# Thoughts on the Evolution of TCP in the Internet

Sally Floyd

PFLDnet 2004

February 16, 2004

# Overview of talk:

- Past changes to transport protocols.
- Additional possible changes:
  - HighSpeed TCP et al.;
  - Additional feedback from routers?
  - Communication between link layers and transport?
  - Flow-specific state in routers?
- Fundamental limitations:
  - Of window-based congestion control;
  - Of congestion control without per-flow state in routers;
  - Of best-effort service.
- RFC 3426: General Architectural and Policy Considerations

# Past history of TCP

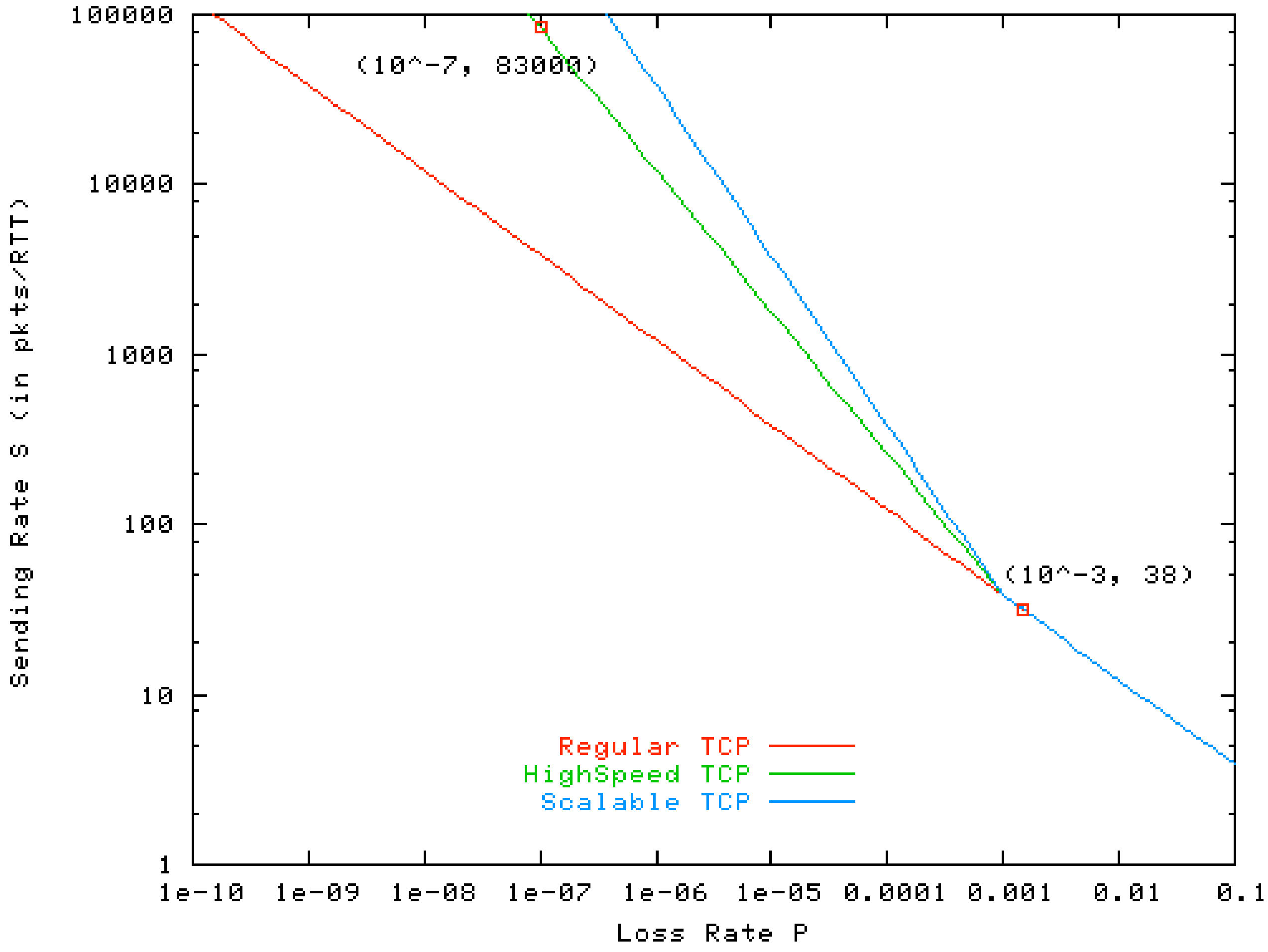
- Reno/NewReno/SACK:
  - Half of servers use SACK, many others use NewReno.
  - Almost all browsers use SACK.
  - DON'T use Reno in simulations or experiments!!!
- Delay-based congestion control:
  - Vegas, FAST
  - TCP-Nice and TCP-LowPriority use delay-based congestion control for low priority TCP .
- ECN:
  - Explicit instead of implicit notification.
  - Standardized but not deployed.

# Past history of TCP

- Quality of Service:
  - Intserv, diffserv, etc.
  - Limited deployment.

- New transport protocols:
  - SCTP: multi-streaming, multihomed transport.
  - DCCP: for unreliable, congestion-controlled transport.

# TCP’s response function



# Convergence Times for HighSpeed TCP et al:

- Different models give different results!
  - Model #1: DropTail queues with global synchronization for loss events.
  - Model #2: Drop Tail queues, some synchronization, depending on traffic mix.
  - Model #3: RED queues, some synchronization.
  - Model #4: RED queues, no synchronization

- Which model is the best fit for the current or future Internet?

# Convergence times for HighSpeed TCP et al:

- Are there modifications to HighSpeed TCP that would improve convergence times?
  - More moderate window increases following loss events?
  - Flow-specific marking or dropping, using flow-specific state?
  - Other?

# Additional Feedback from Routers?

Examples: XCP, QuickStart.

- Explicit feedback from routers would be useful (and necessary) for faster startups.
  - Also for faster recovery after idle periods.
- Per-packet feedback (as in XCP) would give greater power, at greater cost.

# Interactions between transport and link layers:

- Wireless links with variable delay, throughput, etc?

- Hints from transport to link layers, e.g., about robustness to reordering or to delay?

- New issues raised by optical networks, e.g., by optical burst switching?

# Flow-specific state in routers?

- What are the cost/benefit tradeoffs for maintaining state for very large flows?

- Flow-specific marking or dropping, for faster convergence?

- Flow-specific state to help use the bandwidth when a short fat flow ends?
  - (short in time, fat in bytes)

# Fundamental limitations of window-based congestion control?

- The jostling of ACKs can lead to unnecessary burstiness?
  - Rate-based pacing could help.
- Slow start-up?
    - Explicit feedback from routers could help.
- Decreasing the window after a loss event.
  - “Decreasing” does not necessarily require “halving”.
  - Equation-based congestion control (e.g., TFRC) is another alternative.

# Fundamental limitations of no per-flow state in routers?

- For environments with high link utilization, there are limits to faster start-up, and to faster convergence.
  - E.g., a new flow starting up in a high-bandwidth environment with a small number of competing flows.
- For environments with short fat flows, there are limits to link utilization.
  - E.g., many  flows wanting to use high bandwidth, each for a fraction of an RTT.

# Fundamental limitations of best-effort service

- Best-effort flows need to avoid persistent, high drop rates.
- In environments with FIFO scheduling at congested links, best-effort flows need to pay attention to per-flow fairness.
- There are no common assumptions about average or worst-case queueing delay.
- Some flows prefer better-than-best-effort delay, throughput, or loss rates.
- A best-effort flow can’t assume that bandwidth is available when the flow is ready to use it.

# RFC 3426: General Architectural and Policy Considerations

- Justifying the Solution:
  - Why are you proposing this solution, instead of proposing something else, or instead of using existing protocols and procedures?
- Interactions between Layers:
  - Why are you proposing a solution at this layer of the protocol stack, rather than at another layer?
  - Is this an appropriate layer in terms of correctness of function, data integrity, performance, ease of deployment, the diagnosability of failures, and other concerns?
  - What are the interactions between layers, if any?

# Long-term vs. short term.

- Long-term vs. Short-term Solutions:
  - Is this proposal the best long-term solution to the problem?
  - If not, what are the long-term costs of this solution, in terms of restrictions on future development, if any? What are the requirements for the development of longer-term solutions?

# The Whole Picture.

- The Whole Picture vs. Building Blocks:
  - Have you considered the larger context, while appropriately restricting your own design efforts to one part of the whole?
  - Are there parts of the overall solution that will have to be provided by other IETF Working Groups or by other standards bodies?

# Weighing Benefits against Costs.

- EVALUATION QUESTIONS:

- Weighing Benefits against Costs:
  - How do the architectural benefits of a proposed new protocol compare against the architectural costs, if any? Have the architectural costs been carefully considered?

# Robustness

- Robustness:
  - How robust is the protocol, not just to the failure of nodes, but also to compromised or malfunctioning components, imperfect or defective implementations, etc?
  - Does the protocol take into account the realistic conditions of the current or future Internet (e.g., packet drops and packet corruption; packet reordering; asymmetric routing; etc.)?

# Tragedy of the Commons

- Tragedy of the Commons:
  - Is performance still robust if everyone is using this protocol?
  - Are there other potential impacts of widespread deployment that need to be considered?

# Protecting Competing Interests

- Protecting Competing Interests:
  - Does the protocol protect the interests of competing parties (e.g., not only end-users, but also ISPs, router vendors, software vendors, or other parties)?

# Designing for Choice

- Designing for Choice vs. Avoiding Unnecessary Complexity:
  - Is the protocol designed for choice, to allow different players to express their preferences where appropriate?  At the other extreme, does the protocol provide so many choices that it threatens interoperability or introduces other significant problems?

# Preserving Evolvability

- Preserving Evolvability?
  - Does the protocol protect the interests of the future, by preserving the evolvability of the Internet? Does the protocol enable future developments?
  - If an old protocol is overloaded with new functionality, or reused for new purposes, have the possible complexities introduced been taken carefully into account?
  - For a protocol that introduces new complexity to the Internet architecture, how does the protocol add robustness and preserve evolvability, and how does it also introduce new fragilities to the system?

# RFC 3426: Deployment

- DEPLOYMENT QUESTIONS:
  - Is the protocol deployable?