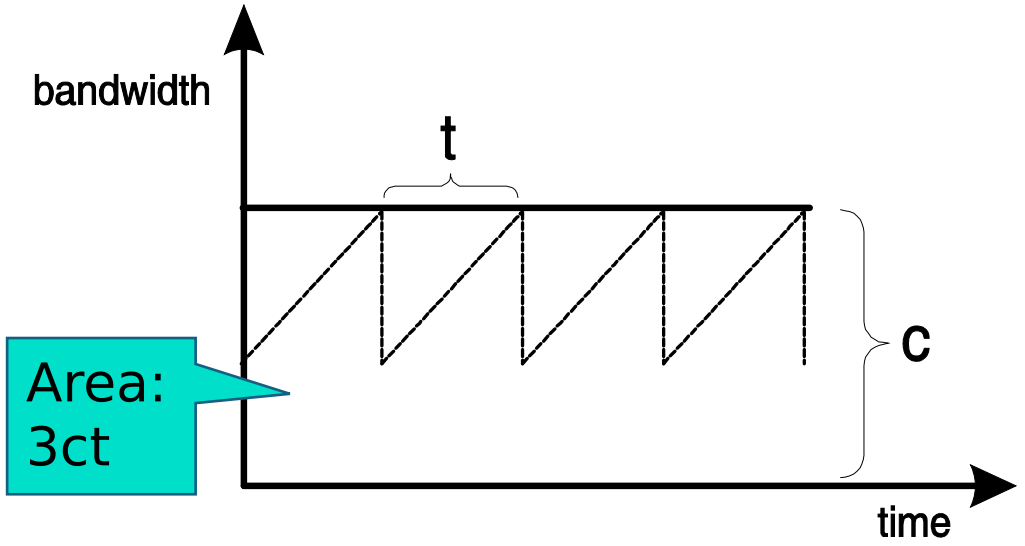
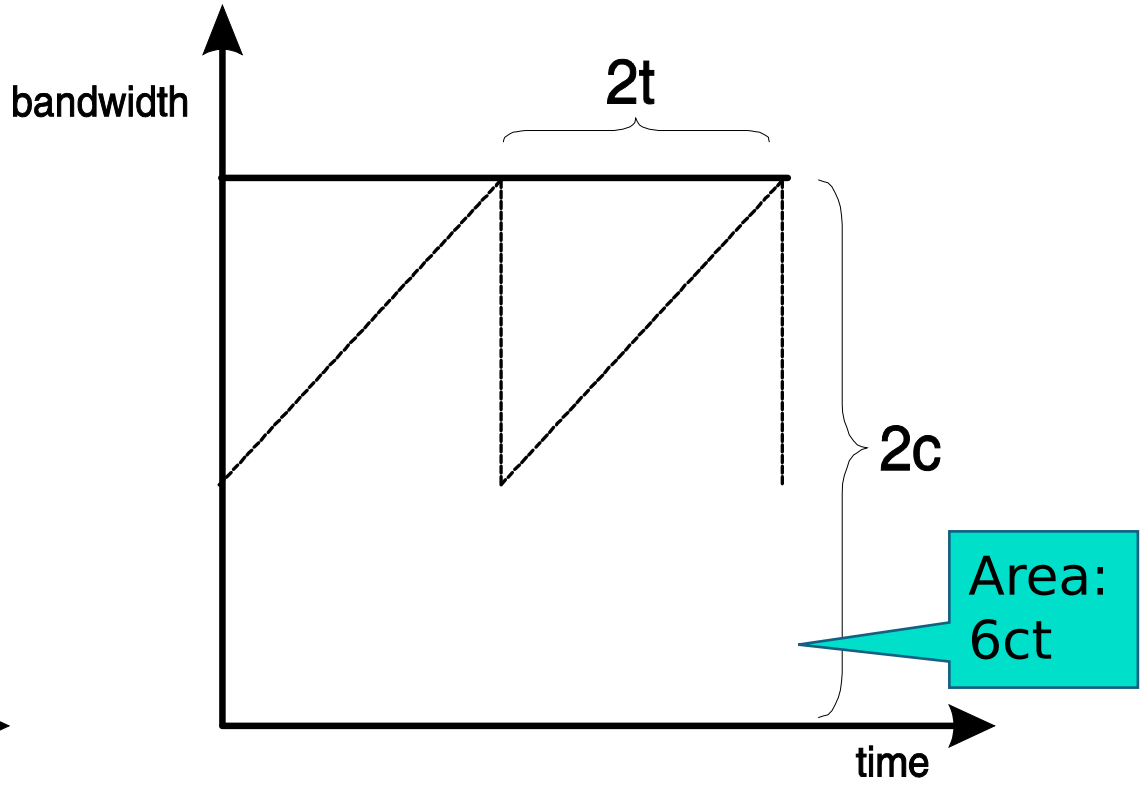
# Congestion Control for Long Fat Pipes: the State of the Art

Michael Welzl

HP-WAN BoF
IETF 121

# Early 2000s problem: TCP with a "Large Fat Pipe"

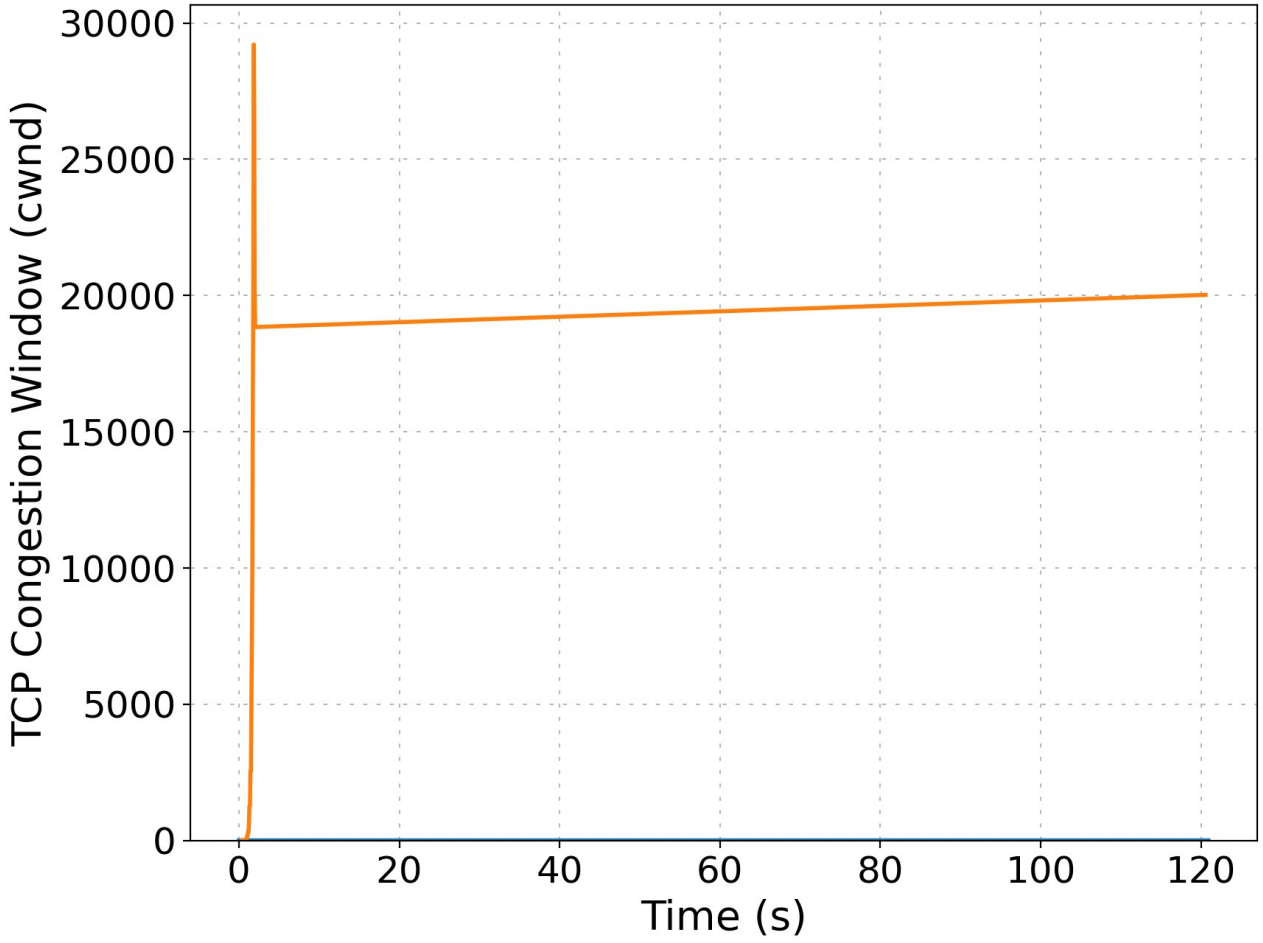



- Theoretically, utilization independent of capacity
- But: longer convergence time

# A testbed experiment

- Diagram shows TCP Reno,
  3 Gbps, 100ms RTT
- 100ms is too long?
  3 Gbps not enough?
- Ok, consider:
  10ms RTT, 30 Gbps
  - This is the same Bandwidth * Delay Product (BDP)

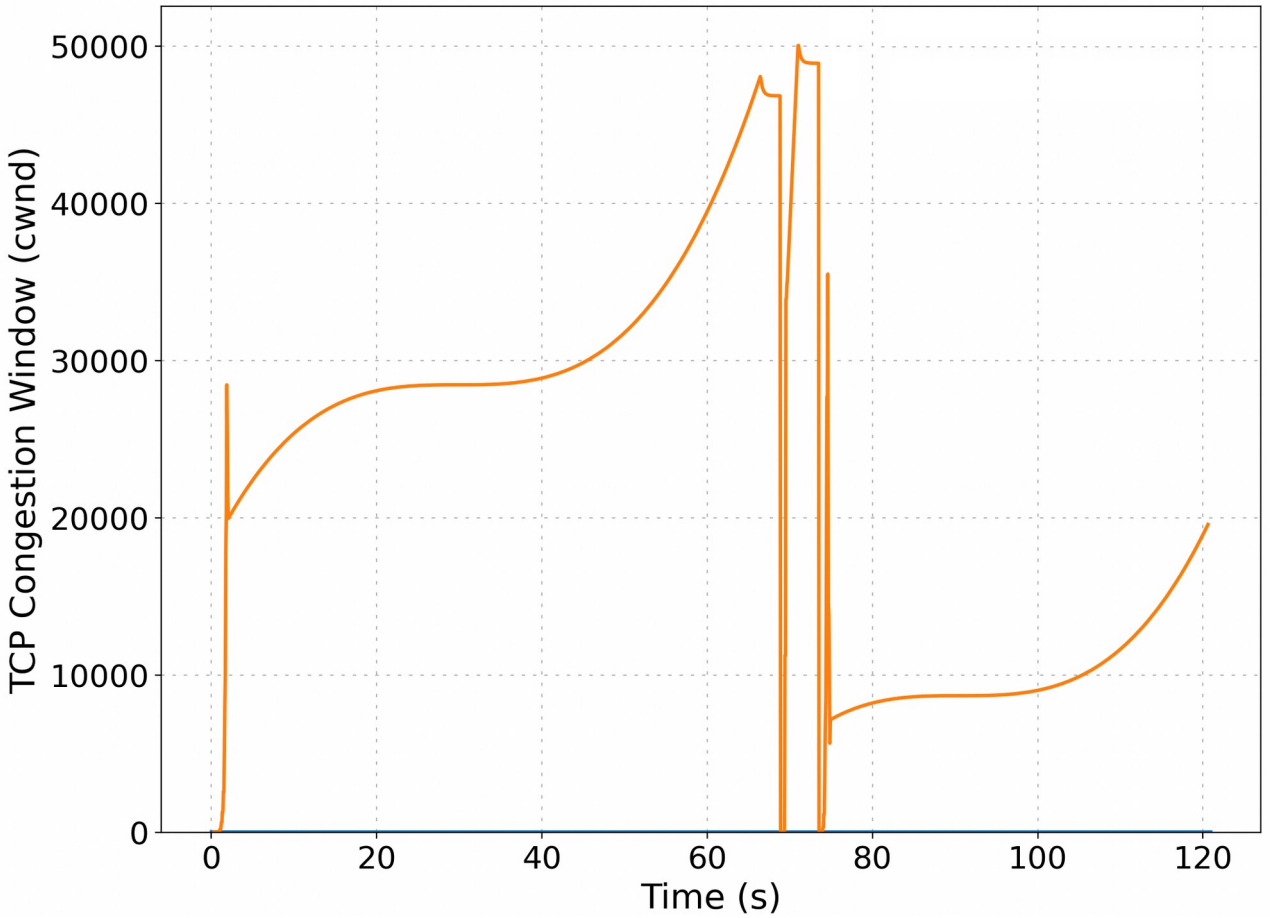


# Consequences

1. Only huge transfers yield a representative long-term average behavior
2. We need **ultra-low packet loss**

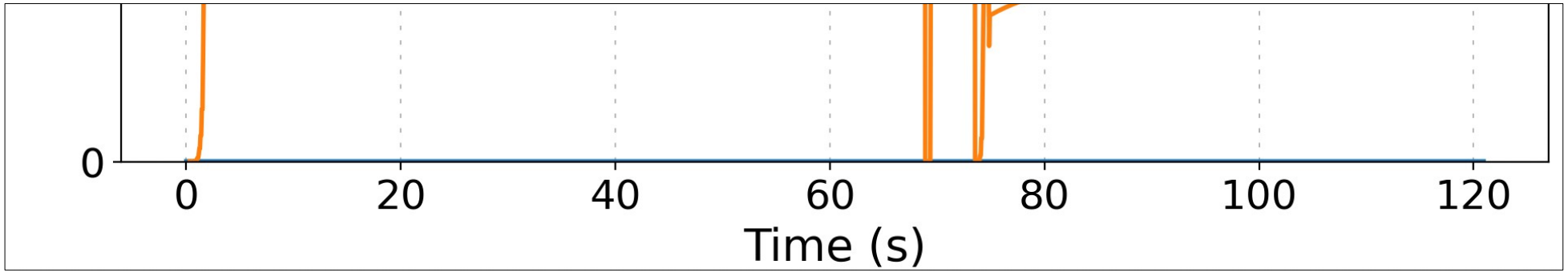
- From RFC 3649 (HighSpeed RFC, Experimental):
  *"For example, for a Standard TCP connection with 1500-byte packets and a 100 ms round-trip time, achieving a steady-state throughput of 10 Gbps would require an average congestion window of 83,333 segments, and a packet drop rate of at most one congestion event every 5,000,000,000 packets (or equivalently, at most one congestion event every 1 2/3 hours). This is widely acknowledged as an unrealistic constraint."*

# Luckily, we got Cubic...




- Looks better, right?
- Something strange happened at around t=70... overshoot, not really able to repair...
  - This is Linux, Debian kernel version 6.1.0
  - Maybe the queue was too large, or too short... let's say: we just got unlucky.
  - **Solution: run it longer. On average, it'll be okay.**

# Longer?




**Zoom into the previous diagram: 2 minutes!**

- E.g., only 1 Gbps: approx. 125 Mbyte / second.
  - 2 minutes: approx. 15 Gbyte

# An excerpt, in numbers

[ 5] 65.00-66.00 sec 295 MBytes 2.48 Gbits/sec
[ 5] 66.00-67.00 sec 12.7 KBytes 104 Kbits/sec
[ 5] 67.00-68.00 sec 14.1 KBytes 116 Kbits/sec
[ 5] 68.00-69.00 sec 53.3 MBytes 447 Mbits/sec
[ 5] 69.00-70.00 sec 292 MBytes 2.45 Gbits/sec
[ 5] 70.00-71.00 sec 157 MBytes 1.32 Gbits/sec
[ 5] 71.00-72.00 sec 14.1 KBytes 116 Kbits/sec
[ 5] 72.00-73.00 sec 8.48 KBytes 69.5 Kbits/sec
[ 5] 73.00-74.00 sec 38.2 MBytes 320 Mbits/sec
[ 5] 74.00-75.00 sec 138 MBytes 1.16 Gbits/sec
[ 5] 75.00-76.00 sec 102 MBytes 857 Mbits/sec
[ 5] 76.00-77.00 sec 105 MBytes 884 Mbits/sec
[ 5] 77.00-78.00 sec 108 MBytes 903 Mbits/sec
[ 5] 78.00-79.00 sec 110 MBytes 924 Mbits/sec
[ 5] 79.00-80.00 sec 113 MBytes 944 Mbits/sec
[ 5] 80.00-81.00 sec 114 MBytes 960 Mbits/sec
[ 5] 81.00-82.00 sec 116 MBytes 973 Mbits/sec
[ 5] 82.00-83.00 sec 117 MBytes 982 Mbits/sec
[ 5] 83.00-84.00 sec 118 MBytes 988 Mbits/sec
[ 5] 84.00-85.00 sec 119 MBytes 994 Mbits/sec
[ 5] 85.00-86.00 sec 119 MBytes 997 Mbits/sec
[ 5] 86.00-87.00 sec 119 MBytes 998 Mbits/sec
[ 5] 87.00-88.00 sec 119 MBytes 999 Mbits/sec
[ 5] 88.00-89.00 sec 119 MBytes 999 Mbits/sec
[ 5] 89.00-90.00 sec 119 MBytes 999 Mbits/sec
[ 5] 90.00-91.00 sec 119 MBytes 998 Mbits/sec
[ 5] 91.00-92.00 sec 119 MBytes 998 Mbits/sec
[ 5] 92.00-93.00 sec 119 MBytes 1000 Mbits/sec
[ 5] 93.00-94.00 sec 119 MBytes 1.00 Gbits/sec
[ 5] 94.00-95.00 sec 120 MBytes 1.00 Gbits/sec
[ 5] 95.00-96.00 sec 120 MBytes 1.01 Gbits/sec
[ 5] 96.00-97.00 sec 121 MBytes 1.01 Gbits/sec
[ 5] 97.00-98.00 sec 122 MBytes 1.02 Gbits/sec
[ 5] 98.00-99.00 sec 123 MBytes 1.03 Gbits/sec
[ 5] 99.00-100.00 sec 124 MBytes 1.04 Gbits/sec
[ 5] 100.00-101.00 sec 125 MBytes 1.05 Gbits/sec
[ 5] 101.00-102.00 sec 127 MBytes 1.07 Gbits/sec
[ 5] 102.00-103.00 sec 130 MBytes 1.09 Gbits/sec
[ 5] 103.00-104.00 sec 133 MBytes 1.11 Gbits/sec
[ 5] 104.00-105.00 sec 136 MBytes 1.14 Gbits/sec
[ 5] 105.00-106.00 sec 140 MBytes 1.17 Gbits/sec
[ 5] 106.00-107.00 sec 144 MBytes 1.21 Gbits/sec
[ 5] 107.00-108.00 sec 149 MBytes 1.25 Gbits/sec
[ 5] 108.00-109.00 sec 154 MBytes 1.29 Gbits/sec
[ 5] 109.00-110.00 sec 159 MBytes 1.34 Gbits/sec
[ 5] 110.00-111.00 sec 166 MBytes 1.39 Gbits/sec
[ 5] 111.00-112.00 sec 173 MBytes 1.45 Gbits/sec
[ 5] 112.00-113.00 sec 181 MBytes 1.52 Gbits/sec
[ 5] 113.00-114.00 sec 190 MBytes 1.59 Gbits/sec
[ 5] 114.00-115.00 sec 200 MBytes 1.67 Gbits/sec
[ 5] 115.00-116.00 sec 210 MBytes 1.76 Gbits/sec
[ 5] 116.00-117.00 sec 221 MBytes 1.86 Gbits/sec
[ 5] 117.00-118.00 sec 233 MBytes 1.96 Gbits/sec
[ 5] 118.00-119.00 sec 246 MBytes 2.07 Gbits/sec
[ 5] 119.00-120.00 sec 260 MBytes 2.18 Gbits/sec
[ 5] 120.00-120.10 sec 26.8 MBytes 2.24 Gbits/sec

# History repeats itself: we need a new CC!

- Cubic reaches its limits
  - It doesn't really scale with the BDP, and it reacts heavily to packet loss

- What we need:
  1. a scalable control
  2. more loss tolerant
  3. yet, backwards compatible (Cubic, ..)

- BBR is *trying* to be that!
  - But it's complex and unfinished
  - Combining 2) and 3) is hard... For HP-WAN, must we ???

| | TCP+BBRv1 0.1%Pkt loss | TCP+BBRv1 1%Pkt loss | TCP+CUBIC 0.1%Pkt loss | TCP+CUBIC 1%Pkt loss |
|---|---|---|---|---|
| Single Stream | 14Gbps | 10Gbps | 8.6Mbps | Null |
| 3 Streams | 41Gbps | 24.5Gbps | 28Mbps | Null |
| 10 Streams | 70Gbps | 61Gbps | 91Mbps | Null |
| 25 Streams | 84Gbps | 84.7Gbps | Null | Null |

*Figure 2: TCP throughput performance,RTT=70ms,MTU=1500*

**draft-yy-hpwan-transport-gap-analysis-00**

# Conclusion

- **Cubic isn't good for HP-WAN**
  - But it's not the only CC algorithm under the sun.
    Popular because beneficial and reasonably downwards-compatible (to Reno)
  - Dropping the compatibility requirement, we could use many others
  - In the 2000s, and after, a ton of CC algorithms were made

- **Hence:**
  - No need to be Internet-compatible? (i.e., somewhat fair to Cubic, .. ?)
    => Probably just need to evaluate / tune existing mechanisms.
  - Must be Internet-compatible?
    => Contribute to BBR, or create something new (research).

# Thank you!

Questions?