Achieving reliable high performance in LFNs
(long-fat networks)

Sven Ubik, Pavel Cimbál
CESNET

End-to-end performance

- E2E performance is a result of interaction of all computer system components:

    - network, network adapter, communication protocols, operating system, applications

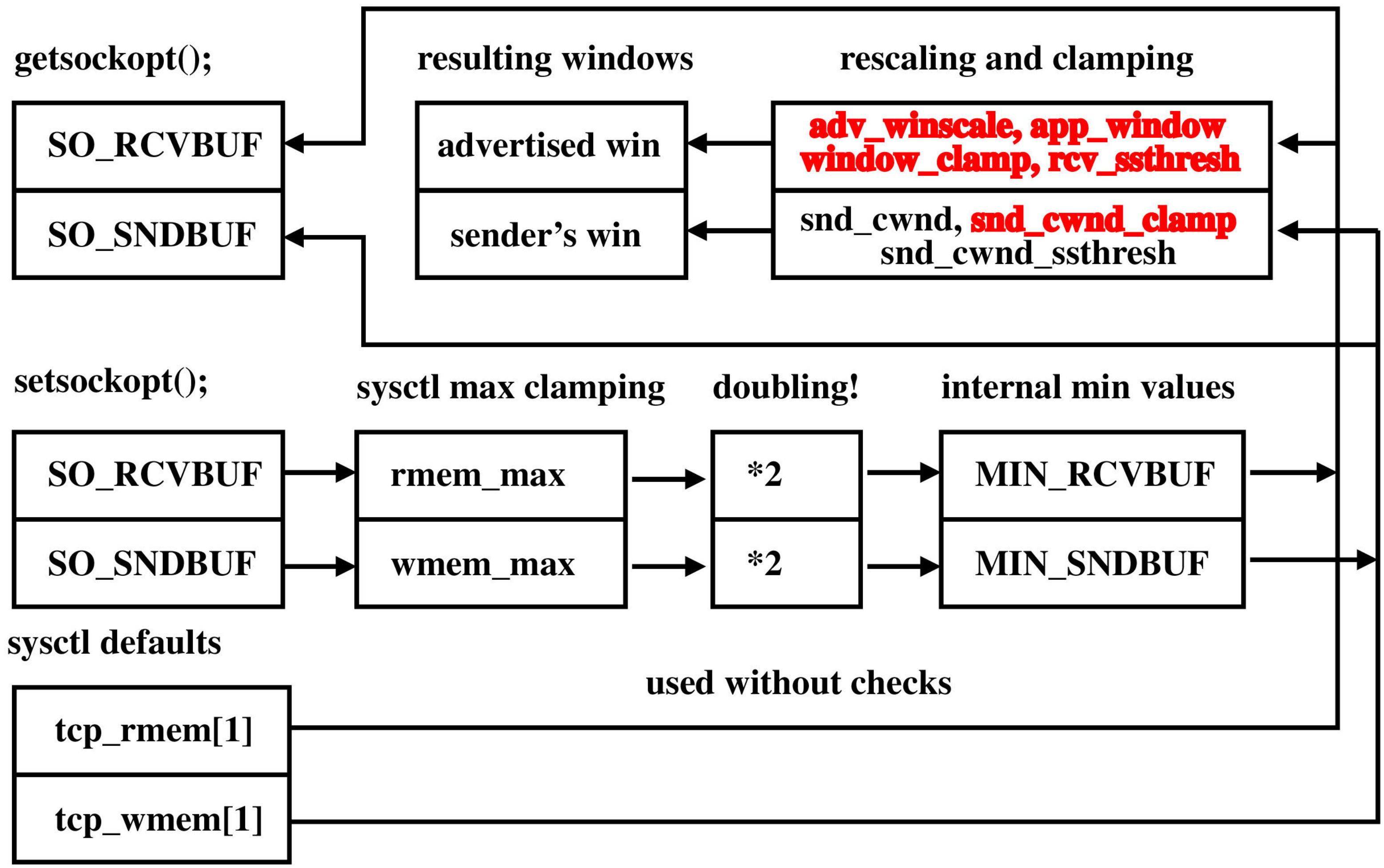
- Decided to concentrate on E2E performance on Linux PCs

- Primary problem areas:

    - TCP window configuration
    - OS / network adapter interaction
    - ssh / TCP interaction

TCP windows configuration

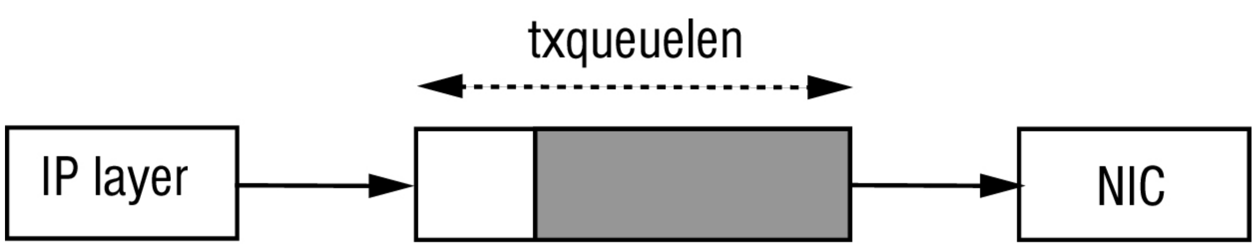
Linux 2.4: how big are my TCP windows?



Throughput with large TCP windows

Interaction with network adapter must be considered:

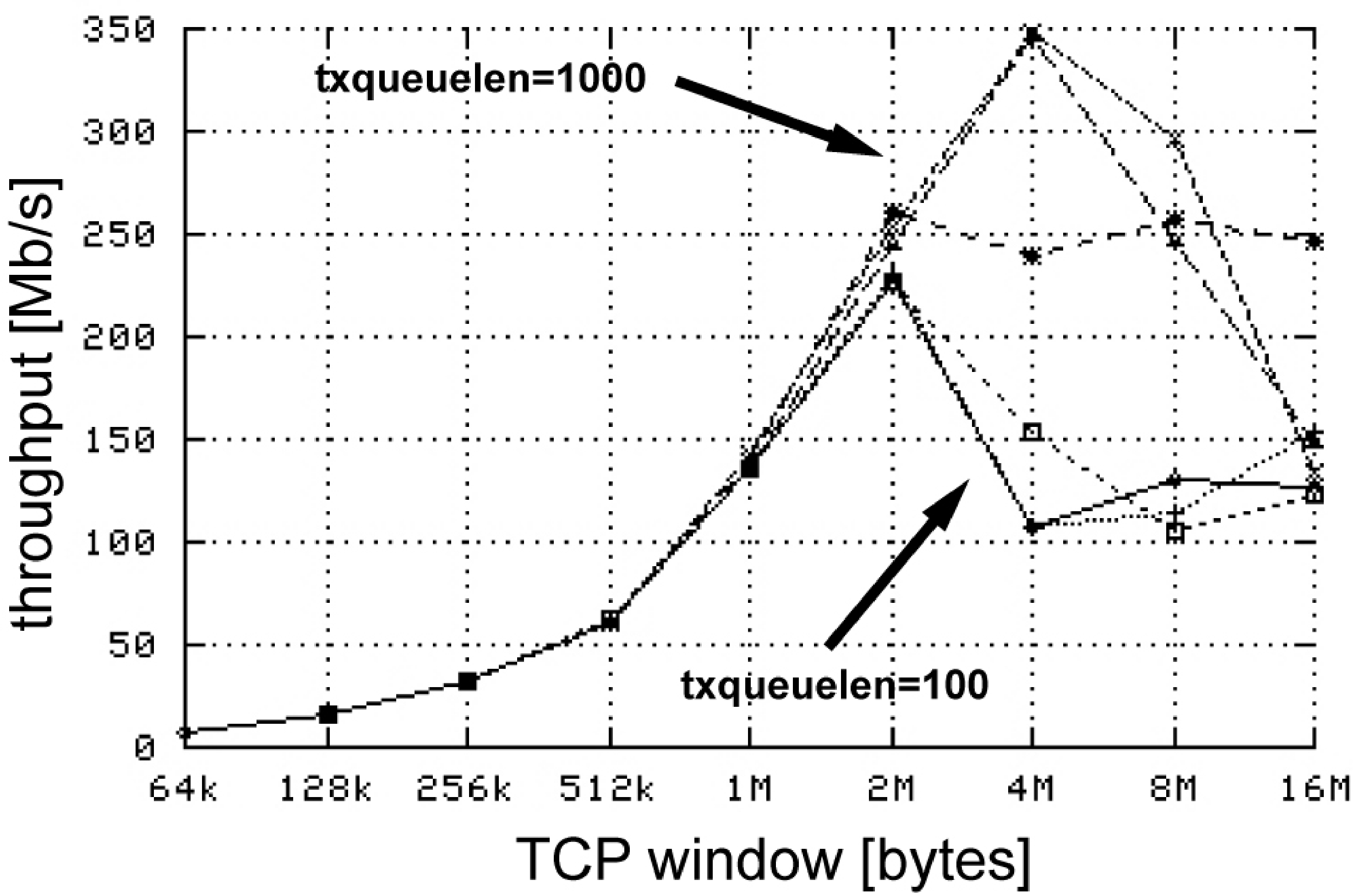
- large TCP sender window allows large chunks of data submitted from IP through txqueue to adapter
- full txqueue -> send_stall() to application and context switch
- no problem as long as txqueue is large enough for a timeslice



for Gigabit Ethernet adapter and standard Linux system timer:
txqueuelen > 1 Gb/s * 10 ms / 8 bits / 1500 bytes = 833 packets

ifconfig eth0 txqueuelen 1000

Throughput with large TCP windows, cont.



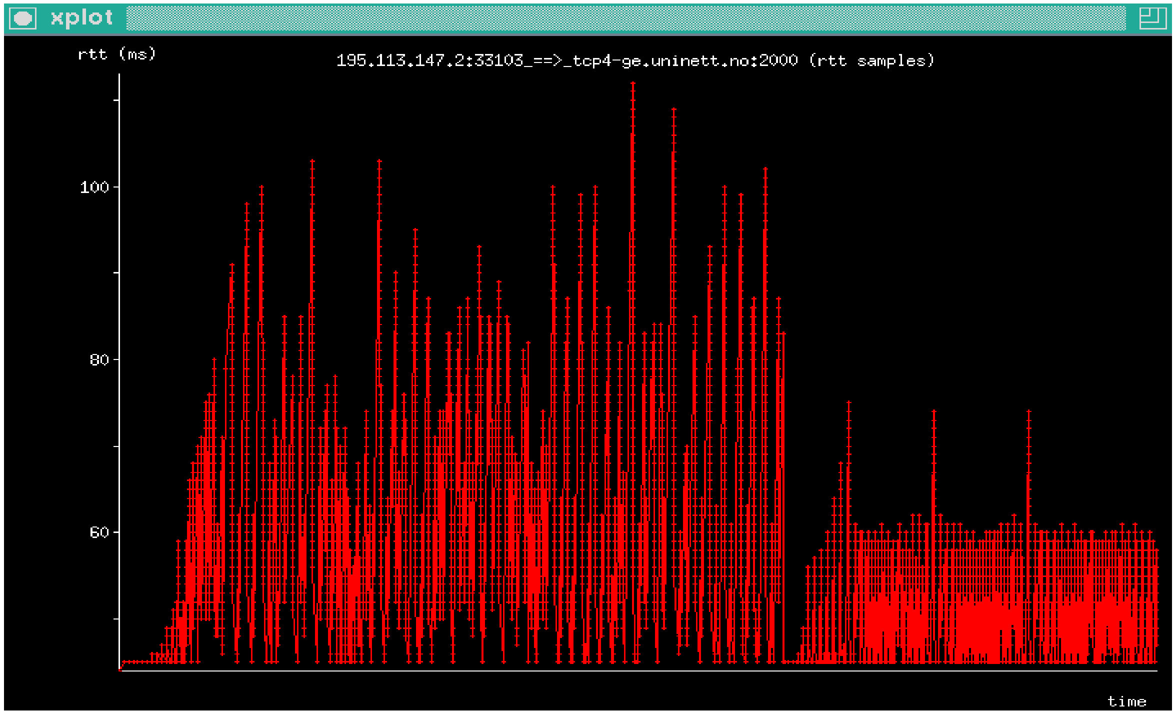
Using “buffered pipe” is not good

Router queues must be considered:

- No increase in throughput over using „wire pipe“
- Self-clocking adjusts sender to bottleneck speed, but does not stop sender from accumulating data in queues
- Filled-up queues are sensitive to losses caused by cross-traffic
- Check throughput (TCP Vegas) or RTT increase ?

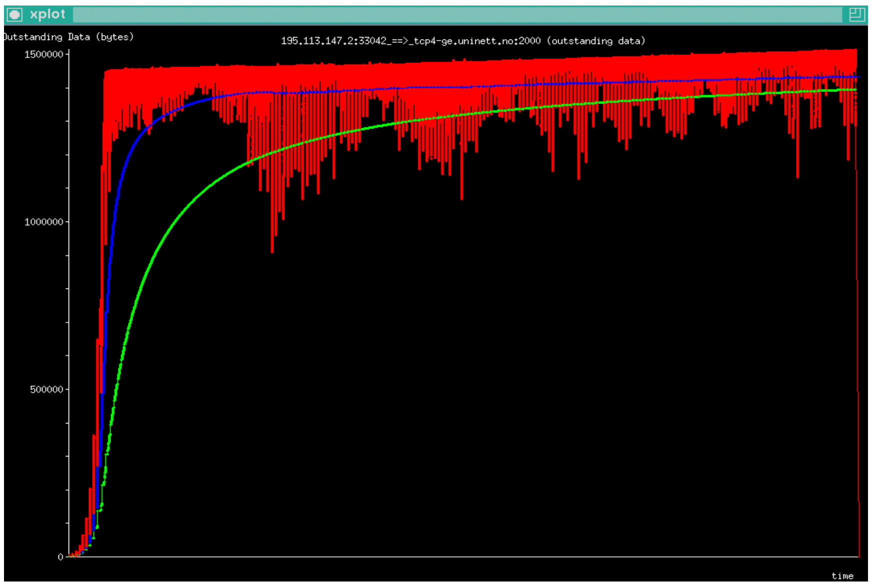
rwnd<=pipe capacity
    bw=rwnd/rtt
rwnd>pipe capacity
    bw~(mss/rtt)*1/sqrt(p)

Flat lower bound
    RTT=45ms
Fluctuations up to
    RTT=110ms
Bottleneck
    installed BW=1 Gb/s
Buffer content ~8 MB

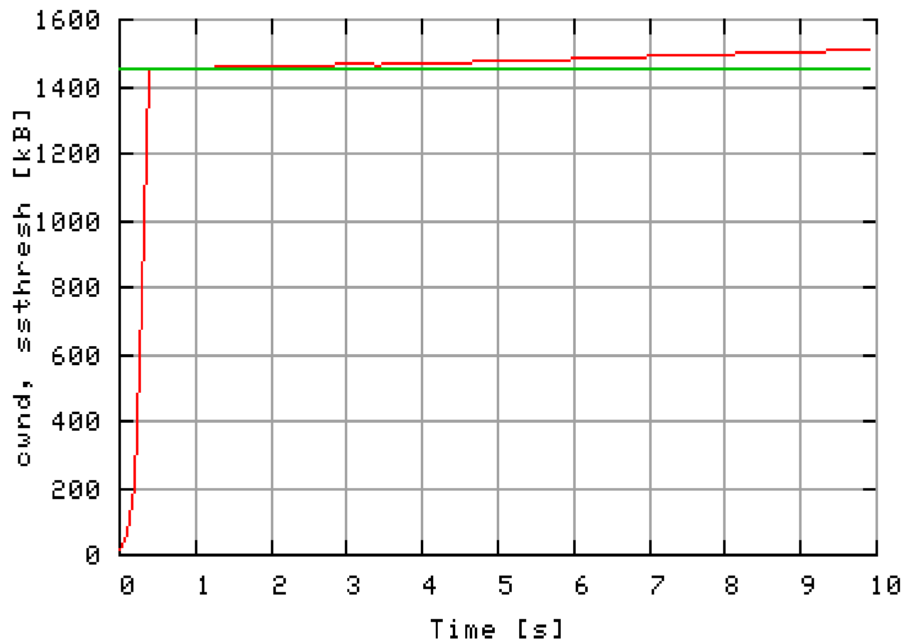


Other configuration problems

TCP cache must be considered



owin development



rwin development

initial ssthresh locked at 1.45 MB

echo 1 > /proc/sys/net/ipv4/route/flush

Bandwidth measurement and estimation

Test paths: cesnet.cz <--> uninett.no, switch.ch

pathload over one week:

- 27% measurements too low (50-70 Mb/s)
- 7% measurements too high (1000 Mb/s)
- 66% measurements realistic (750-850 Mb/s), but range sometimes too wide (150 Mb/s)

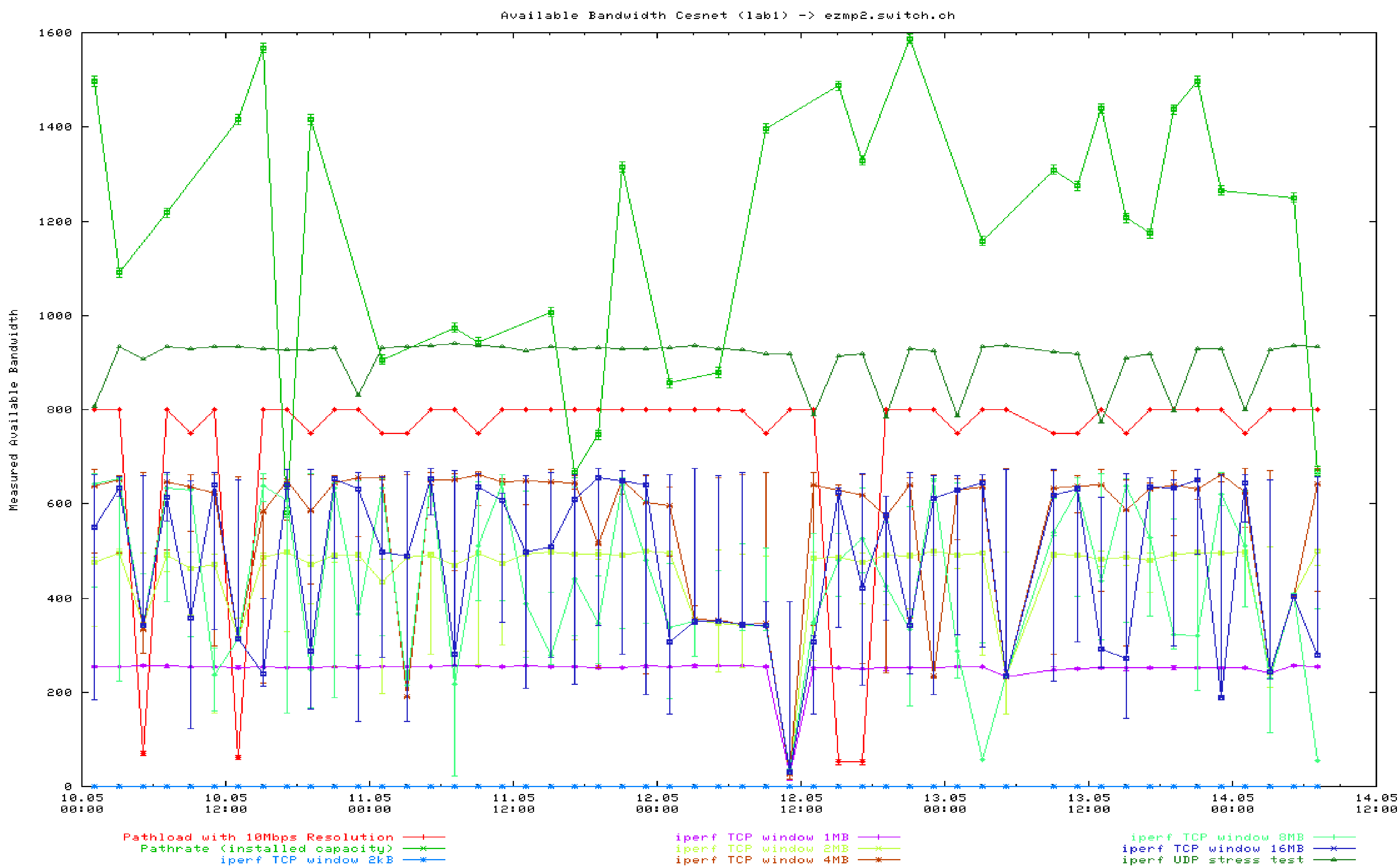
pathrate: lots of fluctuations

UDP iperf: can stress existing traffic

TCP iperf: more fluctuations for larger TCP windows

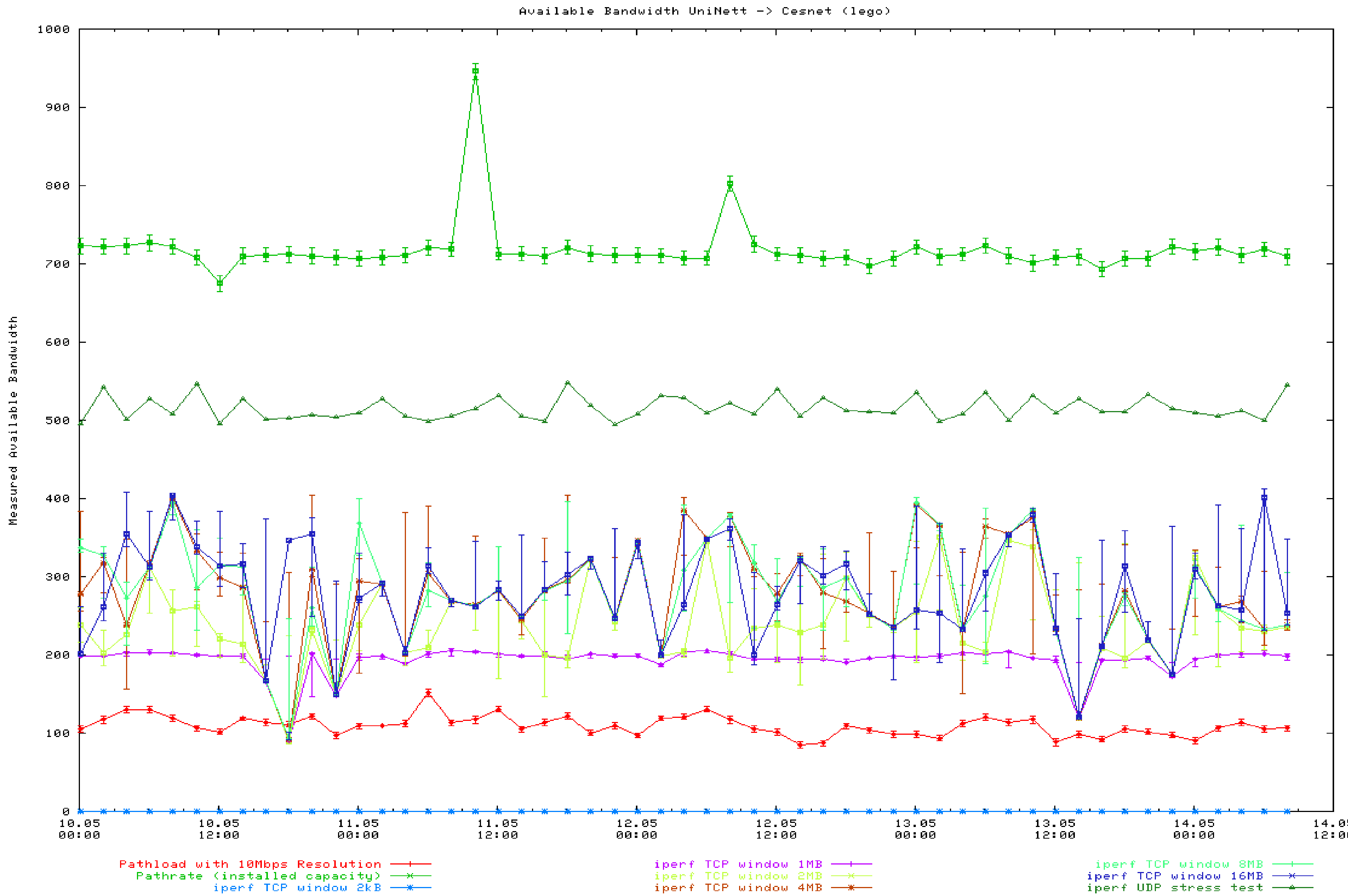
Bandwidth measurement and estimation, cont.

cesnet.cz -> switch.ch



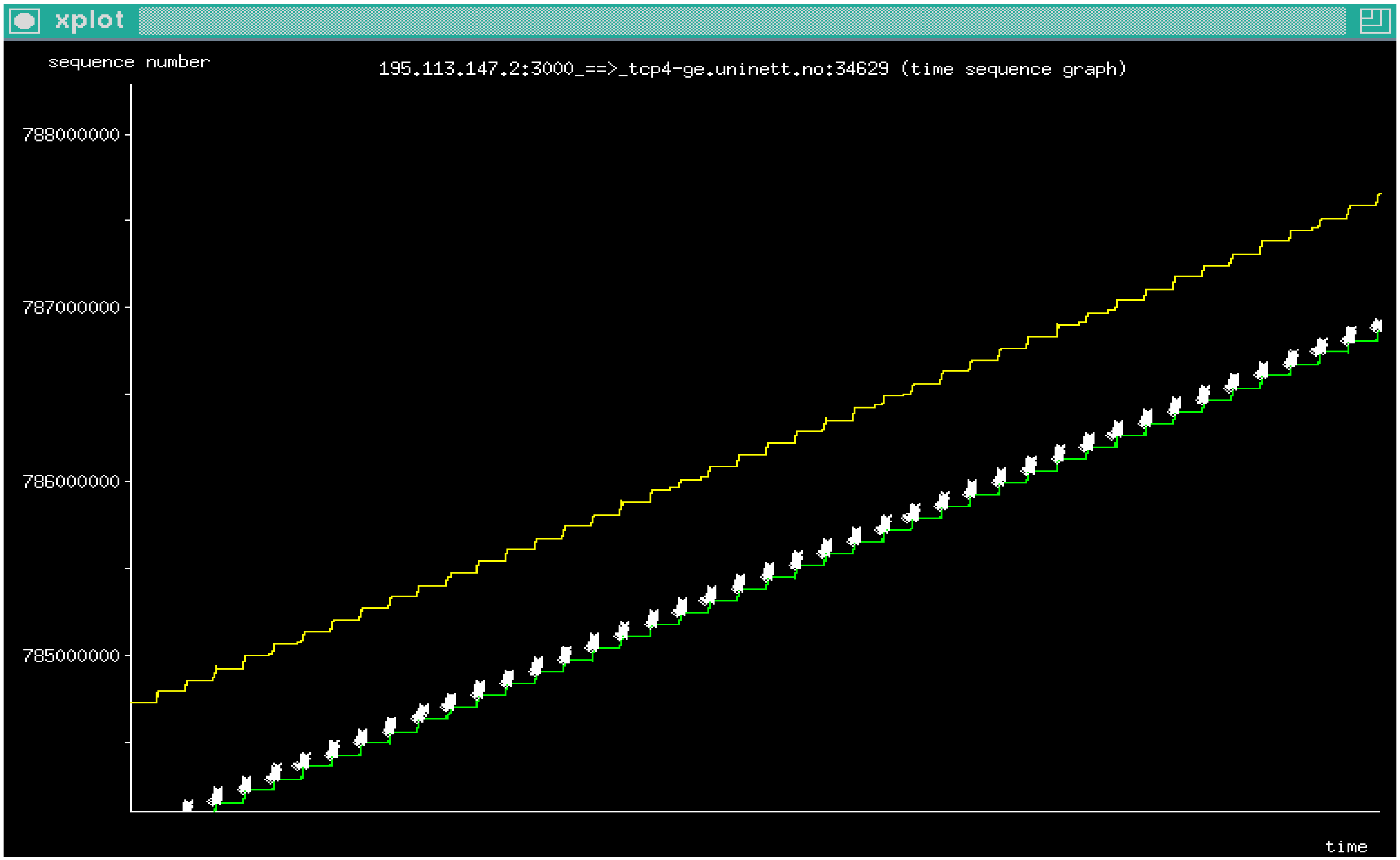
Bandwidth measurement and estimation, cont.

uninett.no -> cesnet.cz



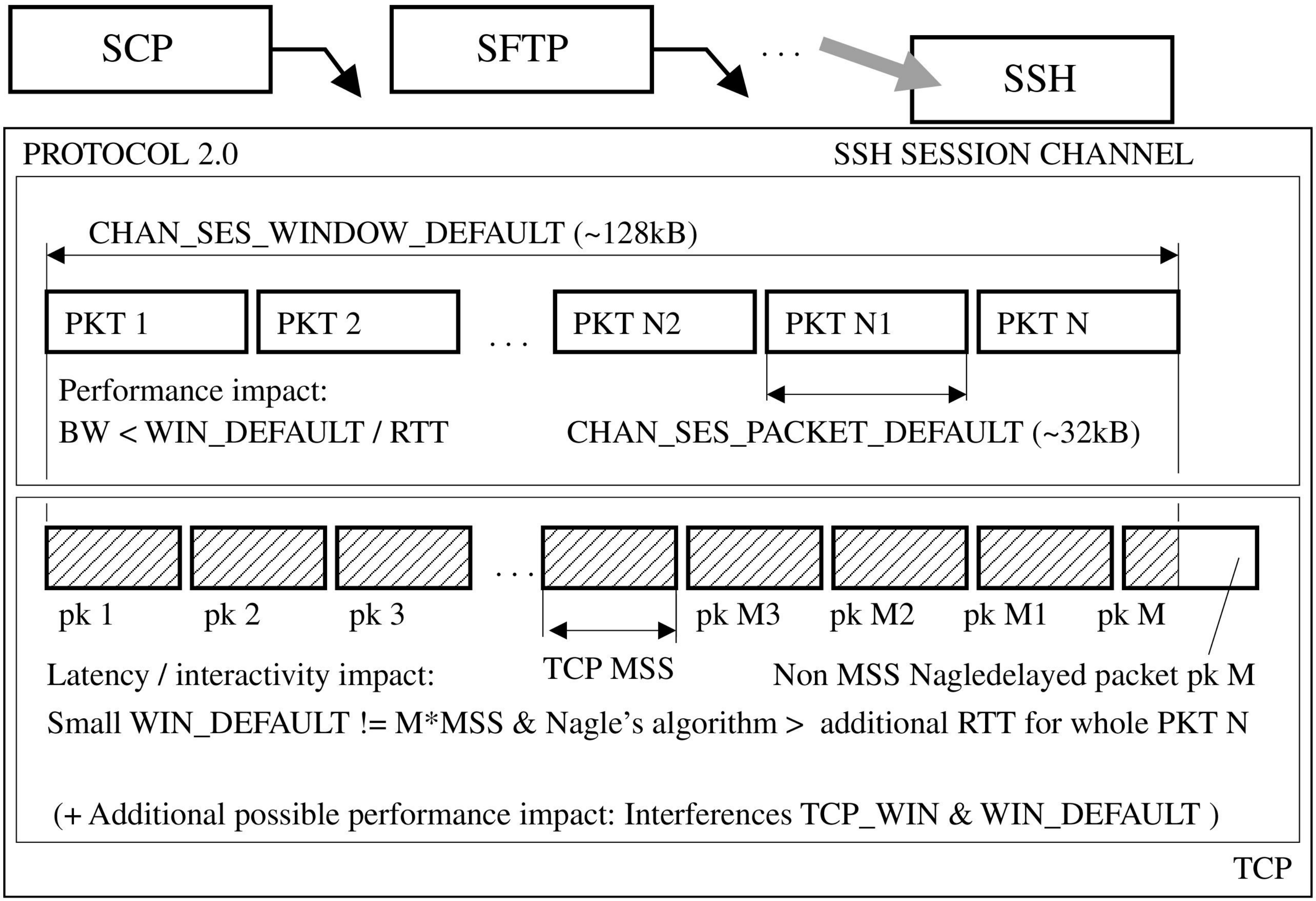
ssh performance

Cesnet -> Uninett, 1.5 MB window, 10.4 Mb/s, 9% load CPU



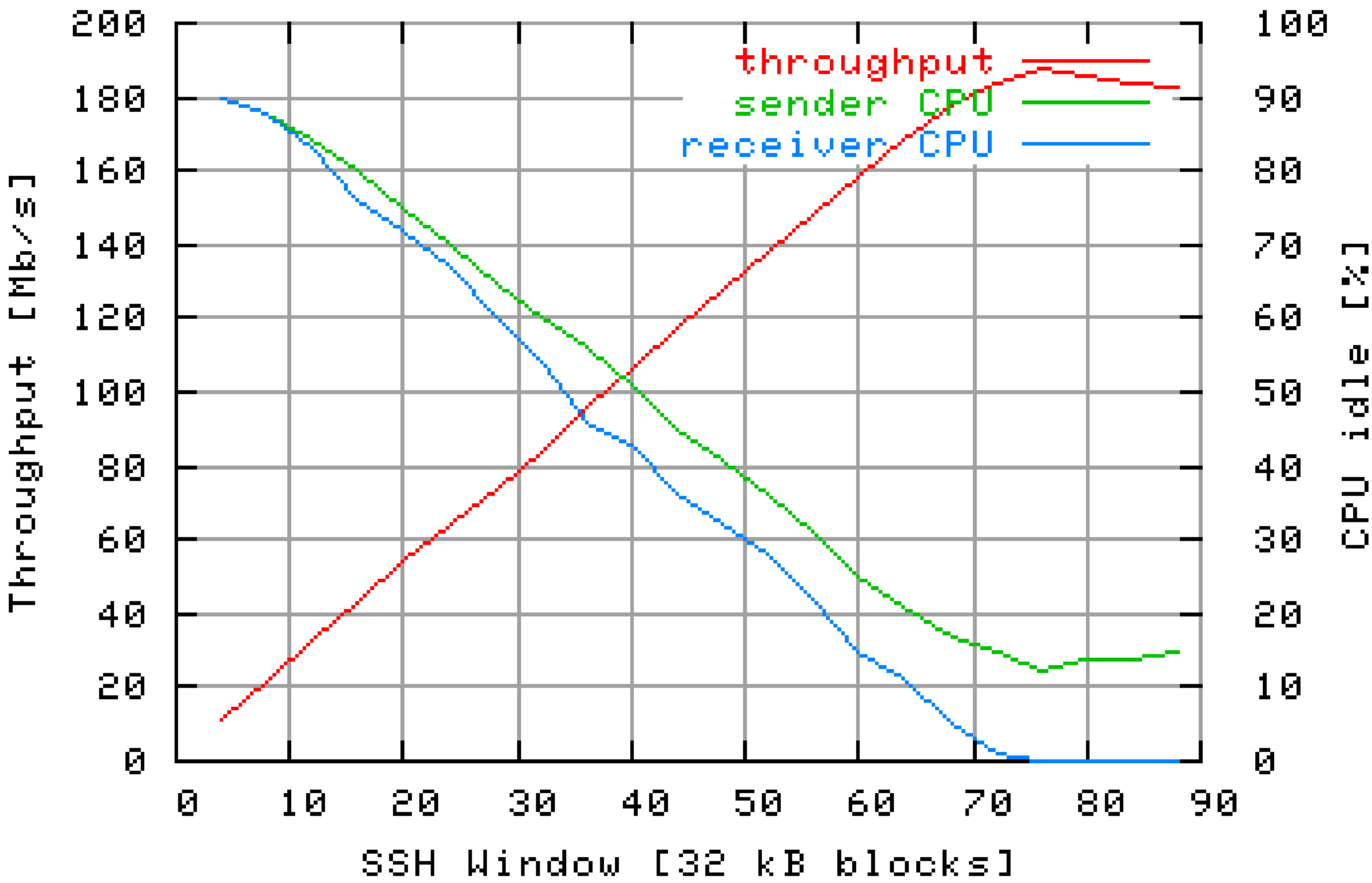
sequence number development - rwin not utilised

ssh performance, cont.



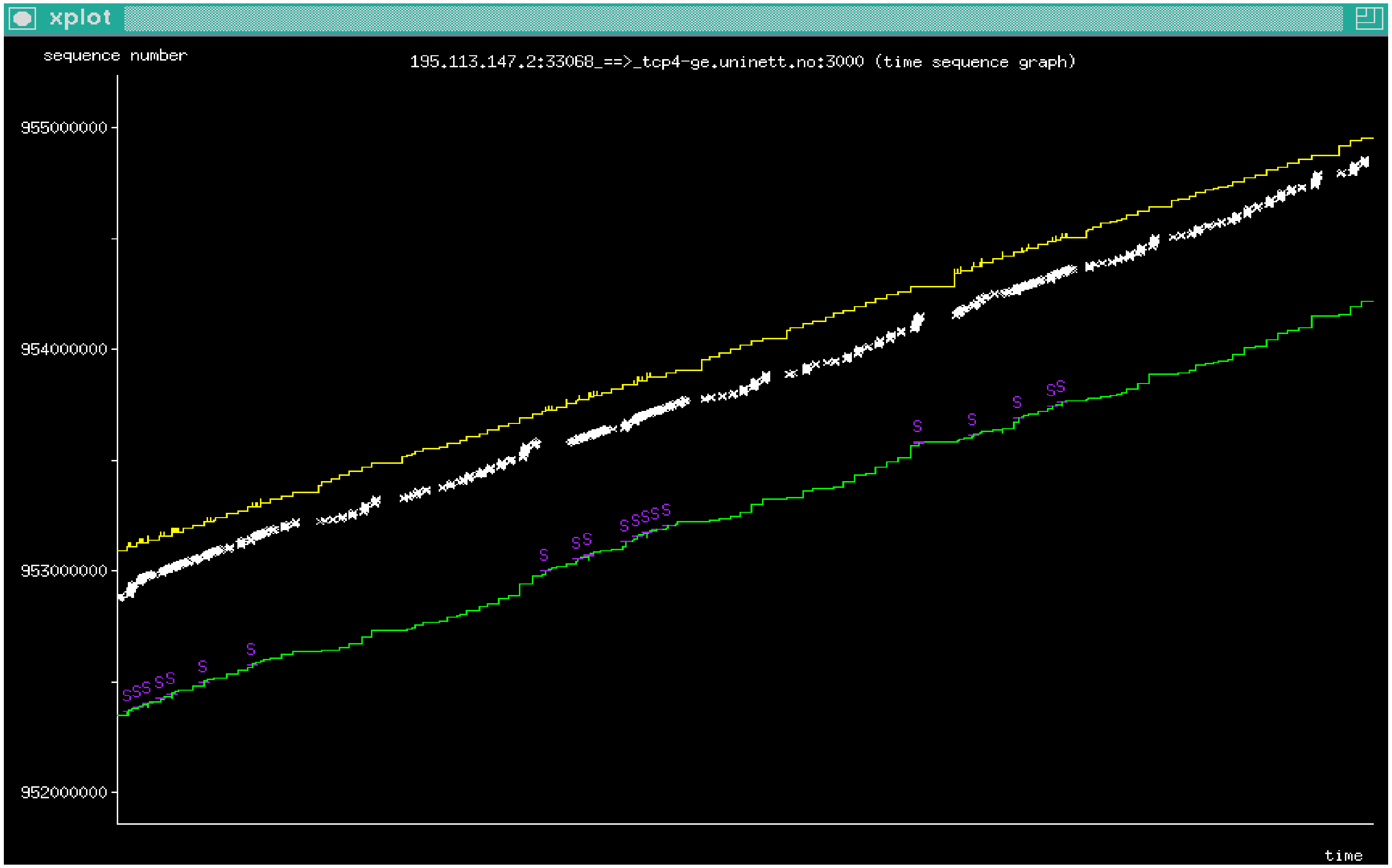
ssh performance, cont.

Bw=1 Gb/s, RTT=45 ms, TCP window=8 MB, Xeon 2.4 GHz



ssh performance, cont.

CHAN_SES_WINDOW_DEFAULT=40 * 32 kB blocks, 85% CPU load



sequence number development - rwin utilised

Conclusion

- Large TCP windows require other configuration tasks for good performance
- Buffer autoconfiguration should not just conserve memory and set TCP windows „large enough“, but also „small enough“
- ssh has a significant performance problem that must be resolved at the application level
- Influence of OS configuration and implementation-specific features on performance can be stronger than amendments in congestion control

Thank you for your attention http://staff.cesnet.cz/~ubik