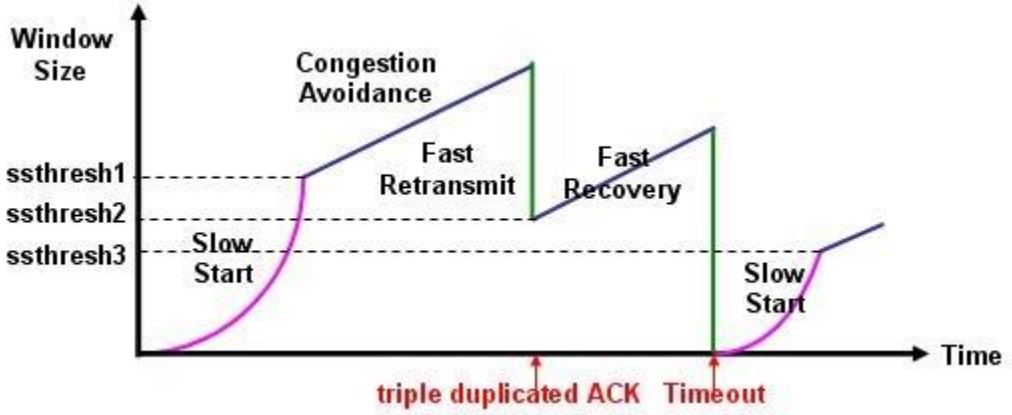
Congestion Control

Journey so far & Future directions

Congestion Control

Simply, avoid this in a network:

But, actually:



1. Conservation of packets principle in equilibrium (self-pacing)
2. Slow Start (reaching equilibrium)
3. Congestion Avoidance (adapting to capacity changes)
4. Timeout with exponential backoff (last resort, re-do slow start)

Congestion control history – Windows TCP

- Prehistoric times ☺
  - NewReno with ABC, traditional slow start
- 2008 onwards (inflection point = High BDP)
  - Windows Server 2008 and R2 – CTCP
  - Windows Vista and Windows 7 – NewReno
- 2012 onwards (inflection point = Datacenters)
  - Windows Server 2012 and R2 – Auto select DCTCP or CTCP
  - Windows 8 – CTCP
- 2016 onwards
  - Windows Server 2016 – Auto select DCTCP or CUBIC, LEDBAT++ Preview
  - Windows 10 – CUBIC
- Now
  - Windows Server 2019 – CUBIC with HyStart (LEDBAT++ and DCTCP configurable)
  - Windows 10 - CUBIC with HyStart

CUBIC with HyStart

- CUBIC congestion control now default for all connections
  - CTCP sensitive to delay fluctuations
    - Worsens due to the virtualized data path
  - CUBIC has better performance than CTCP when sharing the same bottleneck link
  - Higher retransmit rate as compared to CTCP
  - Better throughput for bulk flows on WAN
- HyStart for initial slow start – using delay mechanism
- Switch to Limited Slow Start after HyStart exit

LEDBAT++

- Low priority traffic – software updates
- Uses delay
- LEDBAT++ comprises of the following
  - Round trip latency measurements
  - Slower than Reno cwnd increase with adaptive gain factor
  - Multiplicative cwnd decrease with adaptive reduction factor
  - Modified slow start
  - Initial and periodic slowdown (for probing base RTT)
- Works well when the competing flows are CUBIC
- AQM interactions
- No easy receive side solution

Datacenter Congestion Control

- DCTCP
  - Originally invented to solve the incast problem in datacenters
  - Low latency for short flows and high throughput for long flows
  - Measure extent not presence of congestion using ECN
  - Low latency, low loss
  - Sender, receiver and network need to participate
- DCQCN
  - Lossless RDMA ROCEv2 with PFC
  - Intra DC use cases for high performance workloads
  - Combines elements of DCTCP and QCN

So what’s the problem?

- Bufferbloat
- Non-congestive loss
- Lack of ECN support in network
- AQM
- Wireless networks

Goals for next gen congestion control

- Fair sharing between multiple flows
- Fair sharing with CUBIC (Incremental deployment)
- RTT fairness
- Reduce percentage of retransmissions
- Low delay and minimize bufferbloat
- Work well (>= CUBIC) in a variety of networks
  - Low latency intra-DC
  - Enterprise
  - Wireless (Wifi, LTE)
  - WAN (high BDP)
- Low latency for short flows, high throughput for long flows
- Easy to reason about, model, and implement
- Optionally support “less than best effort”

PCC Vivace

- Online learning: can reason about behavior
- Straightforward mapping in the congestion control interface
- App-limited problem
- Actual rate versus Target rate
- Cwnd as a proxy for Target rate (apply % changes)
- Issues with Utility function on low data rates
- Next steps
  - More measurements around fairness and RTT fairness
  - Make it work for low data rates
  - “Less than best effort” utility function

L4S

- TCP Prague
  - Extend DCTCP
  - Accurate ECN
  - ECN for control packets
- ECN identifier for scalable queue
- DualQ Coupled AQM

Problems with Signals and Mechanisms

- Explicit Signals
  - Dup ACK / SACK
    - Reordering versus loss
  - ECN
    - Not widely deployed, bad middleboxes
- Implicit Signals
  - Timeout
    - Spurious versus not
  - Delay increase
    - ACK compression, LRO, stretch and delayed ACKs, virtualization
- Timers
  - Power efficiency
- Pacing
  - Interaction with TSO, CPU cost
- AQM
  - Bad interaction with delay based congestion control?
- Probing
  - Lots of heuristics and tuning

Challenges

- Can be build a network model that always holds?
- Can we retain the simplicity of current congestion control algorithms?
- How to solve the ECN deployment logjam?
- What should be the set of test cases for new congestion control?
- How do we avoid turning congestion control into an arms race?
- RFC 6077 ☺
  - Network Support
  - Corruption Loss
  - Packet Size
  - Flow Startup
  - Multi-Domain Congestion Control
  - Precedence for Elastic Traffic
  - Misbehaving Senders and Receivers
  - Other Challenges
  - RTT Estimation
  - Malfunctioning Devices
  - Dependence on RTT
  - Congestion Control in Multi-Layered Networks
  - Multipath End-to-End Congestion Control and Traffic Engineering
  - ALGs and Middleboxes