End-to-End Congestion Control

Previously on CS5229,

TCP Congestion Control

Not everyone uses TCP

UDP:

Media streaming
Gaming
VoIP

Why not congestion controlled?

1. UDP has low delay, no need full reliability

- Flash Networks BoosterWare: "For the Internet community at large, NetBooster exploits the capacity of the modem to maintain a constant data flow at its maximum rated speed, regardless of the network traffic load." (Flash Networks Press Release)

  From their White Paper on The BoosterWare Advantage: Enhancing TCP/IP: "BoosterWare, by contrast, abandons the effort to optimize the window size (a key source of bottlenecks) during transmissions; instead, window sizes are fixed according to pre-defined parameters negotiated between the client and the server once a connection has been established. BoosterWare can be viewed as a reliable, "no overhead" UDP (user datagram protocol)..."

- RUN Inc. ("RUN Inc. has found a way to squeeze more bandwidth out of existing TCP/IP networks without changing the network protocols or the applications that run over them.... In field tests over the Internet, runTCP has accelerated data transfers by as much four times." - PC Week Online, Sept. 4, 1997.)

- Sitara Networks Inc. ("Everyone talks about the "World Wide Wait", but no one does anything about it."). As discussed in IP Acceleration Software: Torquing Up TCP/IP, DataCommunications, January 1998: "Speedseeker can selectively suspend the TCP/IP congestion control mechanism when sending audio and video." See About Sitara in the News.

- RealAudio. "RealAudio 3.0 encoding algorithms have four different fixed data rates which can be used depending on the bandwidth requirements." (Audio Bandwidth)

- Jae Chung, Yali Zhu, and Mark Claypool, FairPlayer or FoulPlayer?--Head to Head Performance of RealPlayer Streaming Video Over UDP versus TCP, Technical Report N. WPI-CS-tr-02-17, Worchester Polytechnic Institute Computer Science Department, May, 2002.
  "In times of congestion, most RealVideo over UDP does respond to Internet congestion by reducing the application layer encoding rate, often achieving a TCP-Friendly rate. In times of severe congestion, RealVideo over UDP gets a proportionately larger share of the available bandwidth than does the same video over TCP."

Sally Floyd, http://www.icir.org/floyd/tcp_unfriendly.html

Why not congestion controlled?

2. No incentive. OTOH, there are incentives NOT to use congestion control.

“Unresponsive Flows”

Bad: lead to unfairness and congestion collapse.

Unfairness:

unresponsive flows consume more bandwidth than congestion controlled flows.

NS-2 Demo

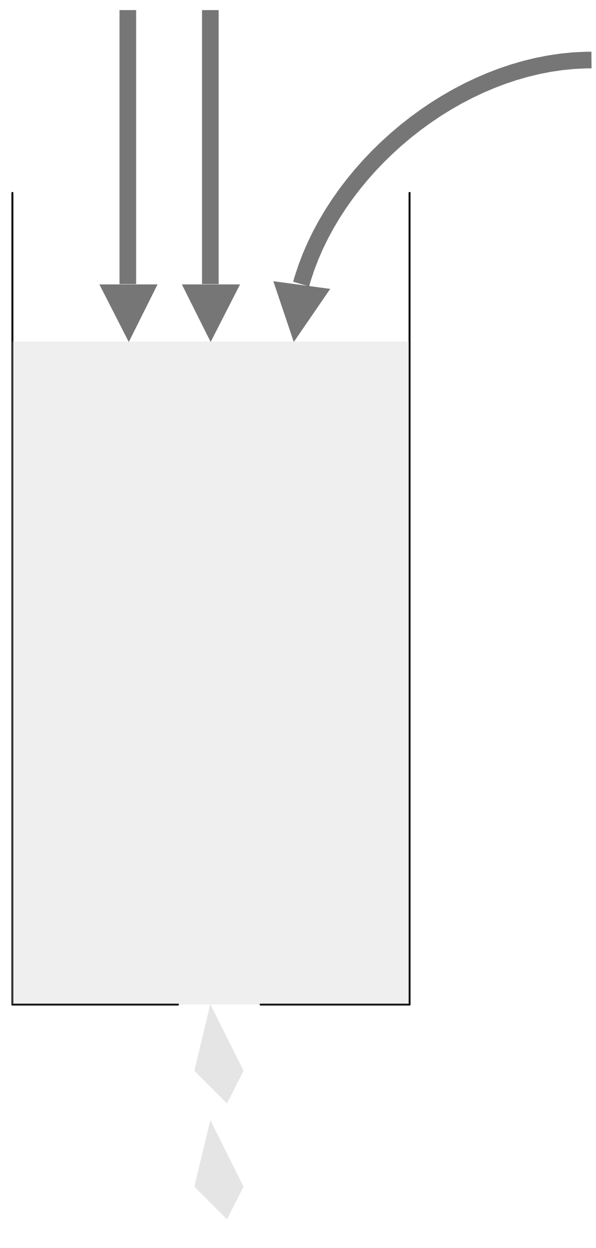
Unfairness also exists between:

1. TCP flows with different RTT

2. Different TCP versions

Bad: lead to unfairness and congestion collapse.

Congestion Collapse:
wasting bandwidth by sending packets that will be dropped



Why not congestion controlled?

UDP has low delay, no need full reliability

Provide Congestion Controlled, Unreliable Transport Protocol

Why not congestion controlled?

No incentive.

Provide Incentives for End-to-End Congestion Control

Sally Floyd and Kevin Fall

"Promoting End-to-End Congestion Control in the Internet"

TON, 1999

What mechanisms can we add to the router to provide incentives for congestion control?

Idea: Identify unresponsive flows, then drop their packets or regulate their rate.

Note: Not scalable to large number of flows (eg in core routers).

How to identify unresponsive flows in a router?

Approach 1:
TCP Un-Friendly Flows

Definition. A flow is TCP Friendly if its arrival rate does not exceed the arrival of a conformant TCP connection in the same circumstances.

"Same circumstances": same loss rate, RTT, packet size

$$B_{TCP} = \frac{MSS}{RTT\sqrt{\frac{2p}{3}} + min(1, 3\sqrt{\frac{3p}{8}})T_O p(1 + 32p^2)}$$

The paper uses a rough approximation

$$B_{TCP} \leq 1.22 \frac{MSS}{RTT\sqrt{p}}$$

$$1.22 \frac{MSS}{RTT\sqrt{p}}$$

MSS: Maximum packet size in bytes over all outgoing links

p: Packet drop rates over all outgoing links

R: Twice the 1-way propagation delay of outgoing links

$$1.22 \frac{MSS}{RTT \sqrt{p}}$$

The expression will overestimate the fair throughput for TCP.

Thus, not all unfriendly flows will be identified.

Approach 2:
Unresponsive Flows

Does the packet arrival rate of a flow reduce appropriately when packet drop rate increase?

If packet drop rate increases by x%, then packet arrival rate should decrease by sqrt(x)%

Does Not Work:

when packet drop rate is constant

Does Not Work:

packet might be dropped by earlier router

Does Not Work:

A flow has an incentive to start with high throughput

Approach 3:

Flows with Disproportionate Bandwidth

A flow should share 1/n of total bandwidth

When congestion is low
(packet drop rate is low),
skewness is OK.

Condition 1: If a flow's bandwidth is more than ln(3n)/n of the aggregate, then it is using disproportionate share.

(ln(3n)/n : magic)

Condition 2: If a flow's bandwidth is more than

$$1.22 \frac{MSS}{RTT \sqrt{p}}$$

For MSS=512 and RTT=0.05s

If a flow's bandwidth is more than ln(3n)/n of the aggregate flow bandwidth, then it is using disproportionate share.

(ln(3n)/n : magic)

Does Not Work:

flows with short RTT will be considered as disproportionate

Does Not Work:

the only flow with sustained demand (long live) will be considered as disproportionate.

Why not congestion controlled?

No incentive.

Why not congestion controlled?

UDP has low delay, no need full reliability

E. Kohler, M. Handley, S. Floyd

"Designing DCCP: Congestion Control without Reliability"

SIGCOMM, 2006

DCCP:

Datagram Congestion Control Protocol

A unreliable transport protocol with “plug-in” congestion control mechanism

Why not application layer?

Different applications would have to implement it.
Hard to implement.

Why not application layer?

Make use of ECN info from IP.

ECN bits in IP header is marked by router if the router is congested, and can be used as congestion signal at the sender.

Why not TCP?

Application can’t choose congestion control algorithm

Why multiple congestion control plug-ins?

Different applications need different congestion control behavior.

Pick one of

CCID**2**: TCP-like

CCID**3**: TFRC

CCID2:
TCP-Like Congestion Control

DCCP uses acknowledgements with “ACK Vector” (similar to SACK block). CCID2 is similar to TCP SACK’s congestion control algorithm.

CCID3: TFRC
TCP-Friendly Rate Control

$$B_{TCP} = \frac{MSS}{RTT\sqrt{\frac{2p}{3}} + min(1, 3\sqrt{\frac{3p}{8}})T_O p(1 + 32p^2)}$$

In CCID3, receiver sends ACK once every RTT to report lost events.

One loss event: one or more lost or marked packets from a window of data.

AIMD: throughput fluctuates
TFRC: smooth throughput

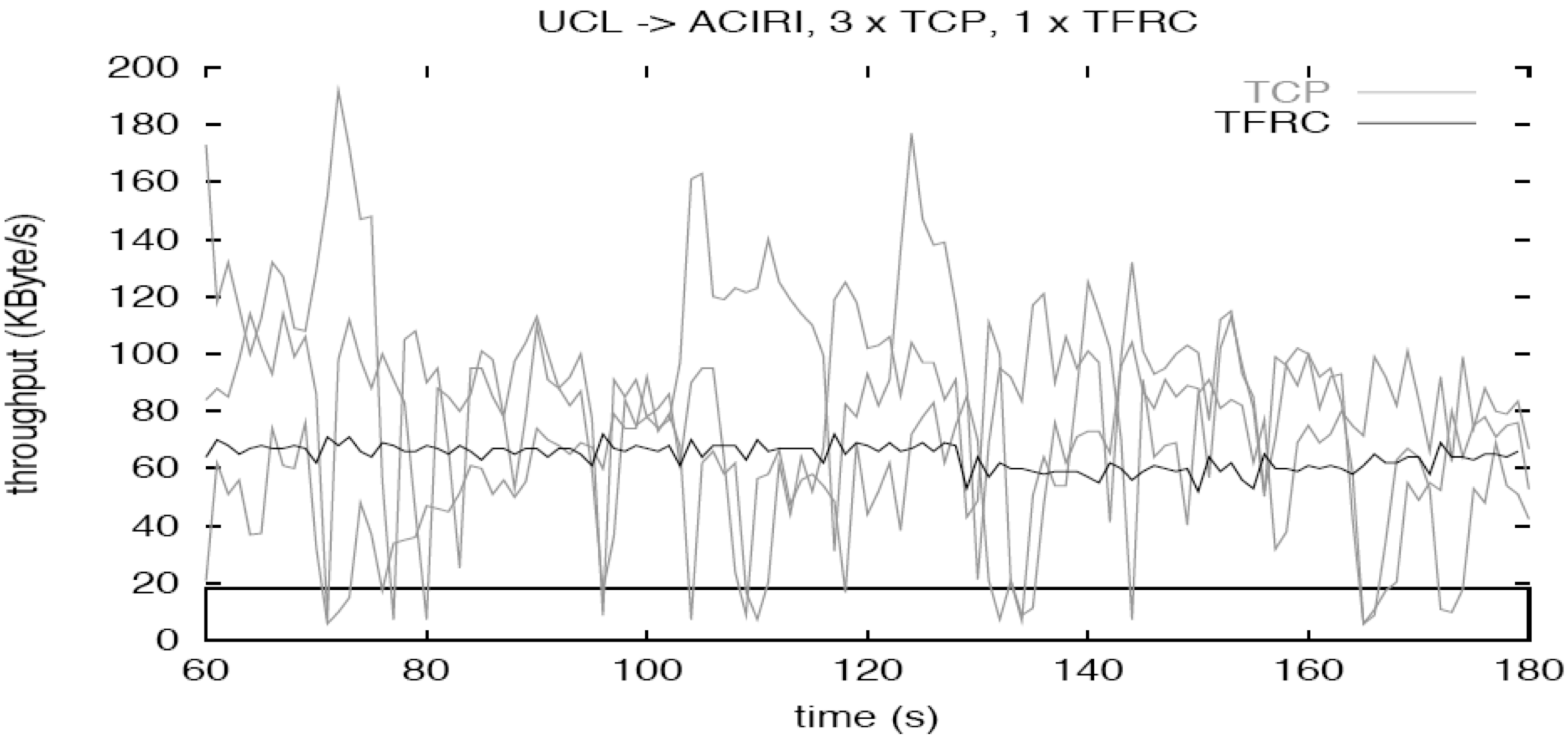


Figure 15: Three TCP flows and one TFRC flow over the Internet.

Equation-Based Congestion Control for Unicast Applications*

Sally Floyd, Mark Handley
AT&T Center for Internet Research at ICSI (ACIRI)

Jitendra Padhye
University of Massachusetts at Amherst

Jörg Widmer
International Computer Science Institute (ICSI)

Other DCCP features:

Reliable connection setup, teardown, negotiation.

Other DCCP features:

A packet stream protocol
(not a byte stream protocol)

End-to-End Congestion Control