Near-Earth Wireless (Satellite) Issues

Mark Allman
NASA GRC/BBN Technologies
mallman@grc.nasa.gov
http://roland.grc.nasa.gov/~mallman/

IAB Wireless Internetworking Workshop

March 2000

Overview

- What are we trying to do?
- Link characteristics
- Solved problems
- Unsolved problems
  - Including half-baked (if that) ideas.

What Are We Trying to Do?

- Commercial satellite folks want to offer high-bandwidth Internet service over LEO and GEO satellite links.
  - To people's home
  - To remote areas of the world
  - To places where there is no terrestrial infrastructure
    - Caterpillar wants to use satellites to communicate between its vehicles and its service people
    - There is no infrastructure where Caterpillars are!

What Are We Trying to Do? (cont.)

- What NASA wants:
  - To put a web server on the space station
  - To use Internet applications to facilitate communication between space and the ground
    - Astronauts can send email to their families
    - We can offer real-time video from shuttle
    - Etc.
  - To communicate with Earth-observing satellites
    - E.g., to transfer data from monitoring equipment in the ocean.
  - And, to do it all in a standard way (i.e., using commercial products without developing specialized solutions)

What Are We Trying to Do? (cont.)

- Our focus has been on transport and application layer challenges.
- Routing problems have not been an issue yet.
  - But, that may very well change...

Link Characteristics

- Links have large propagation delays, but not too long (i.e., communication to Mars is not being considered)
- Links have a non-zero bit-error rate
- Some hosts we would like to communicate with are moving (e.g., space station)
- Moving end-hosts will sometimes have to use different communications links (i.e., we have handoffs)
- A large range of bandwidth (from very small to quite large).

Solved Problems

- We are stuck with long links
  - Long links require big congestion windows, but that has been fixed (RFC 1323)
  - With big windows may come lots of loss, which can be dealt with by using SACK (RFC 2018) or NewReno if SACK is not available (RFC 2582)
- Have recommended cleaning up the noise on links as much as possible with FEC (RFC 2488)
  - SACK should help with remaining losses
- ECN (RFC 2481) helps indicate congestion without dropping segments (especially helpful for interactive and request/response applications over long delay links).

Unsolved Problems

- Autotuned end hosts
  - Would like to see autotuned socket buffers so experts are not needed for end hosts to be able to effectively cope with the long delay [SMM98]
    - Largely a TCP implementation issue.
  - To truly do autotuning today RFC 1323 would need to be on by default:
    - I.e., we want to ability to use large windows if the network can support them.
    - But, we don't want to waste the bits RFC 1323 requires on low bandwidth links.
  - It *might* be nice to be able to enable "large window extensions" in the middle of a connection.

Unsolved Problems (cont.)

- Explicit *Corruption* Notification
  - It would be nice if we had some way to tell the difference between a congestion induced loss and a corruption-based loss.
  - Perhaps a message sent to the originator of the packet when the transport checksum fails?
    - What if the network layer checksum fails? Who gets the corruption message?
  - Some work in this area has been done (see RFC 2760 for an overview)
    - Is it enough?
    - Do we understand the problem and the implications?

Unsolved Problems (cont.)

- Bias against long-delay connections
  - Connections with long RTTs are at a disadvantage when competing with connections with shorter RTTs and end up using less than their "fair share" of the bandwidth [Flo91,HSMK98].
  - Henderson [HSMK98] has suggested a slightly different congestion avoidance mechanism to help eliminate this unfairness.
  - Is some sort of per-flow queue needed to help long-delay flows achieve their "fair share"?
  - Are there other ways?

Unsolved Problems (cont.)

- Slow start is still slow and underutilizes capacity.
  - Lots of connections never leave slow start.
  - Larger initial windows (RFC 2414) help (especially for short transfers)
  - Some form of byte counting [All98,All99] *might* alleviate the problems caused by delayed ACKs.
  - Use bandwidth estimation (ala packet-pair) to increase *cwnd* more rapidly based on the bandwidth estimate and the RTT
    - However, bandwidth estimation doesn't seem to work all that well "in the wild" [AP99]

Unsolved Problems (cont.)

- Big windows cause big bursts.
  - The routers along a network path containing a long-delay link need to be equipped with big queues.
  - Unless we use some form of pacing to smooth out some of the bursts [KCRP99].
  - Pacing is still being studied – it may not be as appealing as initially thought [ASA00].

Unsolved Problems (cont.)

- TCP and mobility.
  - Will TCP tolerate modestly variable propagation delays?
    - Probably.
  - What happens to TCP connections after a handoff (in which there could be lost or duplicated segments)?
  - What will happen when TCP encounters an outage?

Unsolved Problems (cont.)

- TCP and mobility (cont.)
  - Should there be explicit messages that tell TCP the connection is using a different path?
    - I.e., could put TCP "to sleep" and make it wakeup "later" for an outage.
    - We could make TCP slow start after a handoff given that its parameters may be inappropriate for the new path conditions.
  - Should TCP try to infer this information? How?

Unsolved Problems (cont.)

- Network Layer Problems:
  - Routing *might* get ugly when things are moving.

Continuing Struggles

- To the extent possible we'd like to see application layer protocols that are not excessively "chatty" since chatting takes more time in long-delay networks.

References

[All98] Mark Allman. *On the Generation and Use of TCP Acknowledgments*. ACM Computer Communication Review, 28(5), October 1998.

[All99] Mark Allman. *TCP Byte Counting Refinements*. ACM Computer Communication Review, 29(3), July 1999.

[AP99] Mark Allman, Vern Paxson. *On Estimating End-to-End Network Path Properties*. ACM SIGCOMM, September 1999, Cambridge, MA.

[ASA00] Amit Aggarwal, Stefan Savage, Tom Anderson. *Understanding the Performance of TCP Pacing*. Proceedings of IEEE Infocom. March, 2000.

[Flo91] Sally Floyd. *Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic*. ACM Computer Communication Review, 21(5), October 1991.

[HSMK98] Tom Henerson, Emile Sahouria, Steven McCanne, Randy Katz. *On Improving the Fairness of TCP Congestion Avoidance*. Proceedings of IEEE Globecom. November, 1998.

[KCRP99] Joanna Kulik, Robert Coulter, Dennis Rockwell, Craig Partridge. *Paced TCP for High Delay-Bandwidth Networks*. Proceedings of IEEE Globecom. December, 1999.

[SMM98] Jeff Semke, Jamshid Mahdavi, Matt Mathis. *Automatic TCP Buffer Tuning*. ACM SIGCOMM, September 1998.