On the Evolution of End-to-end Congestion Control in the Internet: An Idiosyncratic View

Sally Floyd

IMA Workshop on Scaling Phenomena in Communication Networks
October 1999

Outline of talk:

- The danger of congestion collapse, and the role of congestion control in the Internet.

- Change and heterogeneity as conditions of the Internet.

- Speculations on the future evolution of end-to-end congestion control in the Internet.

Sub-themes:

- The Internet is a work in progress, with no central control or authority, many players independently making changes, and many forces of change (e.g., new technologies, new applications, new commercial forces, etc.)

- So far, the success of the Internet has rested on the IP architecture's robustness, flexibility, and ability to scale, and not on its efficiency, optimization, or fine-grained control.

- The rather decentralized and fast-changing evolution of the Internet architecture has worked reasonably well to date. There is no guarantee that it will continue to do so.

- The Internet is like the elephant, and each of us is the blind man who knows only the part closest to us.
  - The part of the Internet that I see is end-to-end congestion control.

- The danger of congestion collapse, and the role of congestion control in the Internet.

-
-

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

- To avoid congestion collapse.

- Fairness.

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

Why is congestion collapse a concern? The environment of the Internet before 1988:

- Datagram routing, for robustness [Clark88].

  – Of the seven listed goals for the DARPA Internet Architecture, the most important goal was survivability in the face of failure.

  – Datagram routing was selected as the technique for multiplexing, instead of circuit switching, because it matched the applications being supported (e.g., remote login).

- TCP used flow control to control the use of buffer space at the receiver, and Go-Back-N retransmission after a packet drop for reliable delivery.

- FIFO scheduling at routers, packets dropped upon buffer overflow.

- Starting in October 1986, the Internet had a series of congestion collapses.

Classical congestion collapse:

Congestion collapse occurs when the network is increasingly busy, but little useful work is getting done.

Problem: Classical congestion collapse:
Paths clogged with unnecessarily-retransmitted packets [Nagle 84].

Fix: Modern TCP retransmit timer and congestion control algorithms [Jacobson 88].

TCP congestion control:

- Packet drops as the indications of congestion.

- TCP uses Additive Increase Multiplicative Decrease (AIMD) [Jacobson 1988].
  - Decrease congestion window by 1/2 after loss event.
  - Increase congestion window by one packet per RTT.

- 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.

Fragmentation-based congestion collapse:

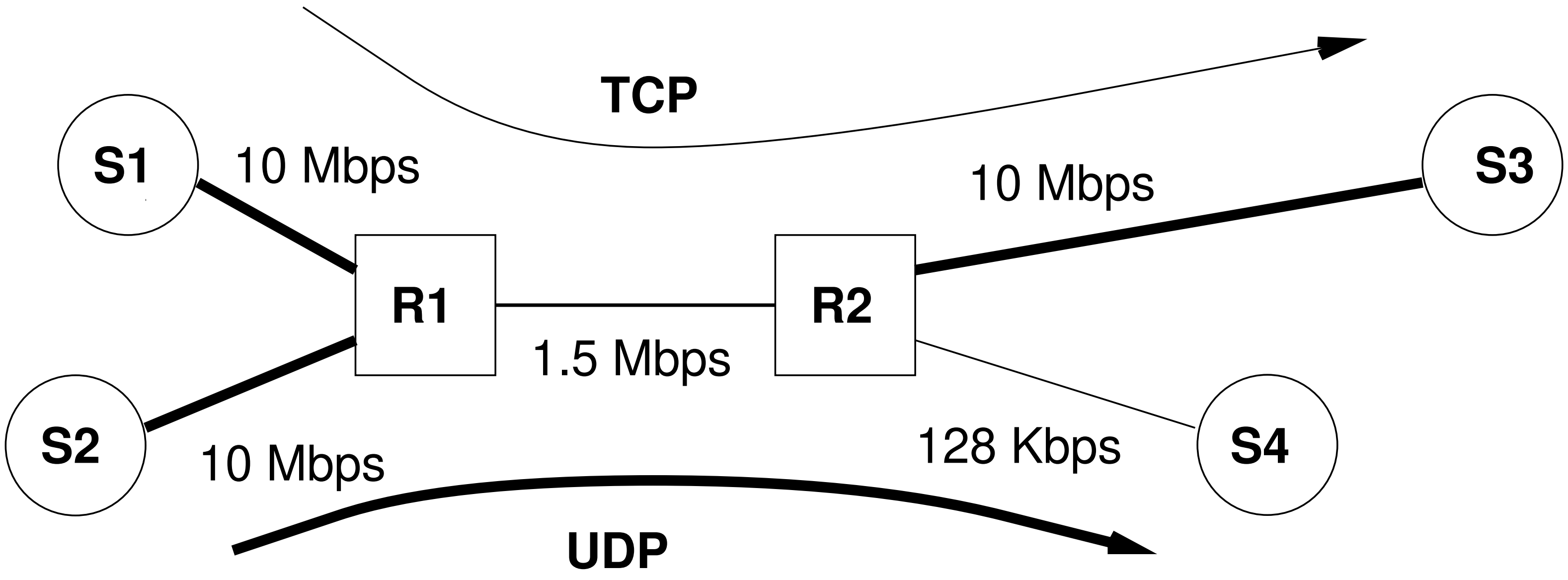
Problem: Paths clogged with fragments of packets invalidated because another fragment (or cell) has been discarded along the path. [Kent and Mogul, 1987]

Fix: MTU discovery [Kent et al, 1988],
Early Packet Discard in ATM networks [Romanow and Floyd, 1995].

Congestion collapse from undelivered packets:

Problem: Paths clogged with packets that are discarded before they reach the receiver [Floyd and Fall, 1999].

Fix: Either end-to-end congestion control, or a “virtual-circuit” style of guarantee that packets that enter the network will be delivered to the receiver.



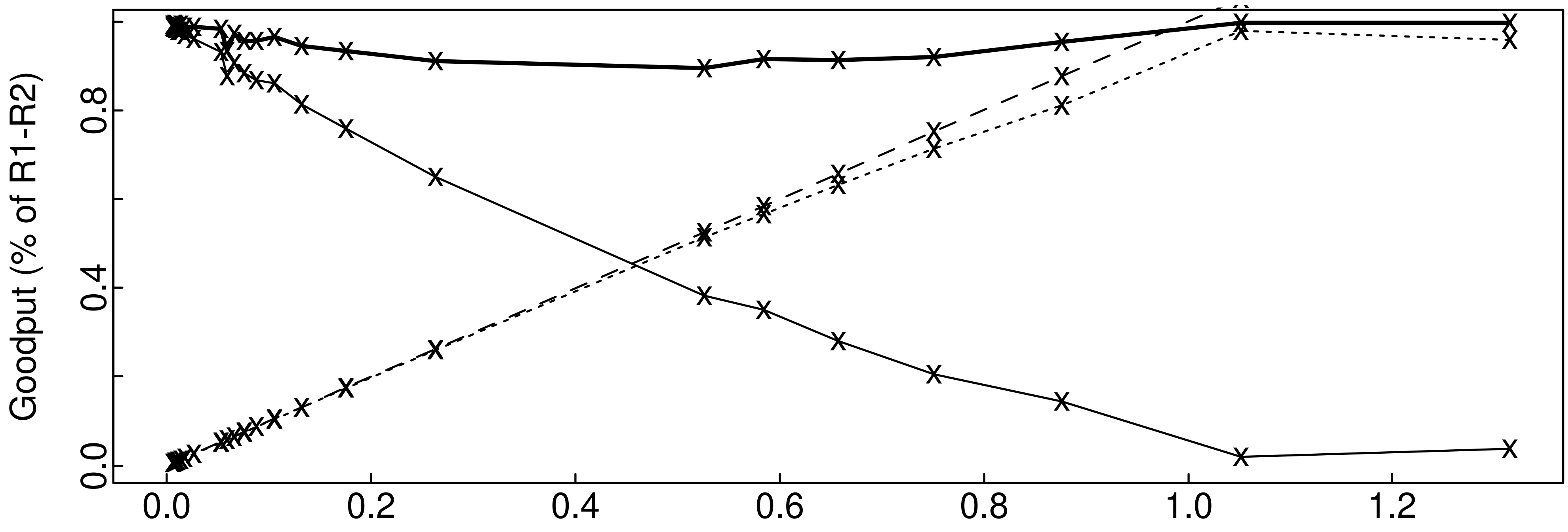
Why do we need end-to-end congestion control?

-
- Fairness.
-

What is the fairness goal? (the pragmatic answer)

- No connection/session/end-node should hog the network resources.
  - TCP is the dominant transport in the Internet (90-95% of the bytes/packets)
  - Routers are likely to use FIFO scheduling.
  - New forms of traffic that compete with TCP as best-effort traffic in FIFO queues should not be significantly more (or less) aggressive than TCP.

Why is fairness a concern?



X-axis: UDP Arrival Rate (% of R1-R2). Dashed Line: UDP Arrivals; Dotted Line: UDP Goodput; Solid Line: TCP Goodput; Bold line: Aggregate Goodput

Simulations showing three TCP flows and one UDP flow (without end-to-end congestion control), with a congested link using FIFO scheduling.

What is the fairness goal? (other possible answers)

- Fairness goals not based on pricing:
  - Min-max fairness: On each link of the network, each entity has an equal claim to the bandwidth of that link. (e.g., Fair Queueing.)
  - "Global" fairness: Each entity has an equal claim to the scarce resources (where an entity traversing N congested links is using more scarce resources than an entity traversing one congested link).
  - Fairness based on the number of receivers for a packet.
  - Other fairness goals ...

- Fairness goals based on pricing:
  - Pricing: For some services, bandwidth is allocated to those willing to pay for it. (E.g., intserv, diffserv.)
  - Congestion-based pricing: The "cost" of the bandwidth on each link varies as a function of the level of congestion (e.g., the packet drop rate).

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.

How can end-to-end congestion control be useful to an application for its own reasons?

• In an environment of either per-flow scheduling or small-scale statistical multiplexing:

– The loss and delay experienced by a flow is affected by its own sending rate.

– The use of end-to-end congestion control can reduce unnecessary loss and delay for that flow.

How can end-to-end congestion control be useful to an application for its own reasons? Part 2:

- In an environment of FIFO scheduling and large-scale statistical multiplexing at all congestion points:
  - The loss rate and delay experienced by a flow is largely independent of its own sending rate (holding the congestion control behavior of all other flows fixed).
  - End-to-end congestion control can be useful to a flow to avoid mechanisms that could be deployed by the network to penalize best-effort traffic that doesn't use end-to-end congestion control in a time of congestion.

- Tragedy of the commons is avoided in part because the "players" are not individual users determining their own end-to-end congestion control strategy and "gaming" against other users.

Outline of talk:

-
- Change and heterogeneity as conditions of the Internet.
-

Changes that affect the evolution of congestion control:

- The web, and the web caching infrastructure.
- Changes to TCP:
  – Fast Recovery, Selective Acknowledgements (SACK), larger initial windows.
- Active queue management (e.g., RED), non-FIFO scheduling in routers.
- Explicit Congestion Notification.
- Applications that don't use TCP:
  – Streaming multimedia, reliable and unreliable multicast.
  – And new end-to-end congestion control mechanisms to support them.

Related issues: Explicit Congestion Notification (ECN)

• Active queue management (e.g., RED) is being incorporated into routers.

– Routers measure the average queue size, and probabilisticly drop packets before buffer overflow, as an indication of congestion to end nodes.

• Given that routers are not necessarily waiting until buffer overflow to drop a packet, routers can set an ECN bit in the packet header instead of dropping the packet, to inform end-nodes of congestion.

• ECN is an experimental addition to the IP architecture [RFC 2481].

– ECN-Capable Transport (ECT) indication from sender to router.

– Congestion Experienced (CE) indication from router to receiver.

– For TCP, TCP-level feedback from TCP receiver to TCP sender about ECN indications.

More changes...

- Differentiated services (diffserv) and integrated services (intserv).

- New link-level technologies.

  – E.g., Wireless links with non-congestion-related packet drops, variable delay, and modile users.

- Changes in the level of granularity:

  – E.g., Mechanisms for sharing congestion control state among connections with the same source and destination IP addresses.

- New pricing mechanisms.

Focusing on one change in progress: New end-to-end congestion control mechanisms.

Why not use TCP for unicast streaming media?

- Reliable delivery is not needed.

- Acknowledgements are not returned for every packet, and the application would prefer a rate-based to a window-based approach anyway.

- Cutting the sending rate in half in response to a single packet drop is undesirable.

Other possibilities for end-to-end congestion control for unicast streaming media?

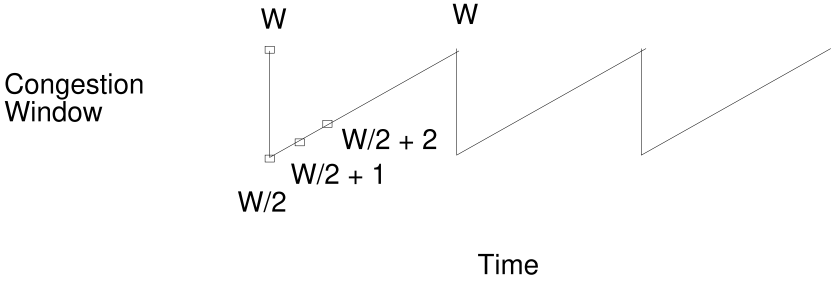
- Use a rate-based version of TCP's congestion control mechanisms, without TCP's ACK-clocking.
  - The Rate Adaption Protocol (RAP) [RH99].

- AIMD with different increase/decrease constants.
  - E.g., decrease multiplicatively by 3/4, increase additively by 3/7 packets/RTT.

- Equation-based congestion control: adjust the sending rate as a function of the longer-term packet drop rate.

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, $\frac{W}{2}$, ...

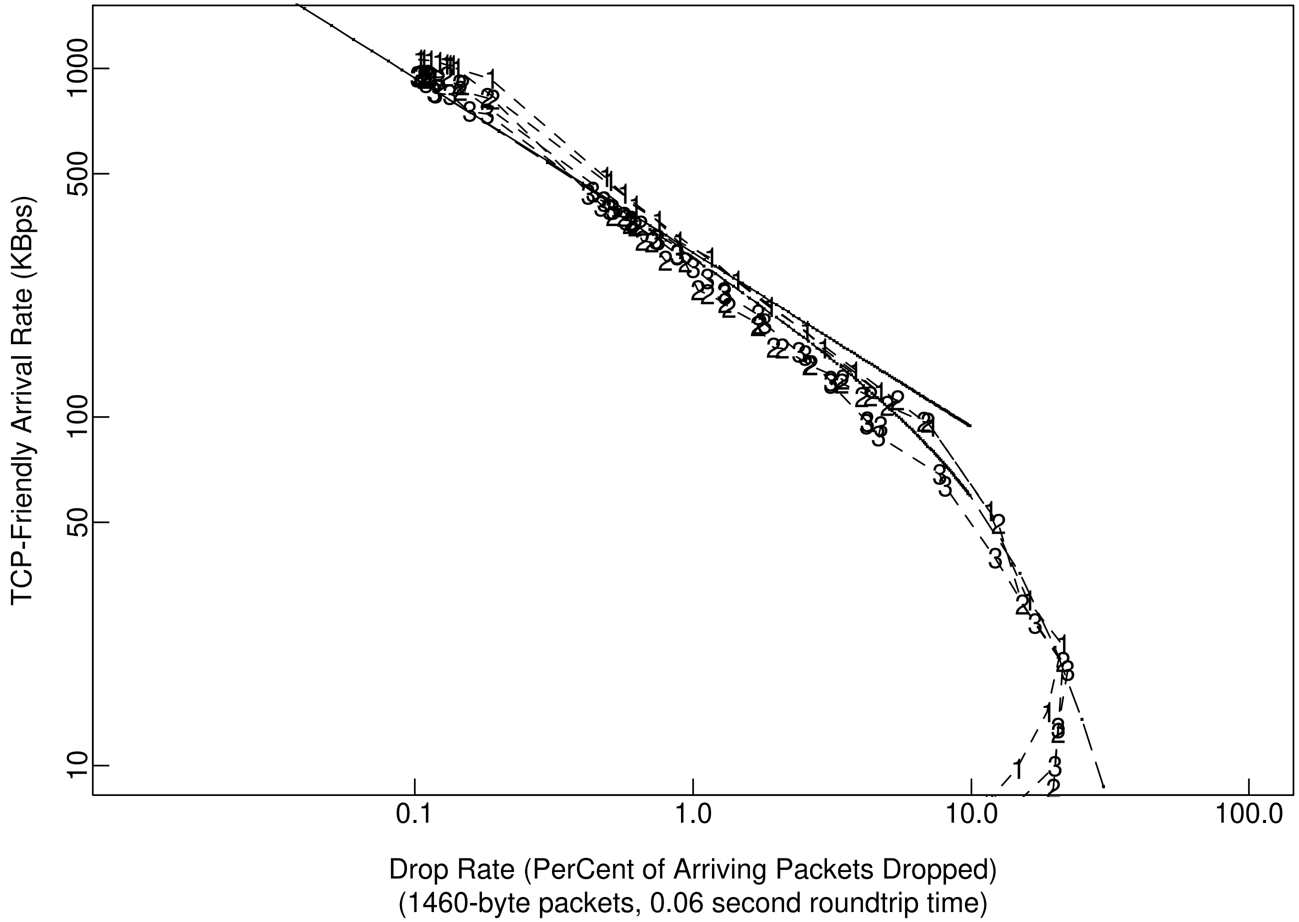


- The maximum sending rate in packets per roundtrip time: W
  - The maximum sending rate in byes per second: WB/R
  - The average sending rate T: $T = (3/4)WB/R$

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

- The result: $T = \frac{\sqrt{6}B}{2R\sqrt{p}} = \frac{\sqrt{3/2}B}{R\sqrt{p}}$

Verifying the “steady-state model” of TCP:



Solid line: the simple equation characterizing TCP

Numbered lines: simulation results

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

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

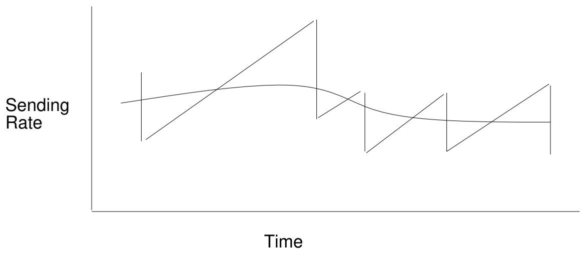
T: sending rate in bytes/sec

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

Equation-based congestion control:



• Use the TCP equation characterizing TCP's steady-state sending rate as a function of the RTT and the packet drop rate.

• Over longer time periods, maintain a sending rate that is a function of the measured roundtrip time and packet loss rate.

• The benefit: Smoother changes in the sending rate in response to changes in congestion levels.

• The justification: It is acceptable not to reduce the sending rate in half in response to a single packet drop.

• The cost: Limited ability to make use of a sudden increase in the available bandwidth.

-

-

- Speculations on the future evolution of end-to-end congestion control in the Internet.

The future of congestion control in the Internet: several possible views:

- View #1: No congestion, infinite bandwidth, no problems.
- View #2: The “co-operative”, end-to-end congestion control view.
- View #3: The game theory view.
- View #4: The congestion-based pricing view.
- View #5: The virtual circuit view.
- The darker views: Congestion collapse and beyond.

My own opinion would be that the network research community can use not only the further development of self-similarity and multifractals in networking, but more analysis and understanding in many domains:

- Congestion control mechanisms.
- Global traffic dynamics.
- Asymptotic behavior.
- ...

Global traffic dynamics:

• Synchronized routing messages [FJ94].

• Undesired synchronization or emergent behavior for other network traffic?

– Possible feedback loop: The TCP feedback loop of a data packet followed by an acknowledgement packet followed by another data packet.

– Possible feedback loop: Feedback loops in the network of connections A, B, and C, with a loop where A and B share a congested link, B and C share a congested link, and C and A share a congested link.

I would not recommend indiscriminate proposals for new architectures that would need to be ubiquitously deployed in all of the routers and end-nodes of the global Internet:

“What simulations and measurements of prototype implementations do you have that show that it is better than alternatives? What objective concrete evidence do you have that it is worth the trouble of changing many 1,000,000s of hosts and many 100,000 routers?”

- [S99], Email to the end2end-interest mailing list.

References:

[Clark88] D. D. Clark, The Design Philosophy of the DARPA Internet Protocols, SIGCOMM 88, August 1988.

[Floyd and Fall, 1999] Floyd, S., and Fall, K., Promoting the Use of End-to-End Congestion Control in the Internet, IEEE/ACM Transactions on Networking, August 1999. URL “http://www.aciri.org/floyd/papers.html”.

[FJ94] Floyd, S., and Jacobson, V., The Synchronization of Periodic Routing Messages. IEEE/ACM Transactions on Networking, V.2 N.2, p. 122-136, April 1994. URL “http://www.aciri.org/floyd/papers.html”.

[GK99] R.J. Gibbens and F. P. Kelly, Resource pricing and the evolution of congestion control, Automatica 35 (1999).
URL “http://www.statslab.cam.ac.uk/ frank/evol.html”.

[HRS99] Hari Balakrishnan, Hariharan S. Rahul, and Srinivasan Seshan, An Integrated Congestion Management Architecture for Internet Hosts, SIGCOMM 99, September 1999.
URL “http://inat.lcs.mit.edu/papers/BRS99.html”.

[Jacobson 88] Jacobson, V., Congestion Avoidance and Control. Proceedings of SIGCOMM ’88 (Palo Alto, CA, Aug. 1988) URL “http://www-nrg.ee.lbl.gov/nrg-papers.html”.

[Kent and Mogul, 1987] C. Kent and J. Mogul, ”Fragmentation Considered Harmful,” ACM Computer Communication Review, vol. 17, no. 5, Aug. 1987.

[Kent et al, 1988] J. C. Mogul, C. A. Kent, C. Partridge and K. McCloghrie, ”IP MTU discovery options,” Internet Engineering Task Force, RFC 1063, Jul. 1988.

[Nagle 84] John Nagle, ”Congestion control in IP/TCP internetworks,” ACM Computer Communication Review, vol. 14, no. 4, pp. 11–17, Oct. 1984.

[RFC 2481] Ramakrishnan, K.K., and Floyd, S., A Proposal to add Explicit Congestion Notification (ECN) to IP. RFC 2481, Experimental, January 1999. URL “http://www.aciri.org/floyd/papers.html”.

[RH99] RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet, R. Rejaie, M. Handley, D. Estrin. Proc. Infocom 99. URL “http://www.aciri.org/mjh/papers.html”.

[Romanow and Floyd, 1995] Romanow, A., and Floyd, S., Dynamics of TCP Traffic over ATM Networks. IEEE JSAC, V. 13 N. 4, May 1995, p. 633-641. URL “http://www.aciri.org/floyd/papers.html”.

[S99] Vernon Schryver, Email Message-Id: 199910191533.JAA22180@calcite.rhyolite.com to the end2end-interest mailing list.

[TCP-Friendly Web Page] The TCP-Friendly Web Page,
URL "http://www.psc.edu/networking/tcp_friendly.html".

[Whetten 98] B. Whetten, J. Conlan, A Rate Based Congestion Control Scheme for Reliable Multicast. Technical White Paper, GlobalCast Communications, October 1998.