Performance measuring

Who? Alexandru Giurgiu (alex.giurgiu@os3.nl)
Jeroen Vanderauwera (jeroen.vanderauwera@os3.nl)

From? System and Network Engineering - UvA

When? June 22, 2010

Table of contents

Introduction

Why?

- Performance monitoring is more an art than a science. We like art!
- Pinpointing bottlenecks is hard and not very straightforward.
- Different tools for different purposes, all reporting in their own format.

Introduction

Research questions:

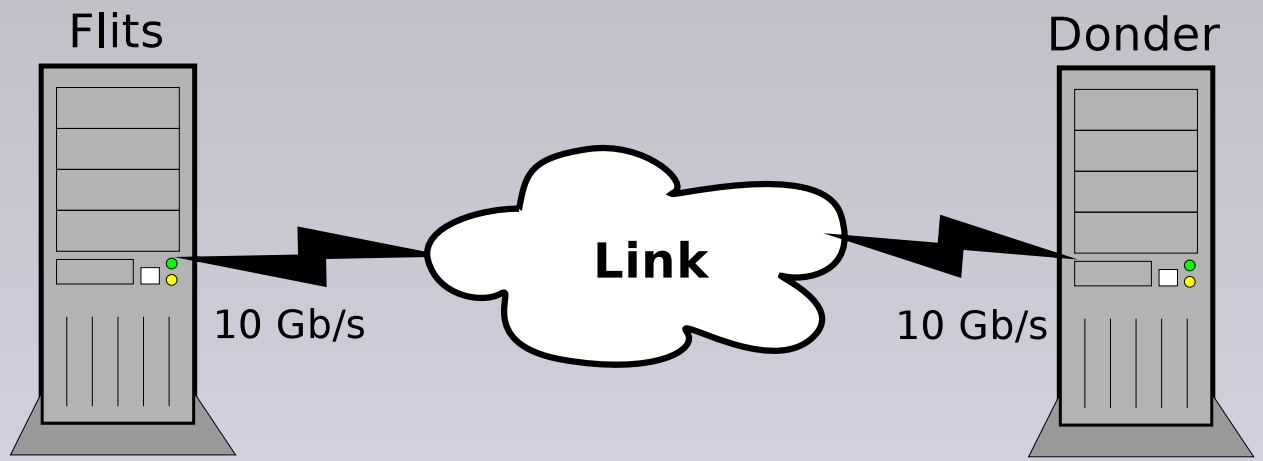
- Is it possible to determine and classify the parameters which affect network performance on the end hosts?
- Is it possible to develop a tool which monitors the parameters and can pinpoint the cause of the reduced network performance?

Introduction

How?

1. Identify and analyze the different hardware and software parameters that influence network performance.
2. Create an application:
   - integrate information from other tools.
   - display it in a clear report that helps pinpointing the problem.

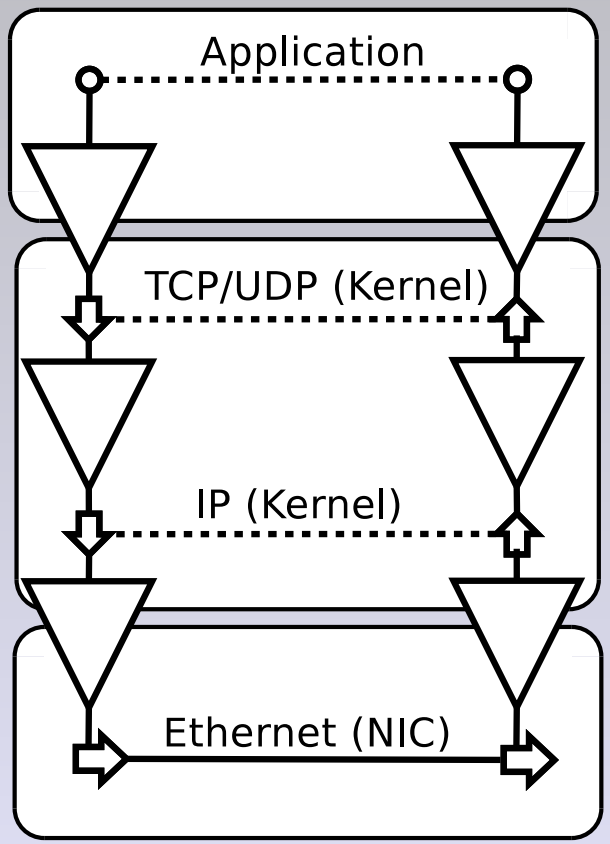
Test environment

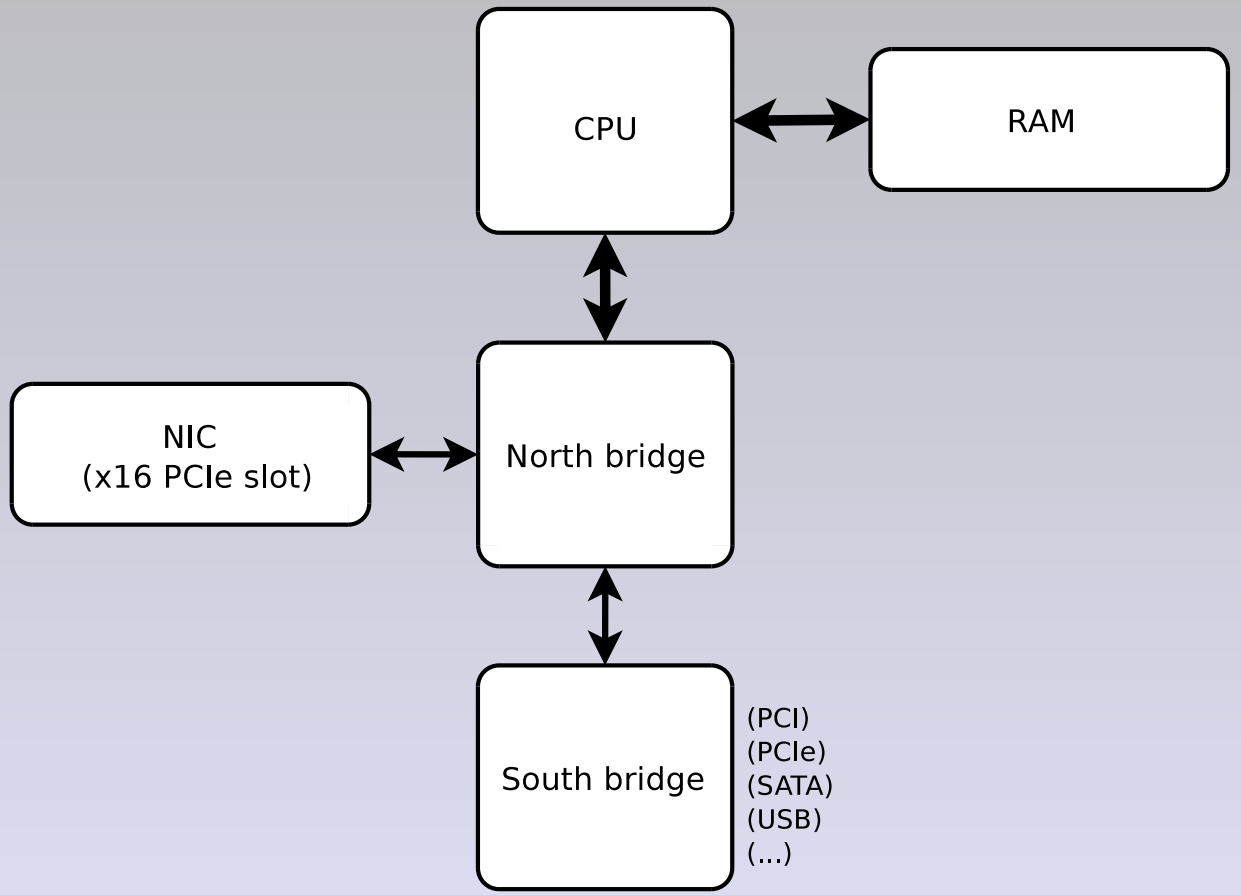


Intel Xeon X5550 2.66 GHz, quad core
6 GB DDR3 1333 MHz
Intel 10 GbE PCIe NIC 82598

Intel Core 2 Duo E6550 2.33 GHz
2 GB DDR2 667 MHz
Intel 10 GbE PCIe 82598

Layered and hardware view





Identified parameters

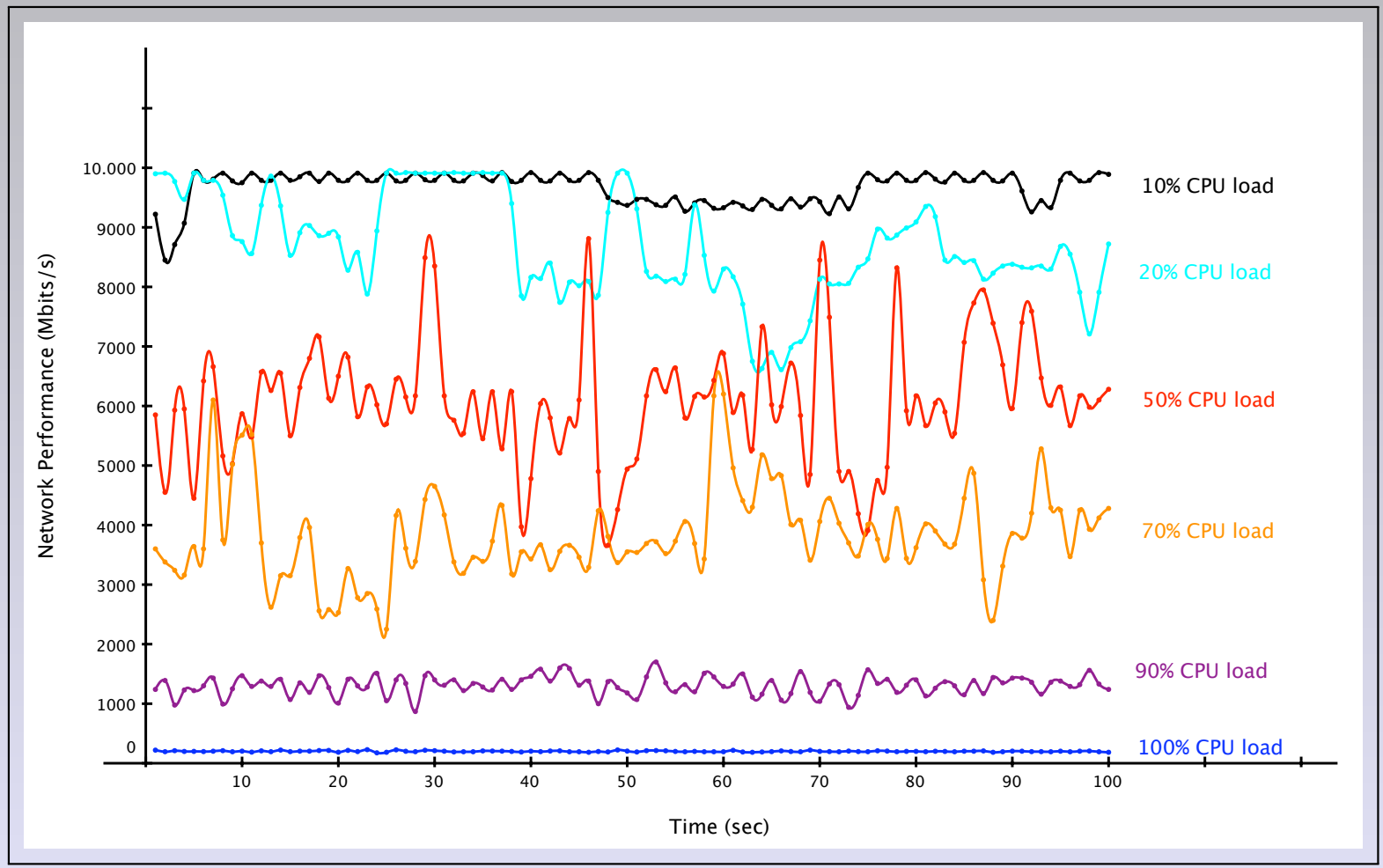
Hardware

- CPU
- Memory
- Network interface
- PCIe bus and slots

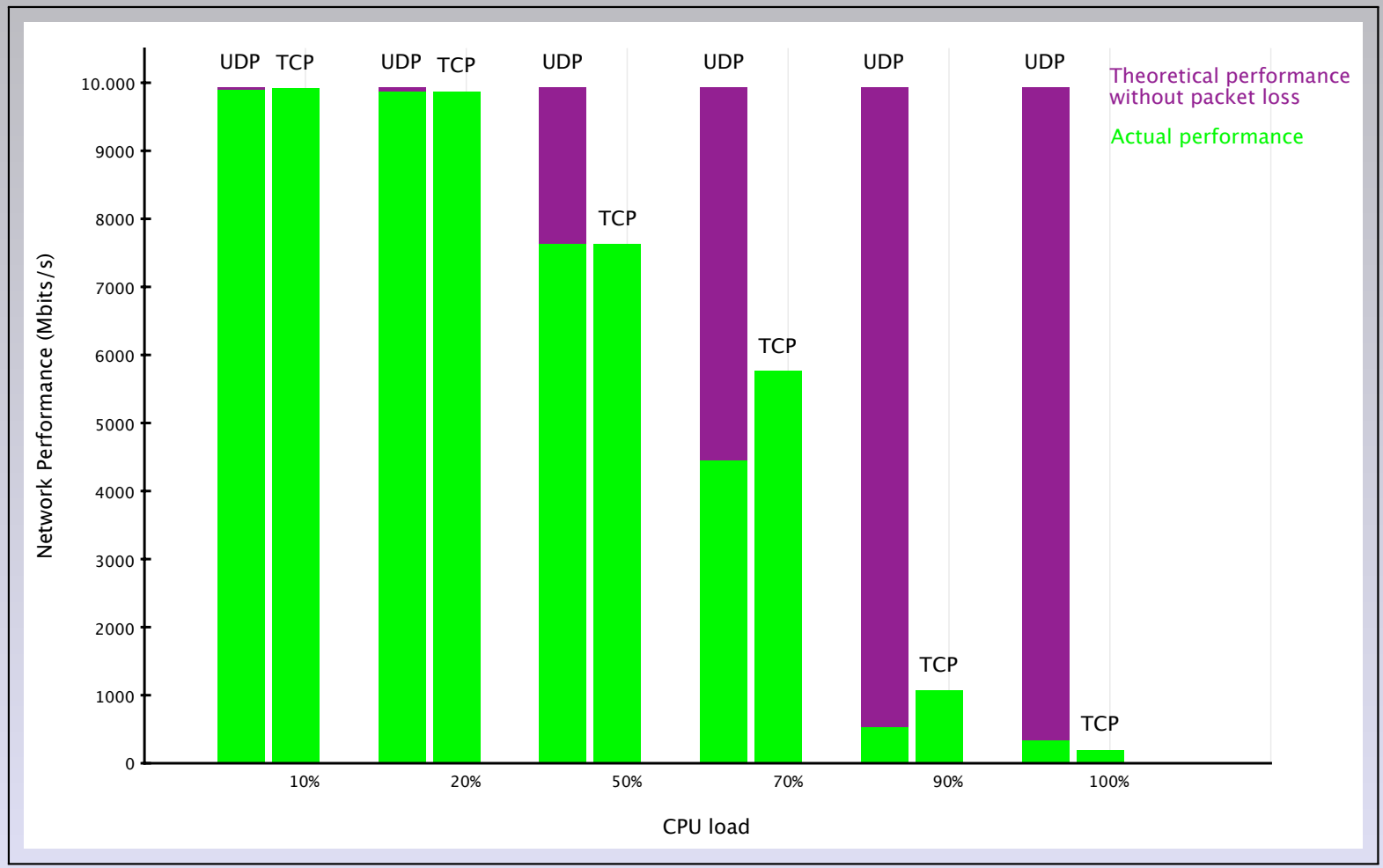
Software

- MTU (Ethernet and IP)
- TCP window size
- Maximum TCP buffer space
- UDP buffer size (per socket and overall)
- Flow control
- TCP Selective Acknowledgements Option

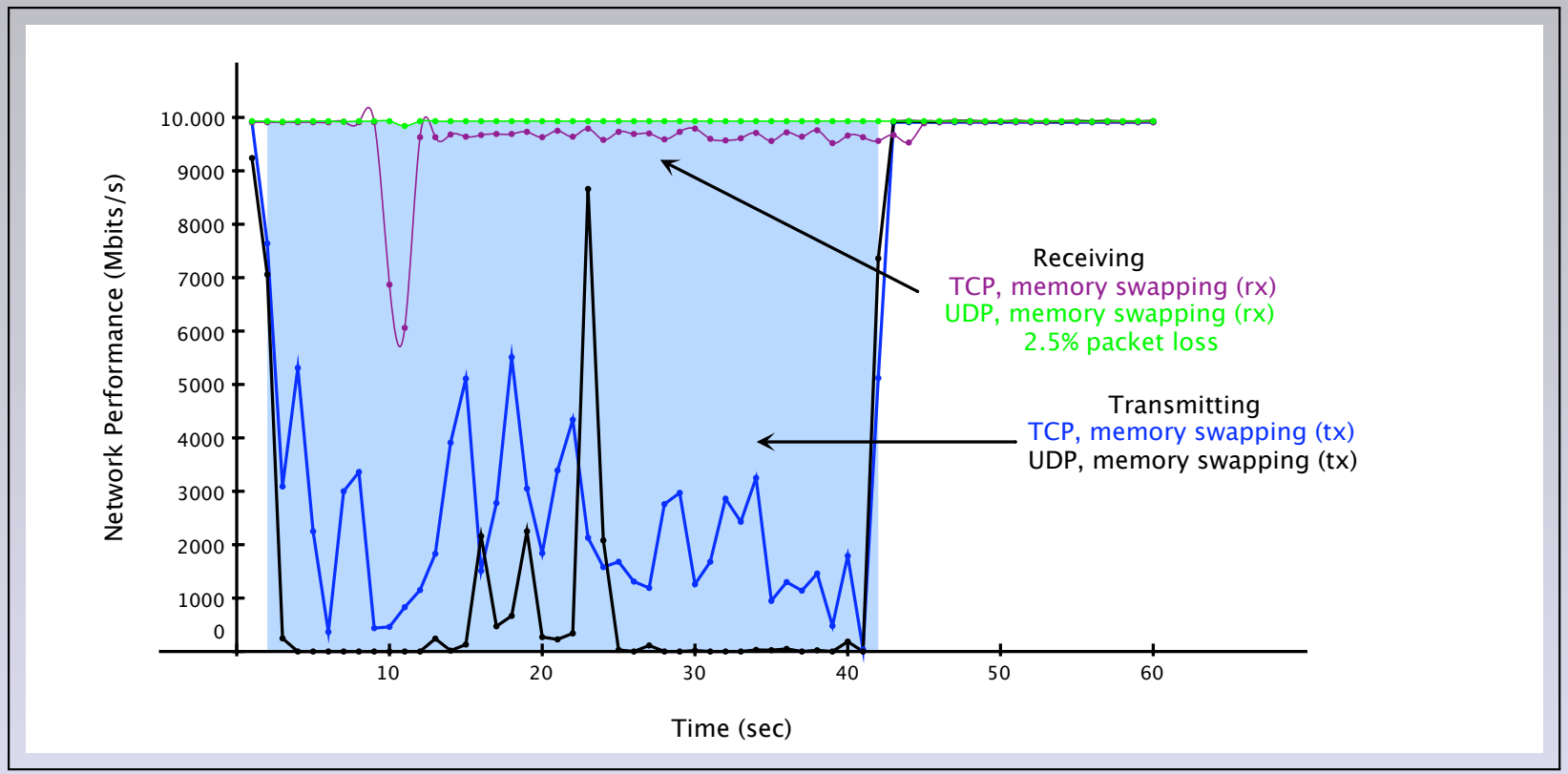
CPU: influence on TCP (sending side)



CPU: receiving side



Memory and swapping

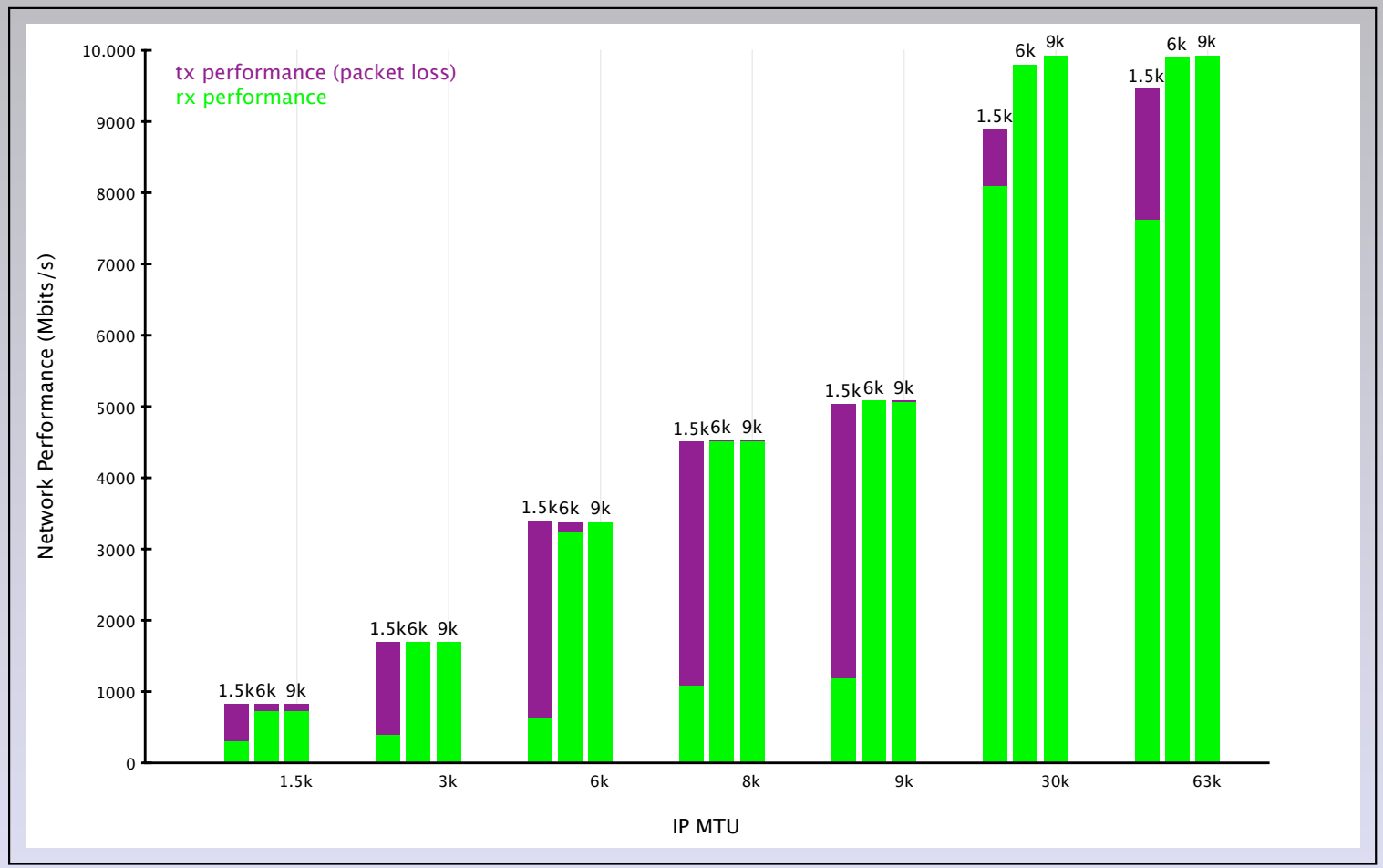


Network interface and bus speed

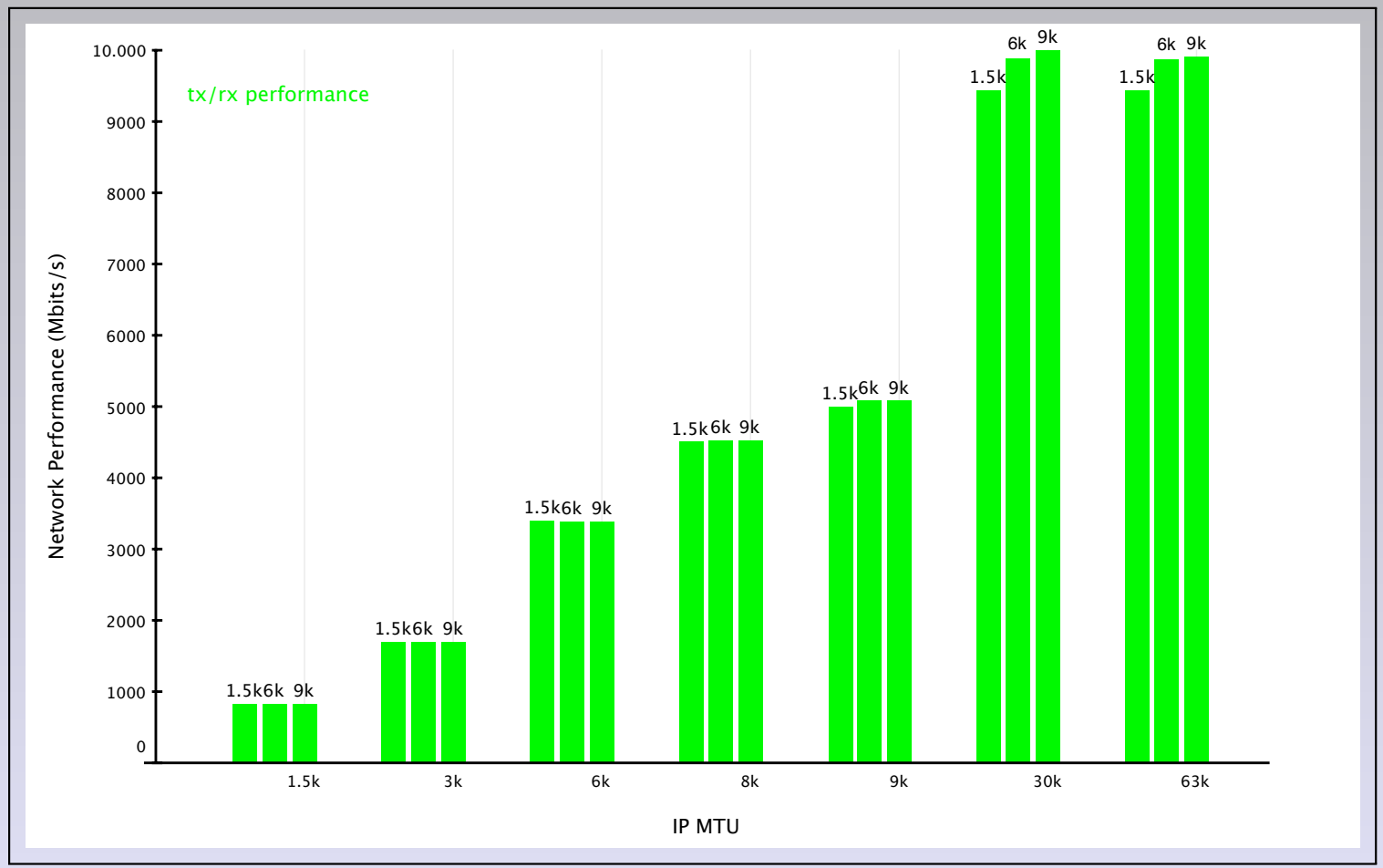
- Obviously throughput cannot exceed the the maximum speed support by the NIC.
- The PCI slot can be a limiting factor (PCIe 2.0 x4 slot or PCIe 1.0 x8 slot required for 10 Gb/s)

	4 lanes	**8 lanes**	**16 lanes**
PCIe 1.0	8 Gb/s	16 Gb/s	32 Gb/s
PCIe 2.0	16 Gb/s	32 Gb/s	64 Gb/s
PCIe 3.0	31.5 Gb/s	63 Gb/s	126 Gb/s

MTU: UDP performance



MTU: TCP performance



TCP window and UDP buffer size

TCP window size

Window size	Network performance
32k	1.14 Gb/s
128k	3.84 Gb/s
512k	9.47 Gb/s
1M	9.91 Gb/s
8M	9.92 Gb/s
128M	9.92 Gb/s
195M (Kernel limit)	9.93 Gb/s

UDP buffer size

UDP buffer size	Network performance	Packet loss
128 kbytes	4.13 Gb/s	44%
512 kbytes	9.93 Gb/s	0%
2 Mbytes	9.93 Gb/s	0%
8 Mbytes	9.93 Gb/s	0%
128 Mbytes	9.75 Gb/s	3.2%

Flow control and SACK

Flow control prevents the overrunning of the receiver end. In our case it did not influence performance.

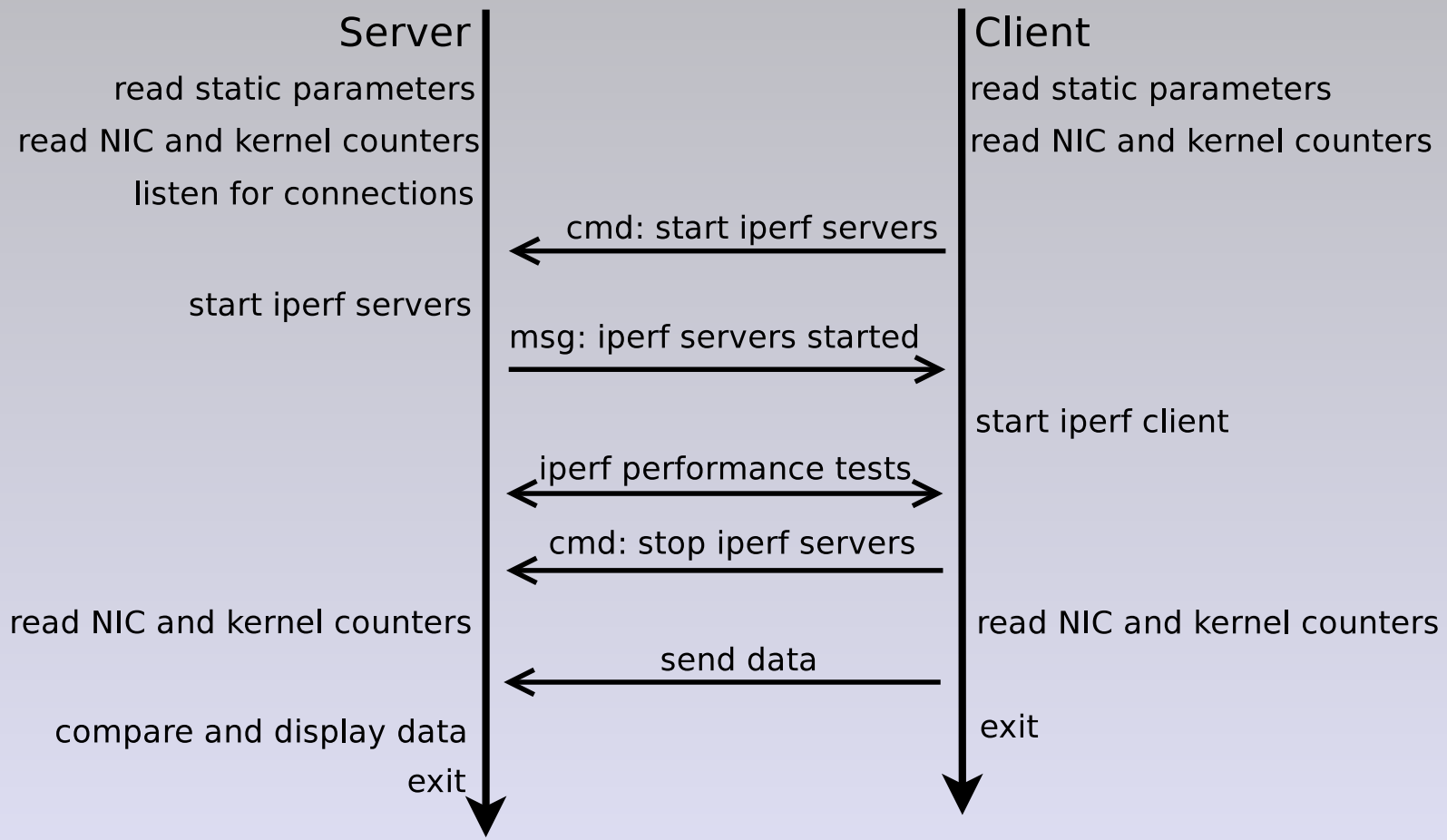
SACK stands for Selective Acknowledgements Option and is a TCP mechanism for improved packet retransmission.

- No difference on our test setup.
- Helps on unreliable links.

The measurement tool

- Written in Ruby(1.9).
- Integrates data from several external tools: ethtool, ifconfig, netstat, dmesg, iperf, etc.
- Client/server model.
- Compares data from both end points side by side.

Tool work flow



Screenshot

	Server(flits)	*Client(donder)*
----------General NIC information----------		
Ip address:	10.0.3.1	10.0.3.2
Link speed:	10000Mb/s	10000Mb/s
Link up [Yes/No]:	yes	yes
Duplex [Full/Half]:	Full	Full
Flow control [On/Off]:	autoneg:on rx:on tx:on	autoneg:on rx:on tx:on
Ethernet MTU:	9000	9000
PCIe slot:	2.5GT/s x8	2.5GT/s x4
----------TCP settings----------		
TCP window size [Initial Default Maximum]:	4096 524287 102400000	4096 524287 102400000
TCP buffer size [Initial Default Maximum]:	572352 763136 1144704	191904 255872 383808
TCP sending buffer [Initial Default Maximum]:	4096 524287 102400000	4096 524287 102400000
TCP SACK [On/Off]:	Off	Off
----------UDP settings----------		
UDP buffer size [Initial Default Maximum]:	572352 763136 1144704	191904 255872 383808
UDP min. receiving buffer:	4096	4096
UDP min. sending buffer:	4096	4096
----------IPERF results----------		
UDP :	3.87 Gbits/sec	3.85 Gbits/sec
UDP loss:	1934/375256	1934/375256
TCP:	3.83 Gbits/sec	3.83 Gbits/sec
----------Packet/byte counters----------		
NIC counter----------		
Packets:	rx:5725576	tx:5725558
	tx:491126	rx:491108

Bytes:	rx:48411519886	tx:48411512340
	tx:26587945	rx:29532462

Errors:	rx:0	tx:0
	tx:0	rx:0
Kernel counter----------		
Packets:	rx:5725576	tx:5725558
	tx:491126	rx:491108

Errors:	rx:0	tx:0
	tx:0	rx:0

Conclusions

The parameters

- A large array of complex parameters that influence network performance.
- Some influence throughput while other influence packet loss.
- Application design is very important.
- In our case the receiving side was less influenced by CPU load and swapping.
- Default settings for the major OSs are inappropriate for high performance networking.

The tool

- Highly dynamic environment makes it hard to pinpoint the problem.
- Works good on the static parameters but hard to make it reliable on the dynamic ones.

Questions

Are there any?