

# GigaTCP—TCP on Gigabit Ethernet

Stanislav Shalunov <shalunov@internet2.edu>

Internet2/NLANR Joint Techs Workshop, Boulder, 2002-07-29

## Problem Statement

Given:

- 2.5 Gb/s OC-48c backbone that doesn't drop packets
- GigaPoPs with 2.5 Gb/s OC-48c connectivity
- Well-built machines with Gigabit Ethernet cards
- Clean path connectivity for these machines

Produce:

- A single TCP connection that takes up significant share of 1 Gb/s (on a sustained basis)
- Understanding of difficulties involved

## Motivation

- Tell how the networks we engineer today will behave few years down the road when TCP connections with such throughput become more commonplace
- Measure impact of things like minor packet re-ordering on TCP throughput
- Validate the assumption of low loss in Internet2 networks
- Learn about effects of network techniques (e.g., SACK)
- Learn about effects of host techniques (e.g., zero copy TCP)
- Prove or disprove that in tomorrow's network TCP can still provide viable transport even with today's algorithms
- Evaluate the real need for dangerous host techniques such as hardware checksumming
- Understand just exactly how hard is it to run hundreds-of-megabits-per-second TCP flows over WAN today and what are the difficulties involved in doing so

## Difficulty Dimension: TCP Congestion Control

- According to an equation due to Matt Mathis,

$$\text{throughput} \approx 0.7 \frac{\text{MSS}}{\text{RTT} \sqrt{\text{loss}}}$$

- Let's set  $\text{RTT} = 70 \text{ ms}$ ,  $\text{throughput} = 1 \text{ Gb/s}$ ; then
- $\text{MSS} = 1460 \text{ B} \Rightarrow \text{loss} \approx 10^{-8} = 0.000001\%$ ,  
or one loss in about 20 minutes (*eight nines* rule?!)
- $\text{MSS} = 4096 \text{ B} \Rightarrow \text{loss} \approx 10^{-7} = 0.00001\%$ ,  
or one loss in about 5–6 minutes
- $\text{MSS} = 8192 \text{ B} \Rightarrow \text{loss} \approx 4 \times 10^{-7} = 0.00004\%$ ,  
or one loss in about 2–3 minutes

## Difficulty Dimension: TCP Congestion Control (Cont.)

- With throughput and RTT fixed: As the MSS increases linearly, the loss rate increases quadratically and time between losses decreases linearly
- With MSS and RTT fixed: as throughput increases linearly, loss rate decreases quadratically and time between losses increases linearly

## Difficulty Dimension: Moving Many Bits Through the Hardware

- Forget about congestion control and the WAN; most people get less than 100 Mb/s with Gigabit Ethernet on a LAN
- 32-bit 33MHz PCI bus has nominal half-duplex throughput just over 1 Gb/s—want 64-bit 66MHz PCI
- NICs are mostly bad; we found that 3c985B-SX, 3c996-SX, and SysKonnect SX cards are good
- SDR ECC SDRAM with interleaving (today, one might play with ECC DDR)
- The chipset is key; we had ServerWorks ServerSet III HE
- We used SuperMicro boards, but chipset is the important thing, as long as interleaving is supported
- Must have  $n$  identical DIMMs to take advantage of  $n$ -way memory interleaving (which both the chipset and the motherboard must support for it to work)
- Did I tell you to get the fastest CPU? Make that two CPUs

## **Difficulty Dimension: Preparing the Software for High Throughput**

- We used FreeBSD; people have achieved good result with Linux, too
- TCP send and receive buffers have to be 8 or 16 MB
- Drivers often need as much memory (often hard-coded)
- IP queue memory wasn't a problem, but could be
- Tune, edit kernel header files, recompile, reboot, loop
- If 'int' is your byte counter type, it rolls over every 17 s

## On a LAN (Testing)

- This one is easy with decent hardware
- Back-to-back, was able to saturate the link with the interfaces set to MTU = 4470 B (kernel sets MSS = 4096 B = one page)
- Window size of 128 KB helps, but 64 KB is fine
- In fact, as window size gets WAN-like (e.g., 8 MB), performance deteriorates
- With MSS = 8192 B, can saturate the link hands down
- No performance deterioration with growing window size when MSS = 8192 B
- Lesson: set MTU to 9000 B or to the highest number your interfaces support and get the campuses and the GigaPoPs carry those frames to Abilene

## Over a WAN (Abilene)

- Can routinely reproduce 200–300 Mb/s
- The best throughput I have seen is 600 Mb/s
- ‘Peak’ rates are meaningless
- Averaging over tens of seconds is required (e.g., by using ‘iperf -s -w8M -i20’)
- Larger MTU sizes really help with both congestion control (less time between losses) and with moving packets through the end hosts (lower packet-per-second rates and page-aligned I/O)

## Lessons Learned

- High throughput is hard
- High throughput is possible
- Higher settings of MTU than 1500 B are important
- Most normal machines cannot move the bits fast enough
- Many Gigabit Ethernet cards cannot move bits fast enough
- Should we change TCP congestion control? Probably.

## How do we continue to get good TCP performance in the future?

- Specifying MTU in units of seconds rather than bytes is hardly possible (we have little influence in IEEE, which will keep repeating the bridging mantra)
- Even moderate MTU increase to 9 KB makes life easier
- Ethernet has this tiny cell it calls a frame and we cannot really increase it; is it time to think about Ethernet SAR?
- Is it time to think about what will replace Ethernet and have MTU specified in seconds (and probably no possibility of shared segments)?
- Are HighSpeed TCP (be more aggressive when already doing well) and IP QuickStart (get to the ultimate rate in one RTT) the right answers?
- Is more radical XCP by Dina Katabi et al. the right answer?
- The presenter has some half-baked ideas of his own

## Doing your own GigaTCP

- Let us know you're working on it—we might be able to help
- Good connectivity: the closer to the backbone the better
- Good MTU: 4096 B payload helps a lot; 8192 B even better
- Good hardware: the chipset and the NIC are most important
- Good software: decent OS, well-tuned buffers throughout
- Host tricks: works without zero copy or hardware checksums
- Networking tricks: can work without SACK or RED
- However, zero copy TCP and SACK are good and could help
- Success back-to-back doesn't guarantee anything on a WAN
- Use large averaging intervals (but watch out for overflows)
- `tcpdump` helps (but Heisenberg uncertainty principle hurts)
- Verify your results: <http://netflow.internet2.edu/weekly/>

## Plea to Network Operators

- Please increase your MTUs to the maximum your equipment supports
- Paging all GigaPoP operators in the room: Abilene supports 9 KB MTU; coordinate with the Abilene NOC to set your Abilene link right; coordinate with campuses to have 9 KB MTU downstream
- Make it possible for users on Gigabit Ethernet to use packets with 8 KB payloads
- No, really: 1500 B is not enough with Gigabit Ethernet and 9 KB *is* better than 4470 B

Must set MTU to 9 KB ...