Emulating Network Latency on High Performance Networks

Berry Hoekstra | Niels Monen

Outline

- Introduction
- Related research
- Approach
- Research
- Results
- Conclusion
- Questions?

Introduction

- Emergence of high-speed connectivity
  - How do protocols and applications behave?
  - New research needed
- Can be tested using:
  - Proprietary equipment
  - On a real-world link
    - Often not available
    - If available, difficult to realise
- High costs and availability can terminate a project
  - Not if off-the-shelf hardware can be used
  - Software emulation

Research question

- *"What are the characteristics of long distance high performance links and to what extent can they be emulated with off-the-shelf hardware?"*

Sub-questions:

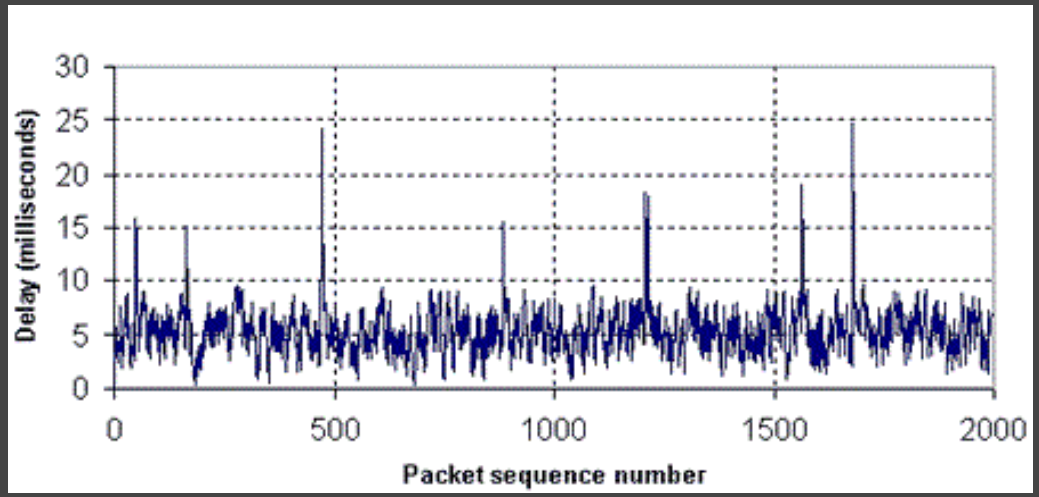
  - *"What solutions are available for this purpose?"*
  - *"What is the effect of using different network parameters?"*
  - *"Does it matter if a real-time or regular kernel is used?"*

Related research

- Li, et al.: Evaluation of TCP on high-speed networks
- Former OS3 students: 10 GigE performance measurements
- Yildirim, et al.: Evaluation of different emulation tools
- Hemminger: Emulating network characteristics using netem
  - netem workings and effects
  - Only "low" speed connections (1 GigE)
- Wu, et al.: 10 GigE emulation
  - Used as reference for test results

Network properties

- Network latency
  - Amount of time it takes for a packet to reach its destination and back (Round Trip Time)
- Network characteristics
  - Delay (RTT)
  - Jitter
  - Jitter distribution
    - Jitter occurrence

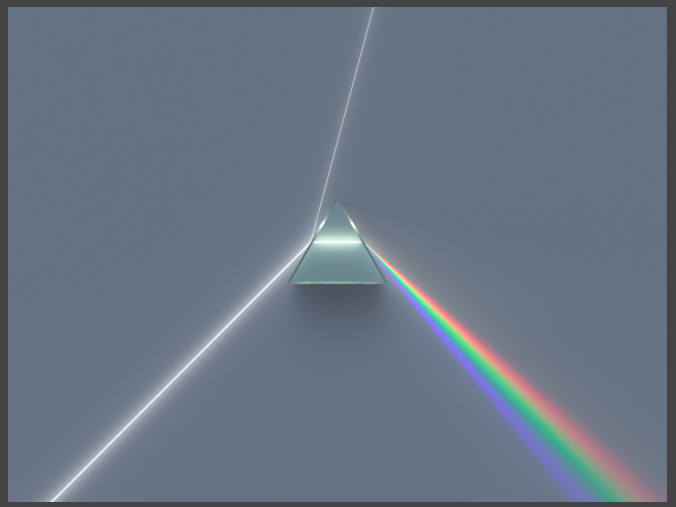


- Fast link with long delay = Long Fat Network (elephant :))
  - High BDP

Bandwidth Delay Product

- BDP = Bandwidth (byte) * Delay (s)
  - Amount of in-flight data
- BDP ≈ TCP Window Size
  - Amount of unacknowledged data on the line
- Calculate optimal Window Size
  - Using known RTT and link speed

Causes of latency



- Optical limitations
  - Light speed limit (~300km/ms)
  - Amplifiers
- Router delay
  - Congested buffers
  - Processing and transmission time
  - Fairness (Quality of Service)

Optimize network parameters

- Path MTU
  - Ethernet frame size
  - Prevents fragmentation along the path

- TCP parameters (set using *sysctl -w net.ipv4.tcp_**)
  - Congestion algorithm
  - TCP window size (Receive/Send Buffer)
  - Remove overhead:
    - Disable SACK and Timestamps

- Set MTU Jumbo frames
  - *ifconfig <NIC> mtu 9000*

- Set packet transfer queue length
  - *ifconfig <NIC> txqueuelen <queue length>*

Existing tools

- Emulators
  - NIST Net
  - Dummynet
  - netem
  - Emulab
  - Web100
- We chose netem
  - In the kernel by default
  - Can use other papers as reference
- Generate traffic using iPerf 2.0.5

Emulation with netem

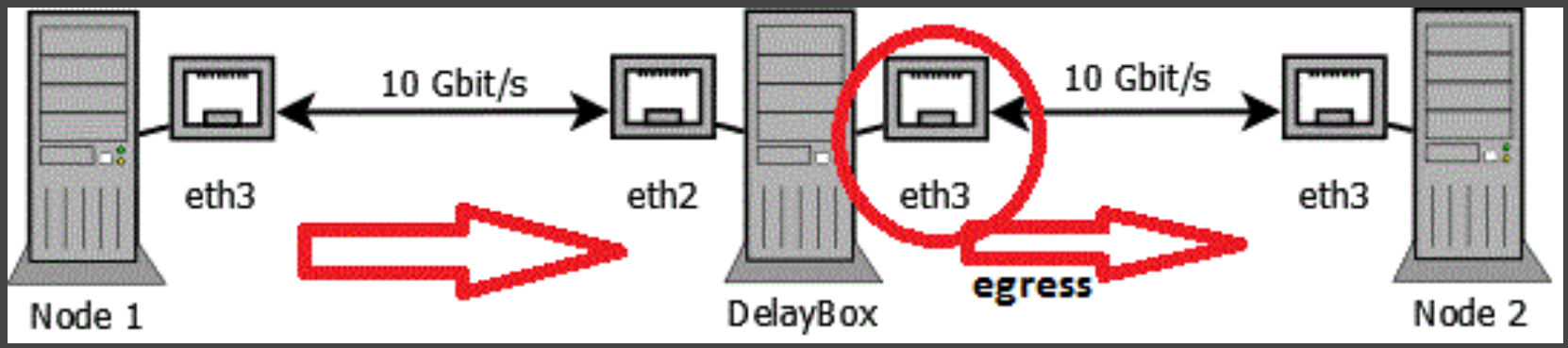
- Kernel module
  - Included by default since Kernel 2.6.7
- Emulation depends on kernel resolution
  - Resolution of 1000 Hz (since Kernel 2.6)
  - Matters to the precision of emulated delay
- Higher resolutions for high-speed connections (40 GigE)
  - More packets per millisecond (theoretical ~5MB/ms)
  - Achieve more fine-grained emulation (<1ms)
  - 10.000 Hz, but no patch for latest kernel
- Hypothesis: Real-Time kernel
  - netem can apply delay in real-time

Kernels

- Kernel ticks
  - New time slice for processes
  - Resolution of 1000 Hz = 1 tick/ms
- Real-Time Kernel
  - Guaranteed system response time
  - Achieve the lowest possible latency at any cost
- Tickless kernel
  - To save energy when idle
  - Ticks "on demand"

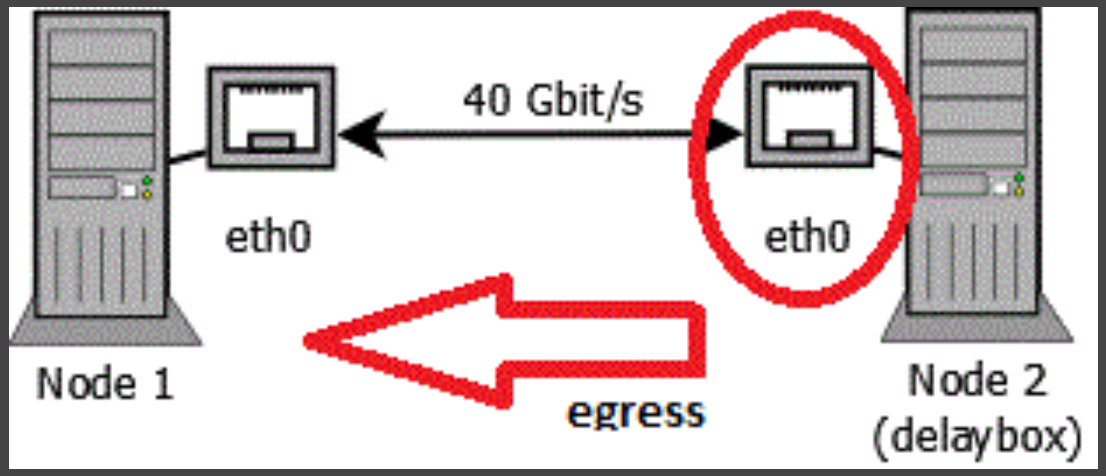
10 GigE Lab setup

- 3x Dell R210 (1U)
  - 2 nodes (sender/receiver)
  - 1 delaybox / bridge (netem)
- Daisychained
  - No intermediate nodes
  - No "outside" influences
- Connectivity
  - 1 GigE Broadcom (onboard)
  - 10 GigE Mellanox/Chelsio



40 GigE Lab setup

- 1x Supermicro Twinnode
  - 2 machines in 2U enclosure
- Directly connected
  - Lack of 40 GigE cards
  - Node + delaybox
- Connectivity
  - 40 GigE Mellanox connected back-to-back



Tests

- Different NICs
  - 1, 10 and 40 Gigabit Ethernet
- Different Kernels
  - 100 Hz, 1000 Hz, Real-Time and Tickless
- Different characteristics and window sizes
  - No delay
  - delay
  - delay+jitter
  - delay+jitter+distribution

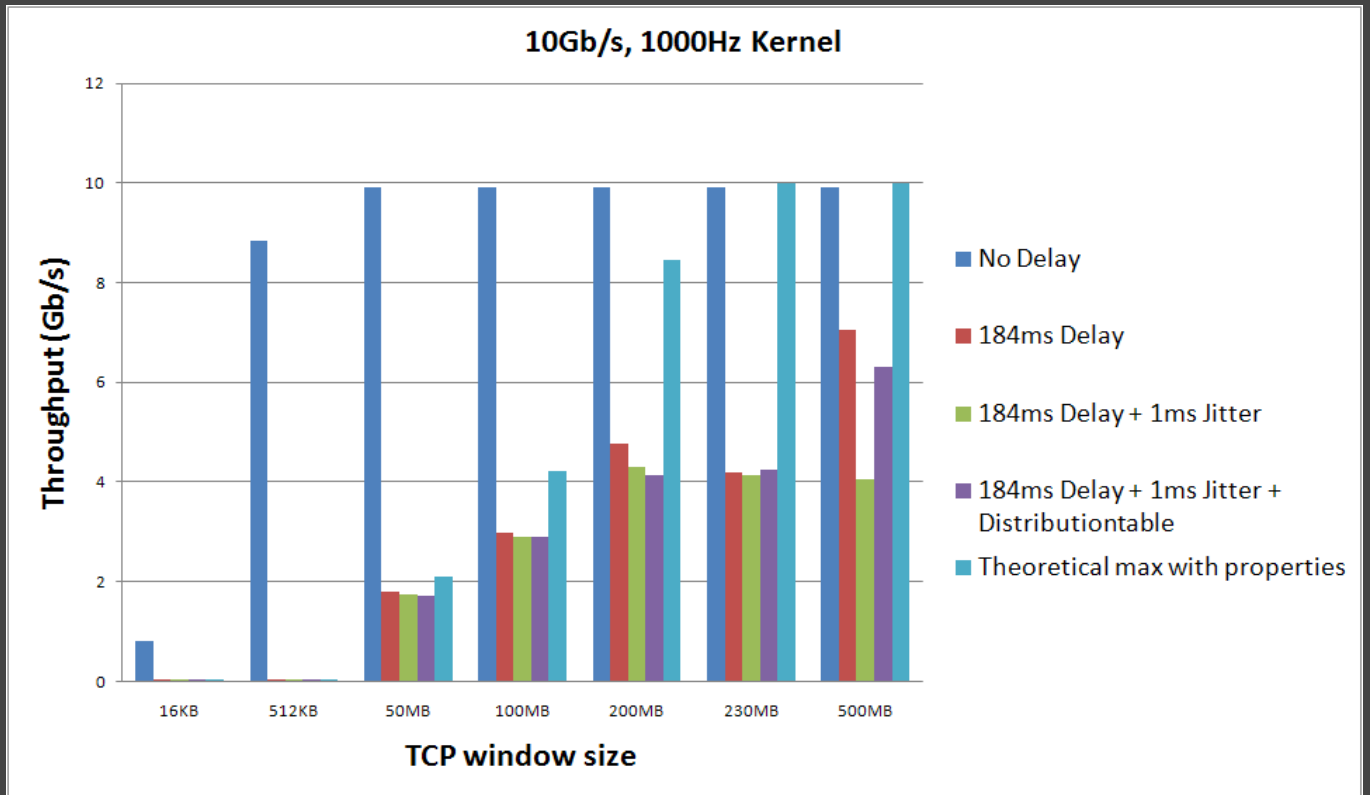
Obtaining real-world properties

- International link from Amsterdam to San Diego
  - 10 Gbit/s shared link on Netherlight (SURFnet)
  - No root access (no tweaking!)
  - Throughput: ~5 Gbit/s UDP and ~1 Gbit/s TCP
  - See if it is possible to emulate
    - Capture 24 hours of ping data (characteristics)
    - Extract RTT properties from ping data
    - Extract RTT, jitter and jitter distribution table
      - RTT = 184.000071 ms
      - Jitter = 0.008450 ms
      - Dist table = /usr/lib64/tc/sdiego.dist

Results (1)

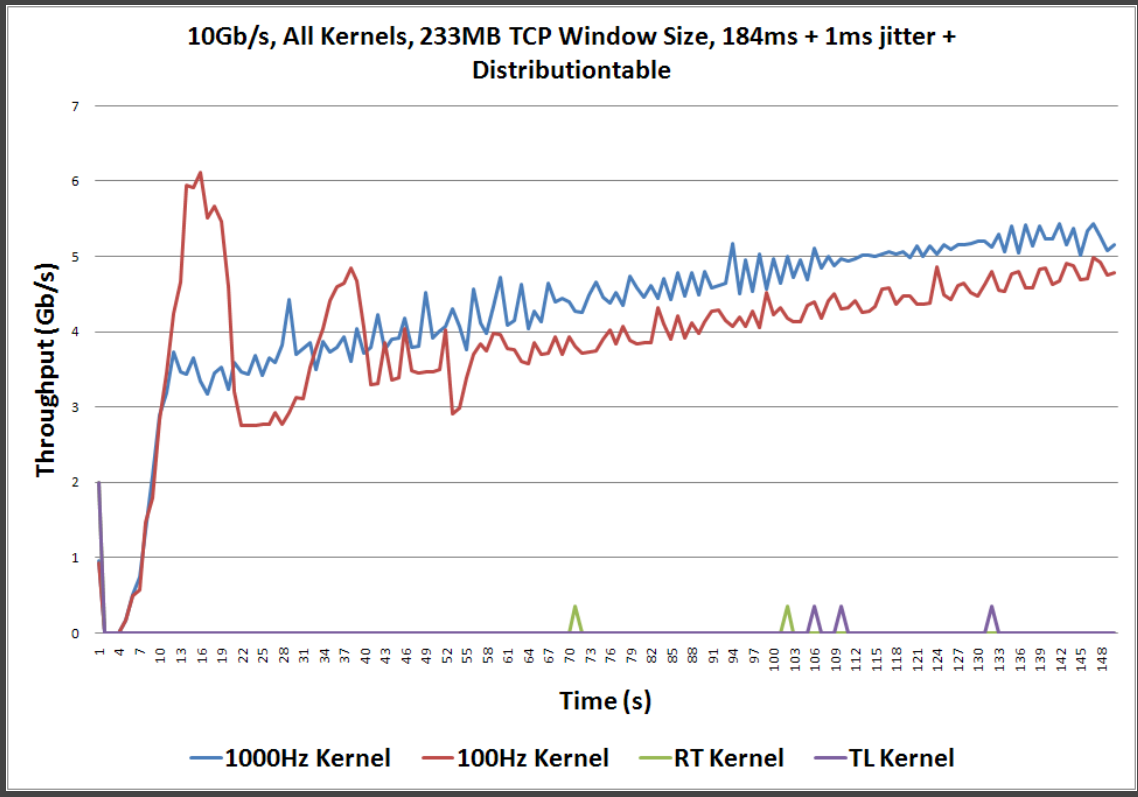
- 10 Gigabit Ethernet - 1000Hz kernel

- With the optimal window size, we should get ~10Gb/s throughput
- Only get ~4Gb/s
- Netem can't emulate on such high speeds
- Suspect CPU bottleneck
  - 1 core@100%
  - 1 thread



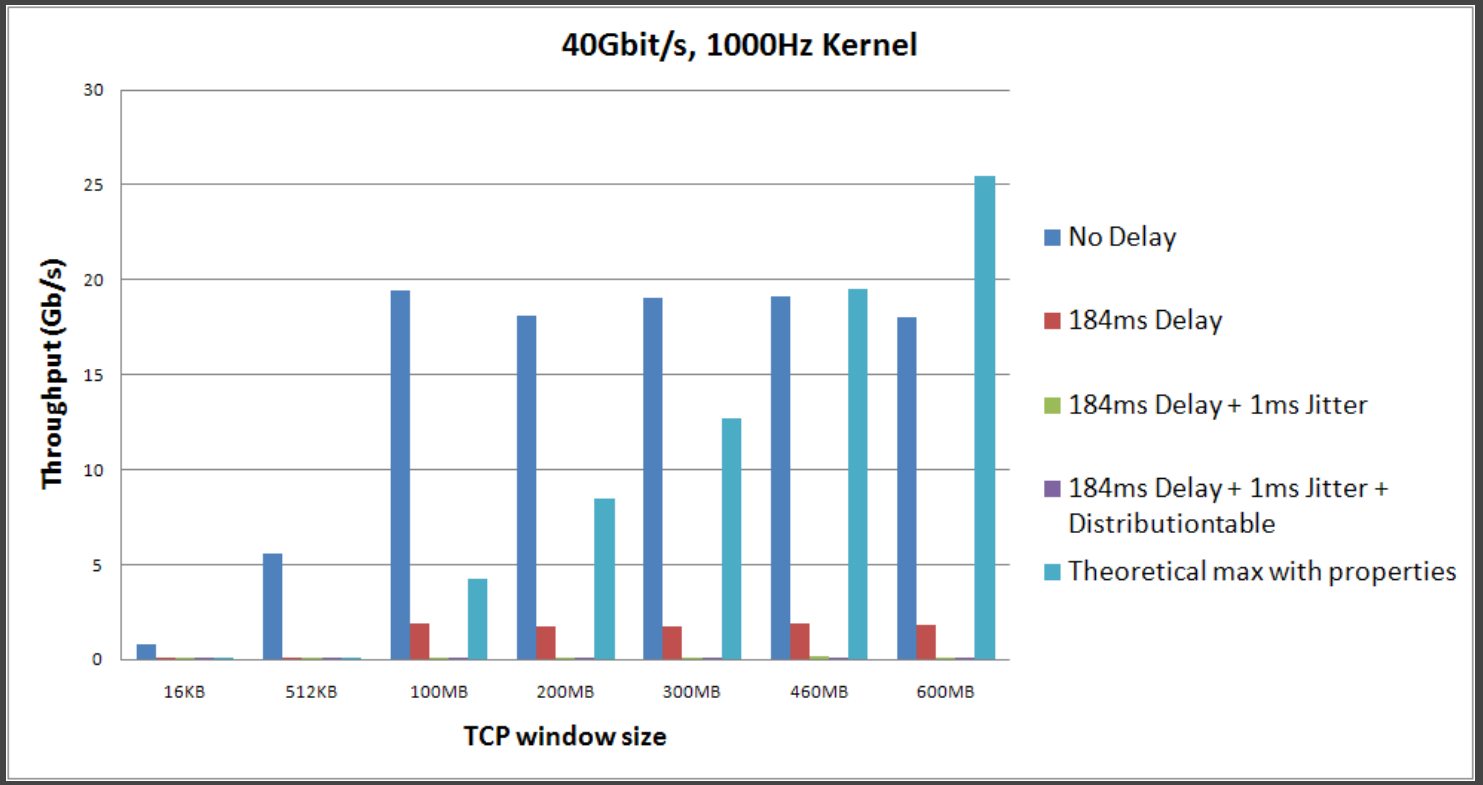
Results (2)

- 10 Gigabit Ethernet - all kernels
- 100 Hz and 1000 Hz
  - Slowly builds up
  - Congestion control kicks in (HTCP)
  - 100 Hz RTT has additional 10ms delay
- RT and Tickless
  - No performance
  - CPU busy with interrupts



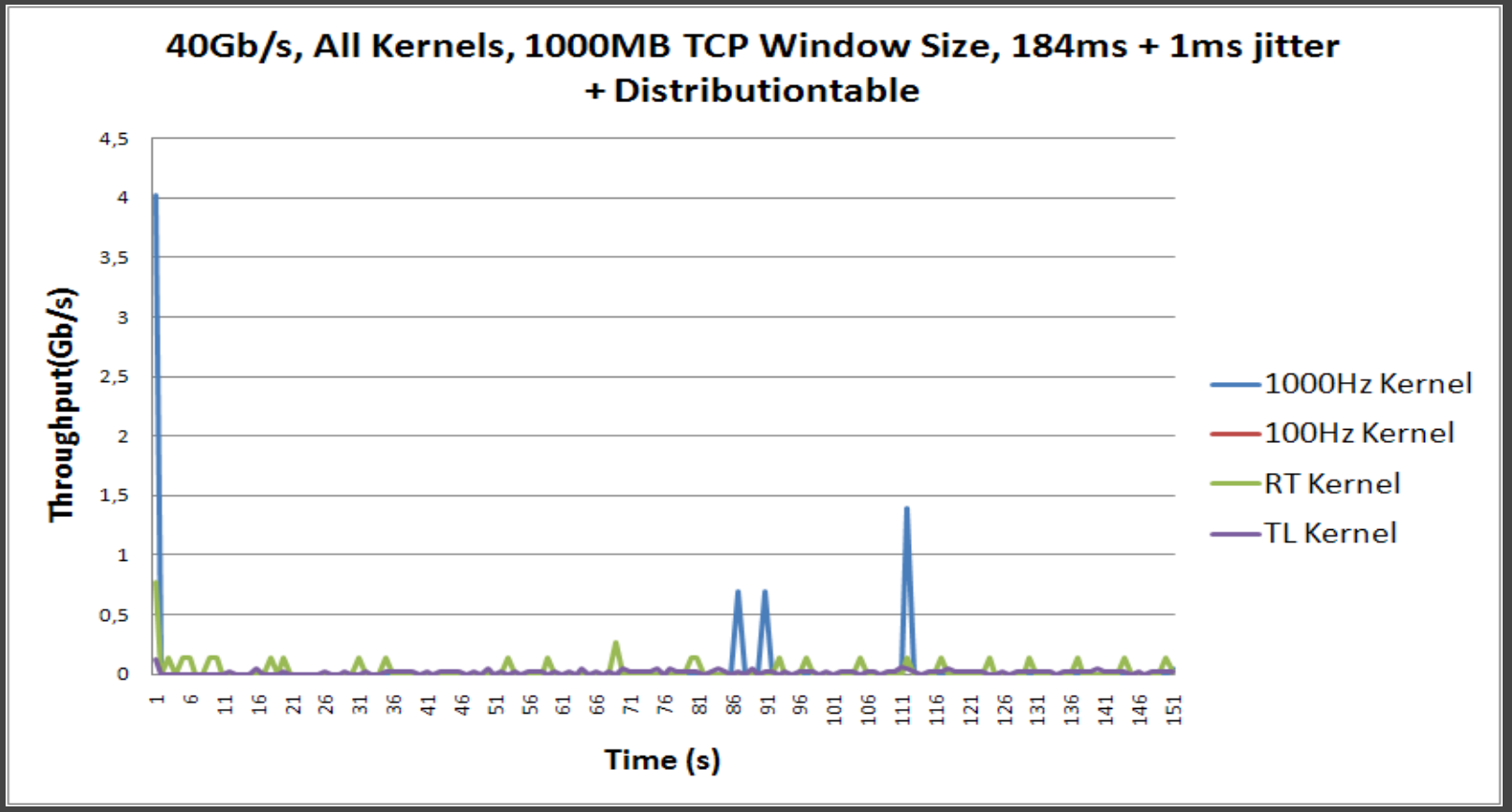
Results (3)

- 40 Gigabit Ethernet - 1000Hz kernel
- Max 19Gb/s without delay
  - PCI-E bus limit
- Max 2Gb/s if only adding delay
- Performance drops with delay + jitter
  - Also with distribution table



Results (4)

- 40 Gigabit Ethernet - all kernels
- No performance at all



Conclusions (1)

- Tweak network parameters on high performance links
  - Optimal performance and less overhead
  - Optimize throughput by:
    - Tweaking TCP parameters
    - Set path MTU
    - Packet transfer queue length
- Default Real-Time Kernel is not suitable for emulation
  - Too many cycles needed to process network interrupts
  - Drop in performance
- On the 40Gb/s link huge performance drops on all kernels
- On the 10Gb/s link we see ~4Gb/s max.
- The 100Hz kernel couldn't maintain the correct delay

Conclusions (2)

"What are the characteristics of long distance high performance links and to what extent can they be emulated with off-the-shelf hardware?"

- 10 GigE and 40 GigE don't achieve expected throughput
- No mitigation if different kernel resolutions are used
  - Not even with real-time kernel (too many interrupts)
- Suspect netem is not optimised for high throughput links
  - Unable to cope with the large amount of packets
  - Even though buffers are large enough
- We advise to only use netem if you have a maximum link speed of 4 to 5 Gbit/s

Future work

- Interrupt Coalescence
  - Limit the NIC interrupts
- Real-Time Kernel tweaking
  - CPU resource distribution
- Perform tweaking on the international link
  - Time delay because of time differences
- Re-test when 40 GigE is "production ready"
  - And when there are 4 cards available
- Emulation tool comparison

Questions?

© Google Image Search