Scaling IW with the Internet,

an engineering argument

Matt Mathis
mattmathis@google.com

For ICCRG at IETF 78

30 July 2010

My personal view

- We should permit IW16 (but recommend IW10)
  - As long as TCP is using SACK
- For host vendors, recommend a phased approach
  - Raise shipped IW in steps, with lots of evaluation
  - Corresponding stack and application changes
    - Adapt IW per interface type
    - Set Initial (adaptive) rwin per IW
    - Moderate the number of browser threads
- For content providers, recommend measurements
  - Should not cause extra losses during IW
    - Exact criteria may be hard to agree on
    - Must instrument and measure actual content
- IW10 is a good first goal
  - Assume IW16 will take at least two upgrades

Multiple connections

- Many websites open dozens of connections, some hundreds
  - Browsers open 4, 6 or more connections
  - Sites spread content across multiple domains
    - Multiplicative impact
- For these sites IW16 is clearly too big
  - Expected symptom: latency increases
- We (tcpm etc) can not regain control except by a phased approach
  - Must cause measured pain for greedy apps

- Assume K=4 connections are ok

Bottleneck buffer space

- Each component is optimized in its native context
    - Justified by simple lab experiments & benchmarks
- All (slow) links have common tuning criteria
    - Acceptable worst case interactive performance
        - Buffers not larger than a few seconds
    - Can be filled by a single bulk flow
        - Requires full BDP buffer space for a long path
    - Can be mostly filled w/ bulk plus short flows
        - Synchronized losses requires surplus space
    - "Optimal" experience for contemporary browsers
        - At the time designed (e.g. IE? on XP)
        - 4 connections were typical for many years
- One second queues were fairly standard
    - Predates VOIP

Striking a balance

We want: burst size ≤ queue size

IW * (K * ND) ≤ (RTT * scale) * Rate

- K - Number threads per server
- ND - Number of domains per page
- K*ND - Aggregate application multiplier

- RTT - Composite Internet RTT
- scale - Aggregation compensation
  - 2 or more at very low rates
  - << 1 at high aggregation backbone rates

- RTT*scale - Drain time

Striking a balance

$$IW * (K * ND) \leq (RTT * scale) * Rate$$

Substituting, rearranging:

$$IW \leq (1/4)(drain\_time)(Rate)$$

i.e. The optimal IW is one quarter of the drain time for some baseline data rate.

Slow access links (non-broadband)

- Less than 256k bps in most of the world
- Relatively rarely shared
    - Too slow
    - Mostly not used to connect LANs to the Internet
        - Mobile AP/tethering a possible exception
- End system typically manages the link
    - E.g. Cell phones, dialup modems, etc.
    - Direct knowledge of data rate and buffer space
- Can set IW and/or initial rwin directly
    - Clamp both inbound and outbound bursts

Faster access links

- At 1 Mb/s
    - 192 ms to drain 16 segments
    - ~3/4 of a second to drain 4*16 segments
        - Would be fine in the pre-VOIP days
- At even higher rates
    - Becomes less likely that buffer space is a problem
    - Browsers discover that more parallelism is faster
        - Mostly because they multiply up IW
        - They do their own context specific optimization
            - This implies that IW3 is too small

In between (256 kbps)

- Traditional 1s queue holds 21 segments
    - Enough for: 7*IW3, 2*IW10
    - Not enough for 4*IW10
- ITU G.114 calls for queuing times under 150 ms
    - To better support VOIP
    - Only 3 1500 Byte segments at 256 kbps
        - Not enough for TCP fast retransmit
        - Not enough for >1 connection at any IW
- Can elect to use "slow link" fixes
    - Clamp IW and initial rwin
        - W/ 1s queue, fixes 4*IW10 or even 4*IW42
    - Nothing can help 10*IW3 .....?!?!
- Fewer connections, larger IW is better!

Multiple connections revisited

- Greedy apps have already usurped congestion control
- Pick the ideal IW for non-greedy apps
    - Assume omniscience
    - This IW will be too large for greedy apps
        - Expect them to hurt themselves
- Consider IW10 and IW16 measurement data
    - The across the board positive results for IW10 suggests that it is too conservative
    - We expect the ideal IW to have mixed results

My conclusion

- Raising IW and rehabilitating greedy apps would be a good thing
- Need a phased deployment
    - IW10 a good near term goal
    - IW16 a likely future goal
    - Can't predict beyond that yet
- Clients (host vendors) need tweaks
    - Adapt IW per interface type and rate
    - Set initial rwin per IW
    - Moderate number of browser threads
- Content providers need to use measurements
    - Reduce # domains to offset IW changes