Does current Internet Transport work over Wireless?
Reviewing the status of IETF work in this area

Sally Floyd

March 2, 2000

IAB Workshop on Wireless Internetworking

Observations:

- Transport protocols have to give acceptable performance over a wide range of link-level technologies and underlying routing behaviors.

- TCP changes over time (e.g., SACK), and new transport protocols will emerge.

- Explicit communication between layers (e.g., application, transport, network, and link-level) can be useful.

IETF transport documents:

- RFC 2488, Enhancing TCP Over Satellite Channels...
  - RFC 2760, Ongoing TCP Research Related to Satellites
- An Extension to the Selective Acknowledgement Option for TCP
- SIGTRAN, Framework for SIGTRAN Common Transport Protocol
- End-to-end Performance Implications of Slow Links
  - End-to-end Performance Implications of Links with Errors
  - TCP Performance Implications of Network Asymmetry
  - Performance Enhancing Proxies
  - Advice for Internet Subnetwork Designers:

RFC 2488: Enhancing TCP Over Satellite Channels using Standard Mechanisms.

- Path MTU Discovery.

- Forward error correction (FEC).

(Comment from RFC 2488: The interaction between link-level retransmission and transport-level retransmission is not well-understood.)

- TCP Large Windows.

(For a TCP window larger than 64KB).

- SACK TCP (Selective Acknowledgements).

RFC 2018, Proposed.

RFC 2760, Ongoing TCP Research Related to Satellites

- NewReno TCP

(RFC 2581, Proposed; RFC 2582, Experimental);

  – SACK, FACK with Rate-Halving.

- ECN (Explicit Congestion Notification).

RFC 1482, Experimental.

- TCP larger initial windows.
  - One or two packets (RFC 2581, Proposed).
  - Possibly three or four packets, depending on packet size (RFC 2414, Experimental).

RFC 2760, Ongoing TCP Research Related to Satellites

- Explicit Corruption Notification?
  - Link-level detection and retransmission.
  - ICMP ”corruption experienced” error messages.

- ACK congestion control or ACK filtering, for low-speed return paths.

- Changes to slow-start:
  - Byte-counting instead of packet-counting?
  - Use delayed ACKs only after slow-start is over?
  - Terminating slow-start early?

RFC 2760, Ongoing TCP Research Related to Satellites

- T/TCP (TCP for transactions)?

- Sharing TCP State among Similar Connections?

- Changes to TCP's window increase policy?
(to change the bias against longer round-trip times)

- TCP header compression.

- Rate-based pacing.

An Extension to the Selective Acknowledgement (SACK) Option for TCP

- "This note extends RFC 2018 by specifying the use of the SACK option for acknowledging duplicate packets."

- "A TCP sender could then use this information for more robust operation in an environment of reordered packets, ACK loss, packet replication, and/or early retransmit timeouts."

- draft-floyd-sack-00.txt

SIGTRAN - Reliable UDP Protocol

- Designed for telecommunication signaling protocols.
- Supports persistent associations, in-order delivery within a control stream.
- – No head-of-line blocking;
  – keep-alive for rapid detection of session failure;
  – failure to backup session;
  – limited number of attempts at retransmissions;
  – tighter retransmit time-outs than TCP.
  – Nagle algorithm might be turned off.
- draft-ietf-sigtran-common-transport-00.txt

Transport for Unreliable, Unicast Streaming Multimedia

- Intended for flows that are willing to use TCP-compatible end-to-end congestion control, but would prefer not to reduce their sending rate in half in response to a single packet drop.

- http://www.aciri.org/tfrc/

End-to-end Performance Implications of Links with Errors

- Proposals for Explicit Corruption Notification:
  - Explicit Loss Notification (ELN) [BPSK96]
  - Explicit Bad State Notification (EBSN) [BBKVP96]
  - Explicit Loss Notification to the Receiver (ELNR), Explicit Delayed Dupack Activation Notification (EDDAN) [MV97]
  - Explicit ”negative acknowledgements” to notify the sender that a damaged packet has been received (SCPS-TP)

- ECN (Explicit Congestion Notification) does not eliminate the need for Explicit Corruption Notification.

- draft-ietf-pilc-error-02.txt

TCP Performance Implications of Network Asymmetry

●"This document describes the problems to TCP performance that arise because of asymmetric effects."

●"Solutions to the problem of asymmetry are two-pronged:
(i) techniques to manage the reverse channel used by ACKs, typically using header compression or reducing the frequency of TCP ACKs, and
(ii) techniques to handle this reduced ACK frequency to retain the TCP sender's acknowledgment-triggered self-clocking."

Performance Enhancing Proxies

- “A Performance Enhancing Proxy (PEP) is used to improve the performance of the Internet protocols on network paths where native performance suffers due to characteristic of a link or subnetwork on the path.”

- Transport Layer PEPs:
  - Modify TCP ACK spacing;
  - Generate local TCP acknowledgements;
  - Local TCP retransmissions;
  - Split connection TCP

- Application Layer PEPs:

Performance Enhancing Proxies

- Transparency: the degree of transparency may vary (e.g., transparency to end systems, transport endpoints, applications, or users).

- Other functions of PEPs:
  - Compression;
  - Handling periods of link outage;
  - Priority-based multiplexing;

Performance Enhancing Proxies: Specific environments for PEPs:

- Satellite very small aperture terminal (VSAT) environments
  - TCP PEPs for improving TCP performance, with compression and split connections.

- Mobile wireless WAN (W-WAN) environments
  - variable queueing delays, intermittent link outages,
  - typically the last-hop link to the user.

- Wireless LAN (W-LAN) environments
  - a base station controls a single cell.
  - mobile hosts move from one cell to another.
  - link corruption.
  - PEPs: Berkeley's Snoop protocol.

Performance Enhancing Proxies: Implications of PEPs:

- Maintaining end-to-end semantics:
  - Security (IPsec);
  - Fate-sharing, so that a connection does not depend unnecessarily on state stored in the network;
  - End-to-end reliability;
  - End-to-end failure diagnostics;
  - Requires use of symmetric routing?
  - State handovers for mobile hosts.

- draft-ietf-pilc-pep-01.txt

End-to-end Performance Implications of Slow Links

- Recommends:
  - Header compression, payload compression.
  - MTU sizes that don’t monopolize the link for too long.
  - The TCP receiver limits the receive buffer size, if the host ”knows” it is directly connected to a slow link.
  - Sending new data when a single dup ack is received.

- Suggests:
  - TCP buffer auto-tuning.

- draft-ietf-pilc-slow-02.txt

Congestion collapse

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

- Congestion collapse from undelivered packets: 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.

Research Issues:

- Protection against misbehaving TCP receivers.
- Network protection against misbehaving flows.

Questions that I did not answer:

- The CPU, power consumption, memory, and/or packet header overhead of TCP?
- Mobility and TCP?
- Quality of service?
- ...

Advice for Internet Subnetwork Designers:

- Connection-Oriented Subnetworks

“The ideal subnetwork for IP is connectionless.”

- Reliability and Error Control

“Subnet reliability should be ”lightweight”, i.e., it only has to be ”good enough”, *not* perfect.”

- Compression:

“User data compression is a function that can usually be omitted at the subnetwork layer.”

- Packet Reordering:

“We recommend that subnetworks not gratuitously deliver packets out of sequence.”

- Bandwidth Asymmetries
- Maximum Transmission Units (MTUs) and IP Fragmentation
- Framing on Connection-Oriented Subnetworks
- Bandwidth on Demand (BoD) Subnets
- draft-ietf-pilc-link-design-01.txt