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

Sally Floyd

Nortel Design Forum

April 1999

Outline of talk:

- A quick sketch of the evolution of end-to-end congestion control in the Internet.

- The danger of congestion collapse.

- A discussion of interactions between various pieces.

- 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.

Disclaimers:

- 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.
  - Other parts of the elephant:
    routing; security; the web; the last mile; etc.

- This talk is an evolving attempt to understand where we have been and where we might be going.

- The ideas in this talk are not original, but are drawn from many different sources in the IETF and network research communities.

- A quick sketch of the evolution of end-to-end congestion control in the Internet.

-

-

-

The environment of the Internet before 1988:

• Datagram routing, for robustness (“The Design Philosophy of the DARPA Internet Protocols”, Clark 1998).

– 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.

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

Congestion control in the Internet: 1988.

- Routers:
  – FIFO scheduling;
  – Packets dropped upon buffer overflow;

- Transport protocols:
  – TCP incorporated end-to-end congestion control, based on a principle of ‘conservation of packets’, to prevent future congestion collapse. Tahoe TCP enters slow-start in response to a packet drop, then slowly rebuilds the congestion window [Jacobson 1988];
  – Packet drops as the only indications of congestion;

Changes in the last ten years:

- The web:
  - Many short web transfers ("web mice"), but most packets still belong to the larger transfers ("elephants").
  - Asynchronous instead of synchronous communications is still dominant.
  - A growing web caching and data dissemination infrastructure.

- Transport:
  - TCP is still the dominant transport protocol, but there is increasing heterogeneity:
  - UDP: Realaudio and realvideo;
  - Reliable and unreliable multicast;
  - IP Telephony.

Changes in the last ten years, continued:

- Changes to TCP:

  – Fast Recovery (Reno and NewReno TCP): No need to slow-start after a packet drop. Simply reduce the congestion window in half.

  – Selective Acknowledgements (SACK): More information from the receiver to the sender about data received out of order.

  – Larger initial windows:

    An initial window of two packets is proposed standard,

    an initial window of three or four packets is experimental;

  – TCP over wireless, over satellite.

  – TCP over ATM: changes to ATM (Early Packet Discard), not to TCP.

Congestion control in the Internet: changes in progress:

- The deployment of active queue management (e.g., RED, WRED);

- Diverse scheduling algorithms in routers
  (e.g., per-flow or class-based scheduling);

- Differentiated services (diffserv);

- Continued development of the web caching and data dissemination infrastructures.

Changes still in the research or standardization stages:

- Explicit Congestion Notification.

- New end-to-end congestion control mechanisms:
  - equation-based congestion control for unicast and multicast;
  - layered multicast, with receivers subscribing and unsubscribing;

- Mechanisms for sharing congestion control state among connections with the same source and destination IP addresses.
  - More speculatively, mechanisms for sharing congestion control state among macroflows.

- Router mechanisms for encouraging end-to-end congestion control.

- Possible changes to TCP:
  - Fewer retransmit timeouts?
    (From improved responses to a single duplicate acknowledgement.)
  - A more moderate response to a single packet drop? (Coupled with a more moderate packet increase rate.)
  - Smarter slow-start procedures?
  - Explicit Loss Notification?
  - Larger initial windows?

(Perhaps contingent on the deployment of mechanisms for sharing congestion control state among multiple TCP connections to the same destination.)

Revolutions or incremental changes that never happened:

- Some of the failed revolutions:
  - Integrated services (intserv).
  - The One Technology: global end-to-end ATM.

- Why didn't these things take?

Were they premature, or were these simply not the right direction?

-
- The danger of congestion collapse.
-
-

Scenarios for 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].

– Unnecessarily-retransmitted packets from users hitting the ”Reload” button on their web browsers are a similar problem at the application level.

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]

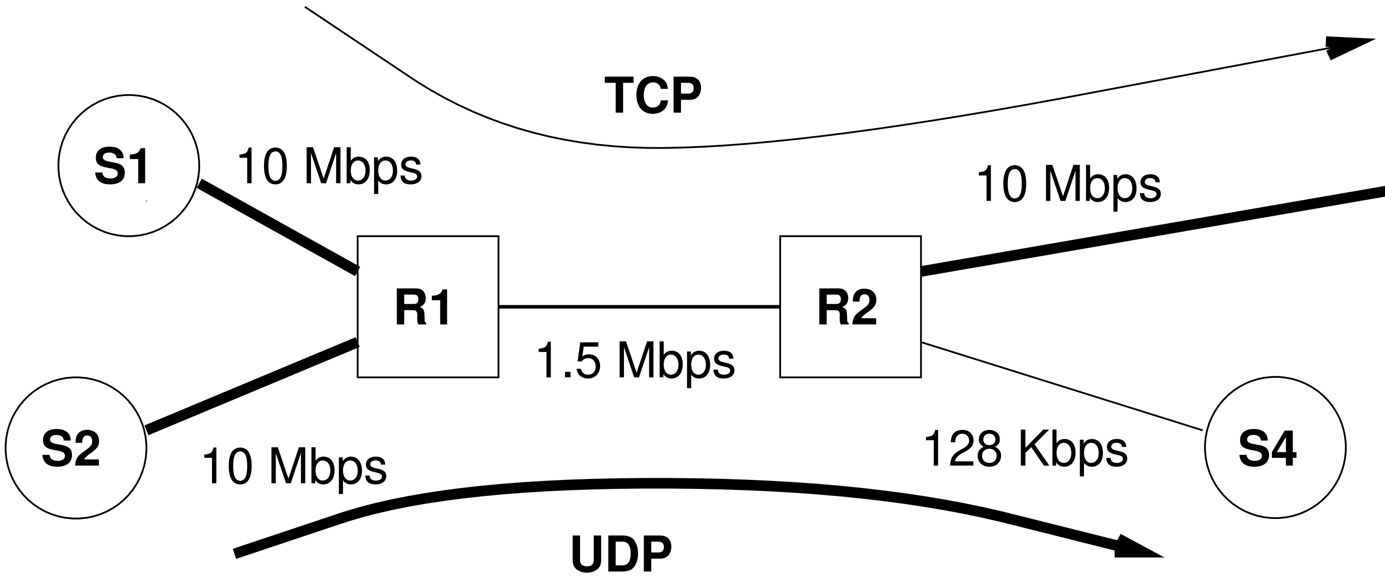
Fix: MTU discovery [Kent et al, 1988],
Early Packet Discard in ATM networks [Romanow and Floyd, 1995].

Congestion collapse from undelivered packets:

Problem: Paths clogged with packets that are discarded before they reach the receiver [Floyd and Fall, 1998].

Fix: End-to-end congestion control (or virtual circuits).

Disincentives at the routers for flows that do not use either end-to-end congestion control or explicit bandwidth allocation mechanisms.



Congestion collapse from undelivered packets, continued.

• Congestion collapse from undelivered packets is a danger even in the presence of better-than-best-effort differentiated services, Explicit Congestion Notification, per-flow scheduling, router mechanisms to encourage end-to-end congestion control, and all.

• There are only two ways to avoid congestion collapse from undelivered packets:

 – The effective use of end-to-end congestion control; or

 – A virtual-circuit style of guarantee that packets that enter the network will be delivered to the receiver.

Congestion collapse from increased control traffic:

Problem: With increasing load, an increasing fraction of the bytes in the network belong to control packets or packet headers.

Fix: Not a present danger. Avoid increasing levels of control traffic as levels of congestion increase.

Congestion collapse from stale or unwanted packets:

Problem: With increasing load, an increasing fraction of the packets arriving at the receiver are no longer wanted (or were never wanted).

Fix: Not a present danger. Avoid unbounded delays or badly-designed web “push” mechanisms.

-
-
- Interactions between various pieces:
-

Interactions between active queue management, scheduling, and Explicit Congestion Notification:

- Explicit Congestion Notification presupposes some form of active queue management.

- Per-flow scheduling is incomplete without active queue management.

- How would Explicit Congestion Notification be used in routers with per-flow scheduling?

Interactions between scheduling, router mechanisms, and end-to-end congestion control:

• Would ubiquitous per-flow scheduling (or ubiquitously-deployed router mechanisms to police flows that don't do end-to-end congestion control) eliminate the need for end-to-end congestion control? Nope.

– The only thing that would eliminate the need for end-to-end congestion control would be network mechanisms that only allowed packets to enter the network when there were reasonable guarantees that the packets would be delivered to the intended receiver(s).

Interactions between differentiated services and end-to-end congestion control:

• Does ECN eliminate the need for differentiated services, by dramatically improving the performance of best-effort traffic?

Nope.

• Does differentiated services (coupled with new pricing models) eliminate the need for end-to-end congestion control?

Nope.

• Would the ubiquitous deployment of new scheduling algorithms (e.g., class-based or per-flow queueing) perhaps with differentiated services, allow the deployment of end-to-end congestion control mechanisms that would not necessarily compete fairly with TCP in an environment of FIFO scheduling?

– Hopefully.

Interactions between web caching infrastructures and end-to-end congestion control:

- A ubiquitous web caching intrastructure might change traffic dynamics, in that many more connections would travel short distances (cache to cache, or cache to user) rather than long ones.

- A ubiquitous web caching intrastructure would open up new possibilities for congestion collapse, in terms of increasing levels of control traffic, or increasing levels of “push” data that is never delivered to a receiver.

Interactions between link-level technologies and end-to-end congestion control:

• Wireless links can have higher bit-error rates that are interpreted as indications of congestion by the transport protocol.

• Link-level retransmissions can interact with end-to-end retransmissions, or with end-to-end estimates of the roundtrip time, or with retransmit timeout algorithms.

• Link-level or cloud-level congestion control can interact with end-to-end congestion control.

PILC, Performance Implications of Link Characteristics.
URL "http://pilc.lerc.nasa.gov/pilc/".

Interactions between higher layers and end-to-end congestion control:

• Many web browsers open four concurrent TCP connections to the same destination, with each TCP connection having its own independent end-to-end congestion control.

– HTTP could use persistent connections, and open several transfers over a single TCP connection.

– A single congestion window could be shared among multiple TCP connections.

– A congestion manager could share congestion control state among multiple TCP and UDP connections.

Multiplexing, TCP, and UDP: Pointers to the Discussion.
URL “http://www.aciri.org/floyd/tcp_mux.html”

Interactions between congestion and pricing:

- One conjecture is that a key component of congestion in the Internet is due not to a lack of available bandwidth, but to the underlying economic structure of an ISP-based Internet.
  - The conjecture is that much of the congestion is at the public exchange points; and that ISPs have an incentive to have limited bandwidth to the public exchange points, to give other ISPs a concrete incentive to enter into peering agreements with them.

-
-
-
- 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 virtual circuit view.
- The darker views: Congestion collapse and beyond.

View #1: No congestion, infinite bandwidth, no problems.

- No congestion, essentially infinite bandwidth, no problems.

Well, if this happens, that is fine. I wouldn't want to count on it in all places all of the time.

View #2: The “co-operative”, end-to-end congestion control view.

- The ubiquitous use of end-to-end congestion control for best-effort and better-than-best-effort traffic, encouraged by policing mechanisms at the routers.

- “Smoother” and less obtrusive mechanisms for end-to-end congestion control, with active queue management, Explicit Congestion Notification, equation-based congestion control, mechanisms for detecting unused bandwidth, etc.

- End-to-end bandwidth guarantees in some form, used by that small subset of traffic with hard bandwidth requirements.

- Traffic dominated by asynchronous communications (helped by a global caching infrastructure), with an significant mix of synchronous unicast and multicast communications.

- Evaluation: It has mostly worked so far, but how well will it scale?

View #3: The game theory view.

- Ubiquitous per-flow scheduling.

- End users greedily optimizing their own utility functions [Shenker 1994].

- A wide range of differentiated services, along with a wide range of pricing structures.

- Evaluation: I believe there can be a danger of congestion collapse in this scenario, in the absence of reasonable end-to-end congestion control. Ubiquitous per-flow scheduling provides fairness, but does not prevent the tragedy of the commons.
  - A pricing structure that makes packet drops expensive gives users an incentive to use end-to-end congestion control.

View #4: The virtual circuit view.

- A “virtual-circuit” style of coordination within the network, so that packets don’t enter the global network unless there are reasonable guarantees that they can be delivered to the end receiver.

- With a virtual-circuit model, there is no need for end-to-end congestion control, and no danger of congestion collapse.

- Evaluation: There are many costs of this approach, in terms of tight couplings in a far-flung global Internet, and missed opportunities for the opportunistic use of available bandwidth.

The darker views: Congestion collapse and beyond

- Periodic congestion collapse, because of an uneven use of end-to-end congestion control.

- The “Balkanization” of the Internet on ISP boundaries, resulting in effective congestion control and differentiated services only within ISP boundaries, and degraded performance for traffic that crosses ISP boundaries.

- No coherent global architecture, and therefore missed opportunities (in the development of differentiated services, of multicast capabilities, of coherent web caching architectures, etc.)

- Unrestrained “optimization” at all levels, and between levels, producing greater efficiency in the short term, but rigidity and an inability to accomodate change in the longer term.

- Short-term fixes are deployed, possibly blocking the path for longer-term evolution.

- Inherently difficult traffic patterns?

Conclusions:

- It won’t be boring or easy.
- Many of the current steps are in the right direction.
- The challenge is to keep a coherent and flexible global architecture.