and its Empirical Validation Proceedings of SIGCOMM’98

Section 5.3 on Fragmentation:

- “All ECN-capable packets SHOULD have the DF (Don’t Fragment) bit set.”

- “Reassembly of a fragmented packet MUST NOT lose indications of congestion.”

The ECN field with Differentiated Services:

- “The above discussion of when CE may be set instead of dropping a packet applies by default to all Differentiated Services Per-Hop Behaviors (PHBs) [RFC 2475].”

- “Specifications for PHBs MAY provide more specifics on how a compliant implementation is to choose between setting CE and dropping a packet, but this is NOT REQUIRED.”

- “A router MUST NOT set CE instead of dropping a packet when the drop that would occur is caused by reasons other than congestion or the desire to indicate incipient congestion to end nodes.”

\- In Section 5.