Transport Protocol Research after 35 Years

Dr. Craig Partridge
BBN Technologies

Overview

- Context
  - Transport protocols to 1983
  - Transport protocols 1983-1993
  - Transport protocols 1993-now
- Themes
- Case Studies
- Some Vague Thoughts

Transport Protocols to 1983

- The idea of a transport protocol was created in a hallway at USC-ISI in late 1977
  - When TCP and IP were split to enable UDP
  - But research started somewhat earlier
- Hard to realize now how simple the environment was
  - 56Kbps links were FAST!
    - There was ARPANET, Ethernet and a bunch of specialized network technologies
  - You could literally watch the packet headers go past
  - Until 1983 nobody had a personal computer
  - We timeshared 1 MIP machines (~1 MHZ)
  - An entire OS fit in 32KB and a big application was about 64KB

The Shakedown Cruise 1983-1993

- The Internet gets turned on in 1983…
- … and we enter roughly a decade of fixing the Internet while it grows explosively
- For most of this time
  - 56Kbps was still fast for long haul
  - Ethernet was coax (you drilled into the copper to attach a tap) and shared (not switched)
  - We had workstations of about 1-2 MIPs and about 2 MB of memory
  - Routers had about the processing power of an 80186 and 250KB of memory or less
  - There are competing technologies (DECNET and OSI)
- The congestion storms were awful
  - More than once a key link found it had more TCP connections than packets in flight (cnwd << 1MSS)

1993 – Now: Performance & Security!!

- By 1993, the gigabit testbed program had brought us much faster links (155 or 622Mbps was typical trunk speed)
  - 10-100 Mbps switched Ethernet
- Laptops had more power than the machines we timeshared 10 years before
- Everyone is on the Internet
  - Great motivation to cause disruption
  - Everyone wants to be safe (and yet talk to anyone)

Broad Comments

- Research trajectories vary but, at least for transport, seem to fall into four groups:
- One and done – someone tackles and problem and solves it the first time
- A problem is posed and a flurry of activity yields a result over a few years
- A problem keeps returning because the environment changes
- A problem festers because we either lack the answer or don't like the answers

One and Done

- The first one there solves the problem so well we don't need to go back
- IP fragmentation a mistake (Mogul/Kent '88)
  - Transport protocol must understand path MTU
  - Path MTU spec
- Jacobson-Floyd on Timer Synchronization
  - (are routing protocols transport protocols?)
  - Periodic data transfer needs protection against sync

A Problem Gets Solved Over Several Years

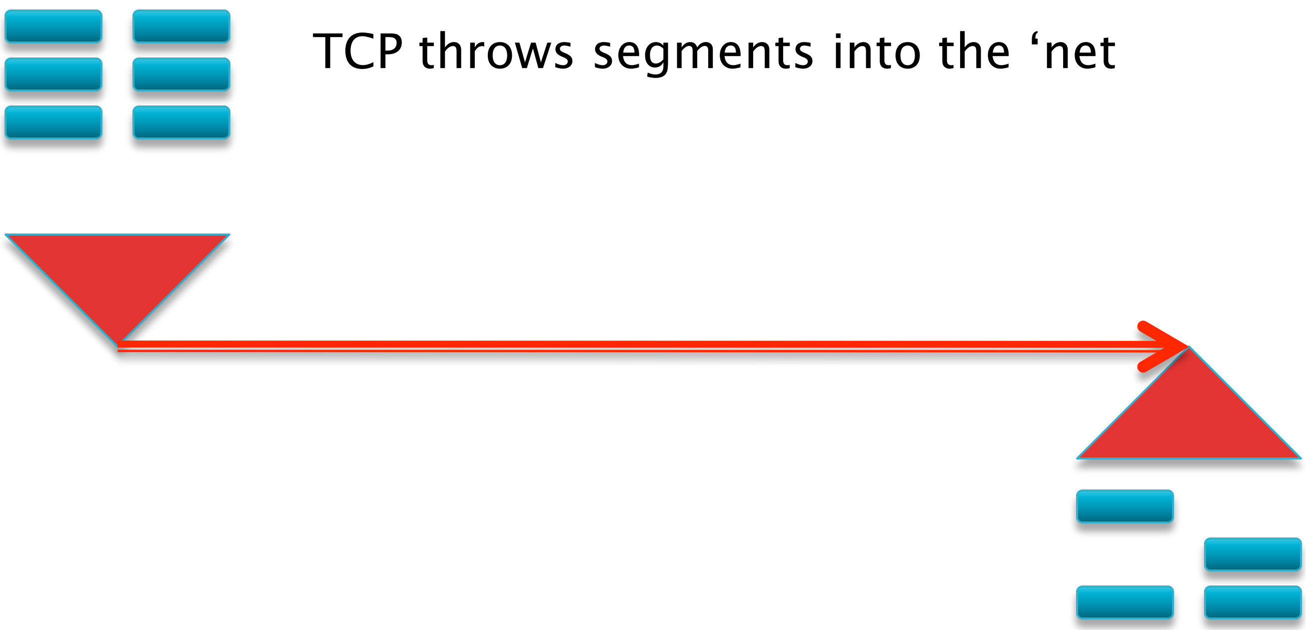
- Someone poses the problem and, typically, makes progress
- Over the next several years, others chime in
- After some time, the problem space is mapped and consensus on solutions
- Perhaps “typical” research trajectory
  - Certainly “classic” research trajectory

Sequence Numbers & Connection Establishment

- The first research problem hit in 1974
- Ray Tomlinson implemented TCP from the Cerf&Kahn paper and ran some tests sending files to a printer
  - He got printouts that mixed text from multiple files
  - Each connection started with the same port & sequence number and segments were getting confused
- Ray invented the 3-way handshake and rules for choosing initial sequence numbers (1975)
- Carl Sunshine refines Ray's ideas
- Dick Watson shows how to skip 3-way handshake using timers (1981)

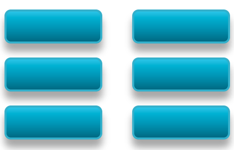
TCP interacts with the ‘net – 1983

TCP throws segments into the ‘net

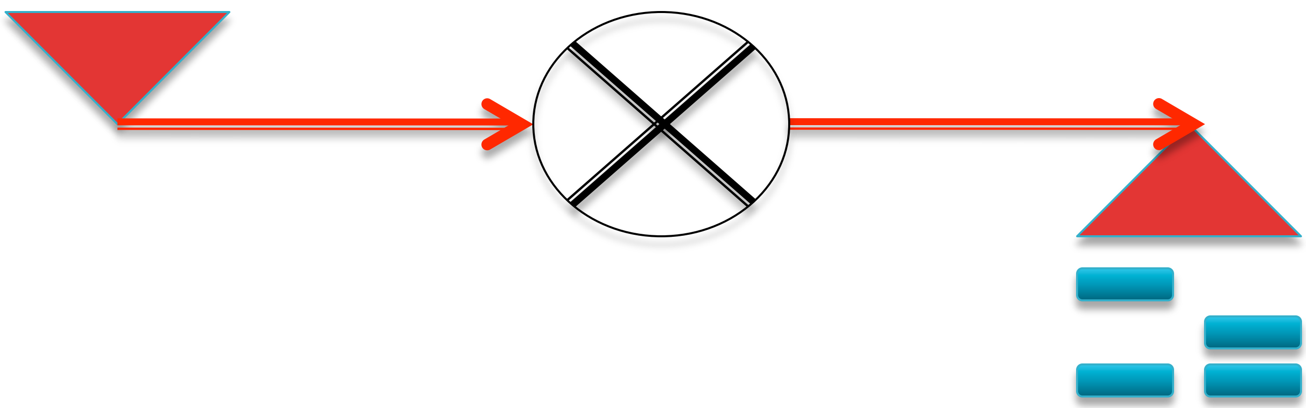


Some segments come out the other end;
Sender uses timeouts to repair losses

TCP interacts with the 'net – Nagle/Postel

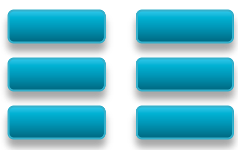


TCP throws segments into the 'net;
Nagle: there's a router in the middle with limited packet rate. Minimize small segments. (1984)

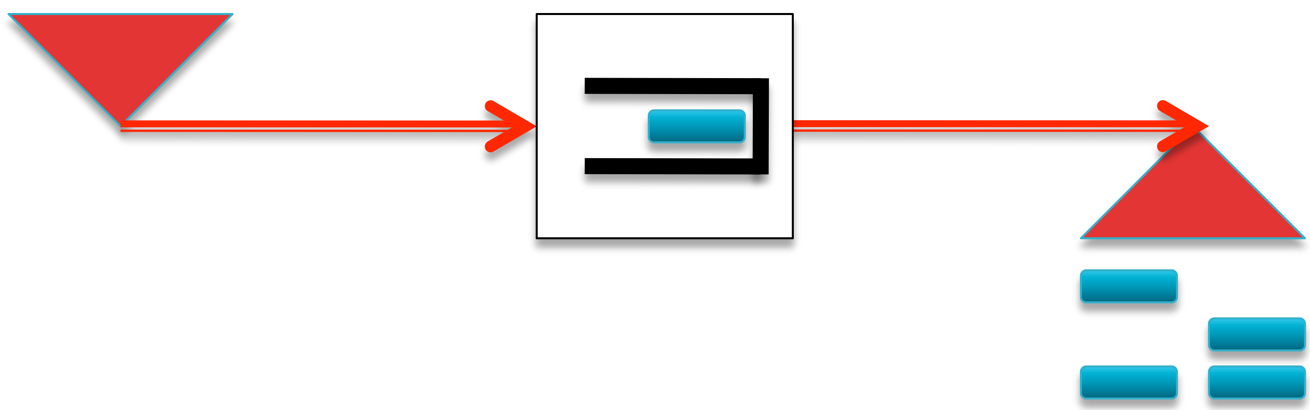


Some segments come out the other end;
Sender uses timeouts to repair losses.
Postel: only retransmit segment you know is lost!

TCP interacts with the 'net – Nagle/Jain+Ramakrishan/Jacobson



TCP throws segments into the 'net;
Nagle: router has a queue and making it bigger doesn't solve congestion ('85)
J+R: if router tells you it is congested, endpoint can respond ('88)



Some segments come out the other end;
~~Sender uses timeouts to repair losses.~~
Jacobson: use loss to learn of congestion and use additive increase/multiplicative decrease + fast retransmit based on dupe acks ('88)

TCP Timers

- Ample evidence by late 1985 that round-trip time estimation is often wildly wrong
  - Estimates too low (spurious retransmissions)
  - Estimates too high (connections paused for ages)
- Jain (early ‘86) -> retransmission ambiguity
  - Always err towards high estimate
- Zhang (late ‘86) -> retransmission ambiguity is an impossible problem
- Karn & Partridge (‘87) -> retransmission ambiguity has a solution, and Karn’s algorithm is in the solution space
- Jacobson (‘88) -> must also fix the RTT estimator
- Jacobson/Braden (‘88) -> even better, put time stamps in the segments and get rid of ambiguity

Problems That Return – but we're always improving

- The classic one is implementation
  - Challenged to make transport protocols more efficient and faster as each new generation of links and mid-points and end-points arrive

Implementation Woes to 1983

- Fitting TCP into a small code footprint in an early operating system….
  - Series of papers by Dave Clark (RFCs 814 & 817, SOSP 1985 paper on upcalls)
  - Plummer's notes on checksum implementation and easy sequence number addition
- All sorts of problems that need a solution
  - Pick a checksum
  - Pick a round-trip time estimation algorithm
  - Segment boundaries soft or hard?
  - Silly Window Syndrome

Implementation (1993–Now)

- In implementation we were ahead of the curve by 1993
  - Jacobson – 10 Mbps TCP on 1 MIPS machine (‘88) using single copy and header prediction
  - Borman – 600 Mbps TCP (‘89) “FDDI there we went..”
  - Partridge/Pink – single copy works for UDP (‘91)
- Except for parallel/multiprocessor systems
  - Thankfully, multiprocessor computers died c. 1995
  - But they’re coming back…
- Cautionary note: Offload Engines
  - A dumb idea that keeps coming back

Problems that keep returning – and we don't like the answers

- TCP over high delay*bandwidth paths
- Transport's version of the QoS problem?

Improving TCP record keeping

- Do a better job of tracking what's in flight and fixing problems
- Selective acknowledgments (Floyd c. '92)
  - Based on idea in RDP by Hinden/Rosen ('84) tested by Partridge ('87)
  - Enhanced by Forward Acknowledgement (Mathis/Mahdavi '96)
- Also allows us to be courageous and let TCP send more than one segment to start

TCP Over Large Delay*Bandwidth paths

- Many innovations that do poorly
  - Alter the basic additive/multiplicative model
  - Pacing
  - Distinguish loss from congestion
- Much very good work to model TCP-friendly
  - Kurose & Mathis & Floyd come to mind

Some Vague Thoughts

- When I started writing this talk
  - I thought most research projects followed "classic path"
  - I thought most research took several years and multiple contributors each added a lot
- At least for transport, that's not true
  - Most times, the first paper gets it 90% right
  - Even on hard problems, things often wrap up in 2-3 years
- A lot of transport research has been filling in the blanks from 1977
  - Rather sobering…