QUIC in Space

Draft-huitema-quic-in-space

Motivation and early results

- Marc asked me, could QUIC work on long delay space links?
  - Define long delay, define space. The Moon is easy!
  - No, not the Moon, more like Mars, one way delays between a few minutes and 20 minutes?
- Quickly whipped up a few simulations. It works!
  - Need only on parameter, the maximum duration of the handshake
  - If the handshake succeeds, the RTT gets configured, and “everything” works
- Really? Define “everything”!
  - Planets rotate! Intermittent connectivity, 12 hours on, 12 hours off?
  - Do we need to change congestion control?
  - What else?

Intermittent connectivity

- Suppose a station on Mars.
  - Mars day in about 24 hour 39 min
  - Station will only see Earth for about 12 hours each day.
- Does it work? Kinda, not well
  - Use timeout to discover breakage
  - QUIC doubles the timeout after each repeat attempt (RFC 9002)
  - Eventually something arrives…
  - But way after start of the new “sol”
- TODO: solve that
  - Limit exponential backoff?
  - Get external signal?

RTT	Iteration	Timeout (s)	Time elapsed (hours)
12000	0	1200	0.33
12000	1	1200	0.67
12000	2	2400	1.33
12000	3	4800	2.67
12000	4	9600	5.33
12000	5	19200	10.67
12000	6	38400	21.33
12000	7	76800	42.67

Do we need long duration connections?

- Connection contexts will be lost
  - Especially if end-to-end between workstation and space craft
  - Workstation might loose power, reboot, perform software update…
- QUIC deployments rely on fast connection resume
  - After long silence, state of connection is dubious
  - Better to just restart
- Might help with intermittent connectivity
  - Workstation knows the time when connectivity is restored
  - Can schedule “resumed” connection at that time
- Need work: support for 0-RTT, remember BDP, etc.

What about congestion control

- Tests show BBR works as expected
  - Everything just scales with the RTT
  - Only time constant is the 10 second “probe RTT” period
  - But it is easy to tune that, e.g. “max of 10 seconds, 10 RTT”
- RENO plain does not work
  - Congestion avoidance phase, CWIN grows at most one packet per RTT
  - Way too slow for deep space
- Cubic also does not work, for complex reasons
  - Pseudo period “K” depends on Wmax, does not depend on RTT
  - K = cubic_root((Wmax - cwnd(epoch))/C),
  - Wmax = BDP, cwnd = 0.7 Wmax, C = 0.4

Cubic to Mars: K lower than RTT !

C	Beta	Packet size (bytes)	RTT (sec)	Data rate (Mbps)	BDP (packets)	K (sec)
0.4	0.7	1500	0.1	1	8	2
0.4	0.7	1500	0.1	10	83	4
0.4	0.7	1500	2	1	167	5
0.4	0.7	1500	2	10	1667	11
0.4	0.7	1500	2	100	16667	23
0.4	0.7	1500	600	1	50000	33
0.4	0.7	1500	600	10	500000	72
0.4	0.7	1500	600	100	5000000	155
0.4	0.7	1500	1200	0.1	10000	20
0.4	0.7	1500	1200	1	100000	42
0.4	0.7	1500	1200	10	1000000	91
0.4	0.7	1500	1200	100	10000000	196

- Cubic formula:
  - $W = C*(t-K)^3 – Wmax$
- After 1200 seconds, with K=196:
  - $W = 41.5 * Wmax$
- Most likely:
  - All buffers filled before any feedback arrives
  - Massive packet losses

Next steps

- Investigate the “intermittent link” issues
  - Build simulations, tune delays, etc.
  - Model possible “network support”, or maybe “wake on packet”
- Investigate “QUIC resume”
  - Does 0-RTT work, how long to ramp up CWIND, etc.
- Look at other issues, e.g., variable delays
- Do end to end simulation with realistic apps