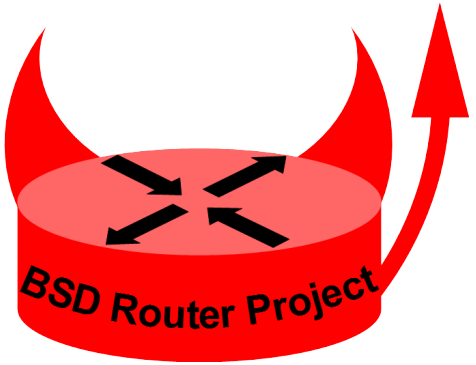
AsiaBSDcon 2018

# Tuning FreeBSD for routing and firewalling

Olivier Cochard-Labbé

# whoami(1)

- olivier.cochard@ 

- olivier@ 


FreeBSD®

# Benchmarking a router

- Router job: Forward packets between its interfaces at maximum rate
- Reference value:**Packet Forwarding Rate** in packets-per-second (**pps**) unit
  - **NOT** a bandwidth (in bit-per-second unit)
- RFC 2544: Benchmarking Methodology for Network Interconnect Devices

# Some Line-rate references

- Gigabit line-rate: 1.48M frames-per-second
- 10 Gigabit line rate: 14.8M frames-per-second
- Small packets: 1 frame = 1 packet
- Gigabit Ethernet is a **full duplex** media:
  - A **line-rate Gigabit** router MUST be able to receive AND transmit in the same time, then to forward at **3Mpps**

# I want bandwidth values!

- Packets-per-second * Packets-size
- Estimated using Simple Internet Mix (IMIX) packet size trimodal reference distribution
- IPv4 layer in bits-per-second:

$$PPS \cdot \left( \frac{7 \cdot 40 + 4 \cdot 576 + 1500}{12} \right) \cdot 8$$

- Ethernet layer, add 14 bytes (switch counters):

$$PPS \cdot \left( \frac{7 \cdot 54 + 4 \cdot 590 + 1514}{12} \right) \cdot 8$$

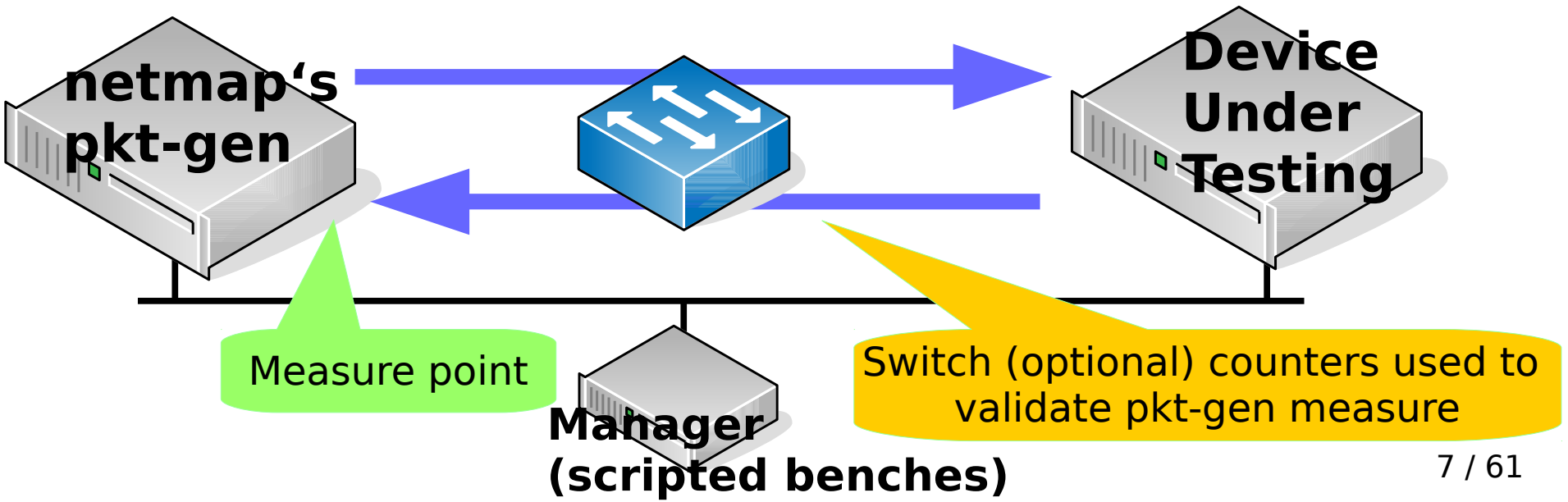
- *Since about 2004, Internet packets size distribution is bimodal (44% less than 100B and 37% more than 1400B in 2006)*

# Minimum router‘s performance

| Link speed | Line-rate router | Full-duplex **line-rate router** | Minimum rate, using IMIX distribution for reaching link speed | Full-duplex minimum IMIX **link speed router** |
|---|---|---|---|---|
| 1Gb/s | 1.48 Mpps | 3 Mpps | 350 Kpps | **700 Kpps** |
| 10Gb/s | 14.8 Mpps | 30 Mpps | 3.5 Mpps | **7 Mpps** |

# Simple benchmark lab

- As a telco we measure the worse case (Denial-of-Service):
  - Smallest packet size
  - Maximum link rate



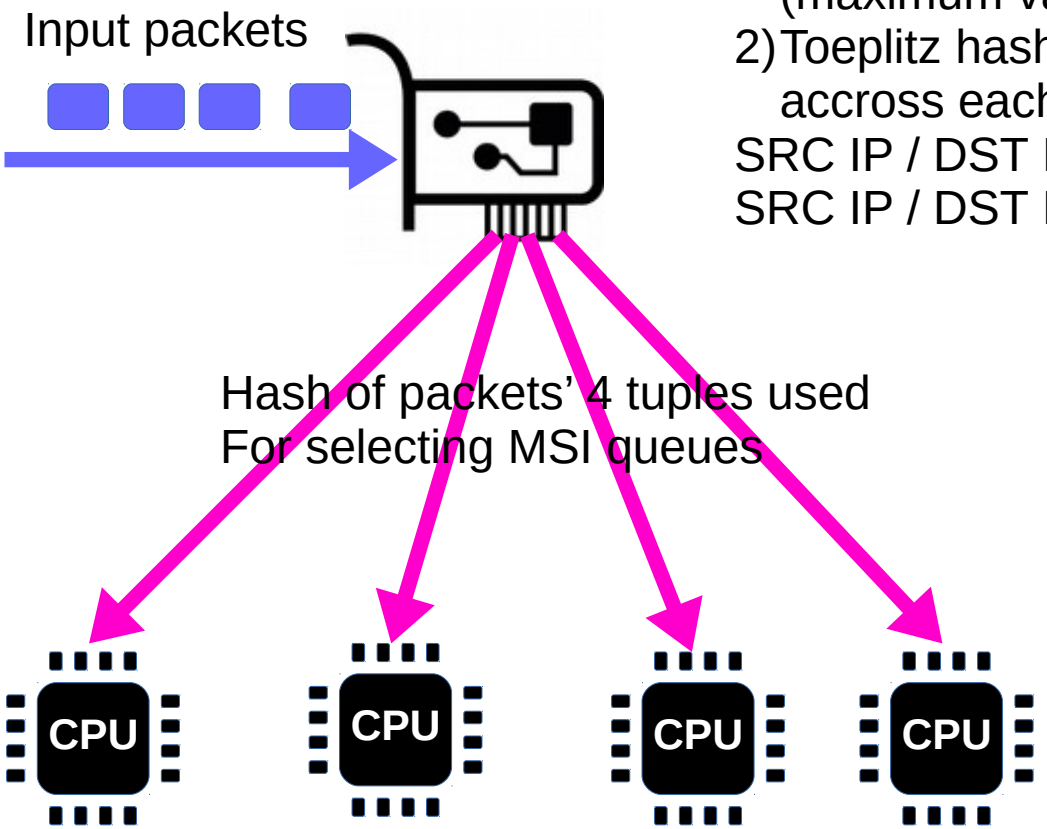
# Hardware details

| Servers | CPU | cores | GHz | Network card (driver name) |
|---|---|---|---|---|
| Dell PowerEdge R630 | Intel **E5-2650** v4 | 2x12x2 | 2.2 | 10G Intel 82599ES (**ixgbe**)<br>10G Chelsio T520-CR (**cxgbe**)<br>10G Mellanox ConnectX-3 Pro (**mlx4en**)<br>10-50G Mellanox ConnectX-4 LX (**mlx5en**) |
| HP ProLiant DL360p Gen8 | Intel **E5-2650** v2 | 8x2 | 2.6 | 10G Chelsio T540-CR (**cxgbe**)<br>10G Emulex OneConnect be3 (**oce**) |
| SuperMicro 5018A-FTN4 | Intel Atom **C2758** | 8 | 2.4 | 10G Chelsio T540-CR (**cxgbe**) |
| SuperMicro 5018A-FTN4 | Intel Atom **C2758** | 8 | 2.4 | 10G Intel 82599 (**ixgbe**) |
| Netgate RCC-VE 4860 | Intel Atom **C2558** | 4 | 2.4 | Gigabit Intel i350 (**igb**) |
| PC Engines APU2 | AMD **GX-412**TC | 4 | 1 | Gigabit Intel i210AT (**igb**) |

No 16 cores-in-one-socket CPU

Same DAC for all 10G: QFX-SFP-DAC-3M

# Multi-queue NIC & RSS




1) NIC drivers creates one queue per core detected (maximum values are drivers dependent)
2) Toeplitz hash used for balancing received packets accross each queues.

SRC IP / DST IP / SRC PORT / DST PORT (4 tuples)
SRC IP / DST IP (2 tuples)

# Multi-queue NIC & RSS

1) Needs multiple flows

- Local tunnel (IPSec, GRE,…) presents only one flow: Performance problem with 1G home fiber ISP using PPPoE as example

2) Needs multi-CPUs

- Benefit of physical cores vs logical cores (Hyper Threading) vs multiple socket ?

# Monitoring queues usage

- Python script from melifaro@ parsing sysctl NIC stats (RX queue mainly)
- Support: bxe, cxl, ix, ixl, igb, mce, mlxen and oce

https://github.com/ocochard/BSDRP/blob/master/BSDRP/Files/usr/local/bin/nic-queue-usage

```
[root@hp]~# nic-queue-usage cxl0
[Q0  856K/s] [Q1  862K/s] [Q2  846K/s] [Q3  843K/s] [Q4  843K/s] [Q5  843K/s] [Q6  861K/s] [Q7  854K/s] [QT  6811K/s 16440K/s -> 13K/s]
[Q0  864K/s] [Q1  871K/s] [Q2  853K/s] [Q3  857K/s] [Q4  856K/s] [Q5  855K/s] [Q6  871K/s] [Q7  859K/s] [QT  6889K/s 16670K/s -> 13K/s]
[Q0  843K/s] [Q1  851K/s] [Q2  834K/s] [Q3  835K/s] [Q4  836K/s] [Q5  836K/s] [Q6  858K/s] [Q7  854K/s] [QT  6750K/s 16238K/s -> 13K/s]
[Q0  844K/s] [Q1  846K/s] [Q2  826K/s] [Q3  824K/s] [Q4  825K/s] [Q5  823K/s] [Q6  843K/s] [Q7  837K/s] [QT  6671K/s 16168K/s -> 12K/s]
[Q0  832K/s] [Q1  847K/s] [Q2  828K/s] [Q3  829K/s] [Q4  830K/s] [Q5  832K/s] [Q6  849K/s] [Q7  842K/s] [QT  6692K/s 16105K/s -> 13K/s]
[Q0  867K/s] [Q1  874K/s] [Q2  855K/s] [Q3  855K/s] [Q4  854K/s] [Q5  853K/s] [Q6  869K/s] [Q7  855K/s] [QT  6885K/s 16609K/s -> 13K/s]
[Q0  826K/s] [Q1  831K/s] [Q2  814K/s] [Q3  811K/s] [Q4  814K/s] [Q5  813K/s] [Q6  832K/s] [Q7  833K/s] [QT  6578K/s 15831K/s -> 12K/s]
```

Summary of all queues

Global NIC RX counter

Global NIC TX counter

# Hyper-threading & cxgbe

```
CPU: Intel Xeon CPU E5-2650 v2 @ 2.60GHz (2593.81-MHz K8-class CPU)
(…)
FreeBSD/SMP: Multiprocessor System Detected: 16 CPUs
FreeBSD/SMP: 1 package(s) x 8 core(s) x 2 hardware threads
(…)
cxl0: <port 0> numa-domain 0 on t5nex0
cxl0: Ethernet address: 00:07:43:2e:e4:70
cxl0: 16 txq, 8 rxq (NIC); 8 txq, 2 rxq (TOE)
cxl1: <port 1> numa-domain 0 on t5nex0
cxl1: Ethernet address: 00:07:43:2e:e4:78
cxl1: 16 txq, 8 rxq (NIC); 8 txq, 2 rxq (TOE)
```

cxgbe doesn‘t use all CPUs by default if CPU>8

# Hyper-threading & cxgbe

- Config 1: default (8 rx queues)
- Config 2: 16 rx queues to use ALL 16 CPUs
  - `hw.cxgbe.nrxq10g=16`
- Config 3: disabling HT (8 rx queues)
  - `machdep.hyperthreading_allowed=0`

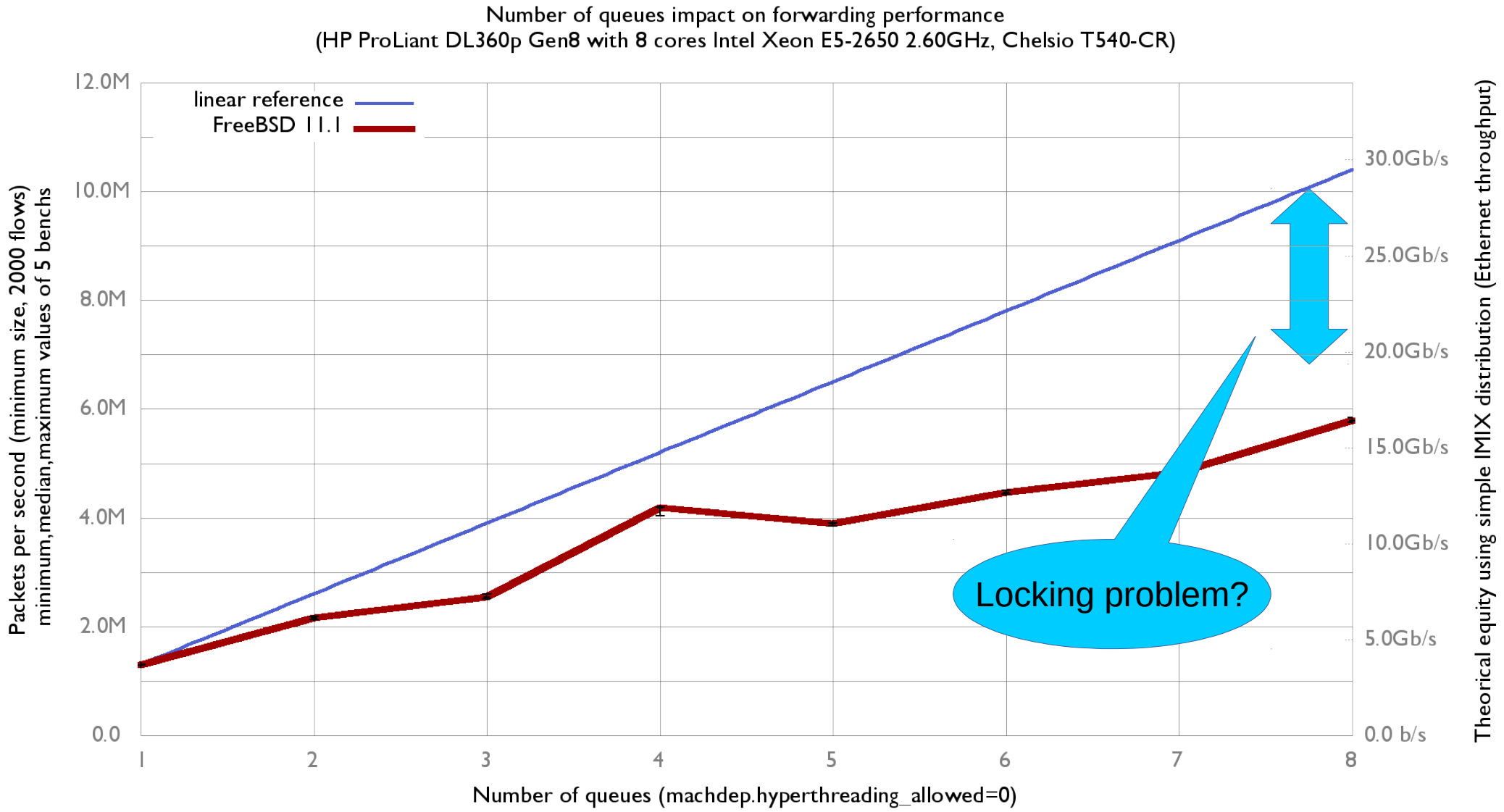
- `FreeBSD 11.1-RELEASE amd64`

# Disabling Hyper-Threading

ministat(1) is my friend

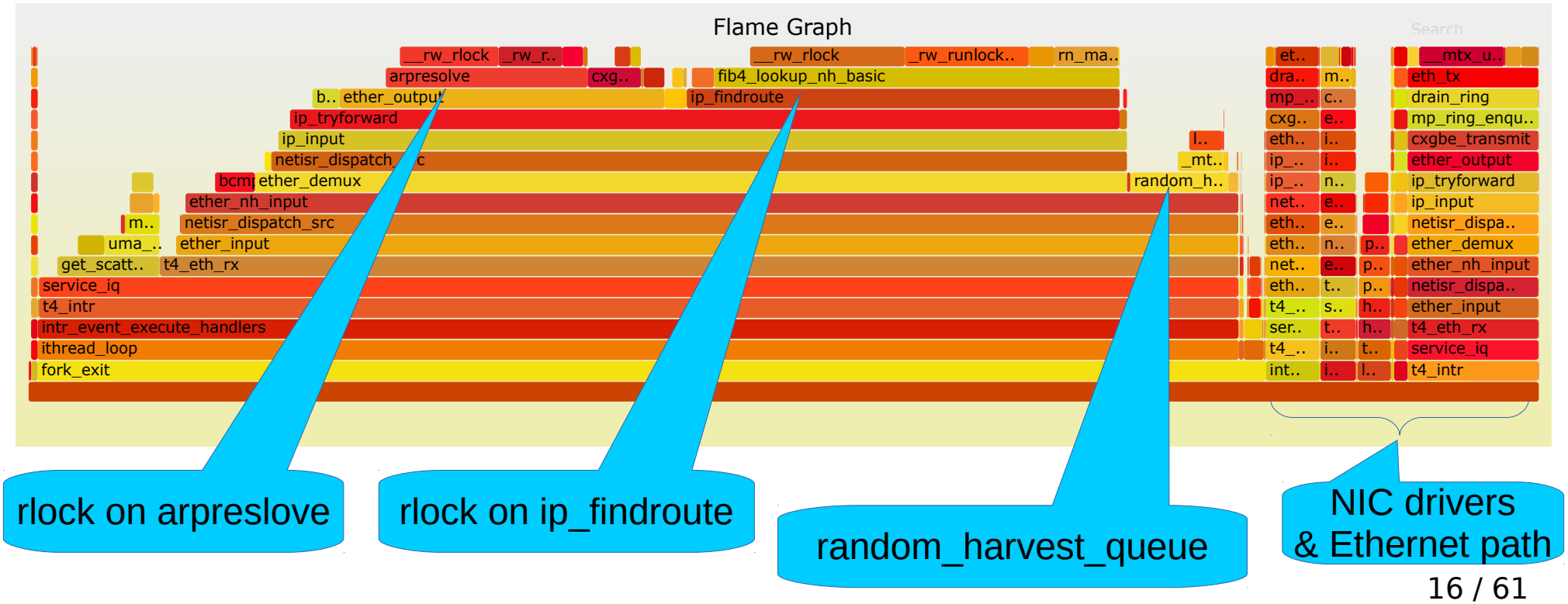
```
x Xeon E5-2650v2 & cxgbe, HT-enabled & 8rxq(default): inet4 packets-per-second
+ Xeon E5-2650v2 & cxgbe, HT-enabled & 16rxq: inet4 packets-per-second
* Xeon E5-2650v2 & cxgbe, HT-disabled & 8rxq: inet4 packets-per-second
+--------------------------------------------------------------------------+
|                                                                       **|
|x      xx    x           +       + + +  +                              ***|
|   |____A_____|                                                           |
|                          |_____AM____|                                  |
|                                                                      |A||
+--------------------------------------------------------------------------+
    N           Min           Max        Median           Avg        Stddev
x   5       4500078       4735822       4648451     4648293.8     94545.404
+   5       4925106       5198632       5104512     5088362.1     102920.87
Difference at 95.0% confidence
        440068 +/- 144126
        9.46731% +/- 3.23827%
        (Student's t, pooled s = 98821.9)
*   5       5765684     5801231.5       5783115     5785004.7     13724.265
Difference at 95.0% confidence
        1.13671e+06 +/- 98524.2
        24.4544% +/- 2.62824%
        (Student's t, pooled s = 67554.4)
```

# Queues/cores impact



# Analysing bottleneck

```
kldload hwpmc
pmcstat -S CPU_CLK_UNHALTED_CORE -l 20 -O data.out
stackcollapse-pmc.pl data.out > data.stack
flamegraph.pl data.stack > data.svg
```



# Random harvest sources

```
~# sysctl kern.random.harvest
kern.random.harvest.mask_symbolic: [UMA],
[FS_ATIME],SWI,INTERRUPT,NET_NG,NET_ETHER,NET_TUN,MOUSE,KEYBOARD,
ATTACH,CACHED
kern.random.harvest.mask_bin: 00111111111
kern.random.harvest.mask: 511
```

- Config 1: default
- Config 2: Do not use INTERRUPT neither NET_ETHER as entropy sources

```
harvest_mask="351"
```

Security impact regarding the random generator

# kern.random.harvest.mask

| Setup<br>CPU (cores) & NIC | 511 (default)<br>Median of 5 | 351<br>Median of 5 | ministat |
|---|---|---|---|
| E5-2650v4 (2x12) & ixgbe<br>Xeon & Intel 82599ES | 3.74 Mpps | 3.78 Mpps | No diff. proven at 95.0% confidence |
| E5-2650v4 (2x12) & cxgbe<br>Xeon & Chelsio T520 | 4.82 Mpps | 4.87 Mpps | No diff. proven at 95.0% confidence |
| E5-2650v4 (2x12) & ml4en<br>Xeon & Mellanox ConnectX-3 Pro | 3.49 Mpps | 3.92 Mpps | 11.66% +/- 8.15% |
| E5-2650v4 (2x12) & ml5en<br>Xeon & Mellanox ConnectX-4 Lx | 0 Mpps | 0 Mpps | System Overloaded |
| E5-2650v2 (8) & cxgbe<br>Xeon & Chelsio T540 | 5.76 Mpps | 5.79 Mpps | No diff. proven at 95.0% confidence |
| E5-2650v2 (8) & oce<br>Xeon & Emulex be3 | 1.33 Mpps | 1.33 Mpps | No diff. proven at 95.0% confidence |
| C2758 (8) & cxgbe<br>Atom & Chelsio T540 | 2.83 Mpps | 3.17 Mpps | 12.52% +/- 1.82% |
| C2758 (8) & ixgbe<br>Atom & Intel 82599ES | 2.3 Mpps | 2.43 Mpps | 6.14% +/- 1.84% |
| C2558 (4) & igb<br>Atom & Intel I354 | 951 Kpps | 1 Mpps | 4.75% +/- 1.08% |
| GX412 (4) & igb<br>AMD & Intel I210 | 726 Kpps | 749 Kpps | 3.14% +/- 0.70% |

| | |
|---|---|
| 10Gb/s full duplex IMIX | 7 Mpps |
| 1Gb/s full duplex IMIX | 700 Kpps |

Tips 2: harvest_mask="351"

# arpresolve & ip_findroute

- Yandex contributions (melifaro@ & ae@)
- Published January 2016: projects/routing

https://wiki.freebsd.org/ProjectsRoutingProposal

- Patches refreshed for FreeBSD 12-head:

https://people.freebsd.org/~ae/afdata.diff

https://people.freebsd.org/~ae/radix.diff

- Patches backported to FreeBSD 11.1:

https://people.freebsd.org/~olivier/fbsd11.1.ae.afdata-radix.patch

# Yandex‘s patches

| setup | 11.1 | 11.1-Yandex | ministat |
|---|---|---|---|
| E5-2650v4 (2x12) & ixgbe<br>Xeon & Intel 82599ES | 3.78 Mpps | 6.46 Mpps | 73.58% +/- 7.3% |
| E5-2650v4 (2x12) & cxgbe<br>Xeon & Chelsio T520 | 4.87 Mpps | 9.60 Mpps | 95.36% +/- 3.8% |
| E5-2650v4 (2x12) & mlx4en<br>Xeon & Mellanox ConnectX-3 Pro | 3.92 Mpps | 8.01 Mpps | 100.5% +/- 15.6% |
| E5-2650v4 (2x12) & mlx5en<br>Xeon & Mellanox ConnectX-4 Lx | 0 Mpps | 14.64 Mpps | *NA* |
| E5-2650v2 (8) & cxgbe<br>Xeon & Chelsio T540 | 5.75 Mpps | 10.9 Mpps | 90.56% +/- 1.24 |
| E5-2650v2 (8) & oce<br>Xeon & Emulex be3 | 1.33 Mpps | 1.33 Mpps | No diff. proven at 95.0% confidence |
| C2758 (8) & cxgbe<br>Atom & Chelsio T540 | 3.15 Mpps | 4.2 Mpps | 34.4% +/- 2.9% |
| C2758 (8) & ixgbe<br>Atom & Intel 82599ES | 2.43 Mpps | 3.08 Mpps | 26% +/- 1.18 |
| C2558 (4) & igb<br>Atom & Intel I354 | 1 Mpps | 1.2 Mpps | 20.17% +/- 2.56% |
| GX412 (4) & igb<br>AMD & Intel I210 | 747 Kpps | 729 Kpps | -2.37% +/- 0.58% |

| | |
|---|---|
| 10Gb/s full duplex IMIX | 7 Mpps |
| 1Gb/s full duplex IMIX | 700 Kpps |

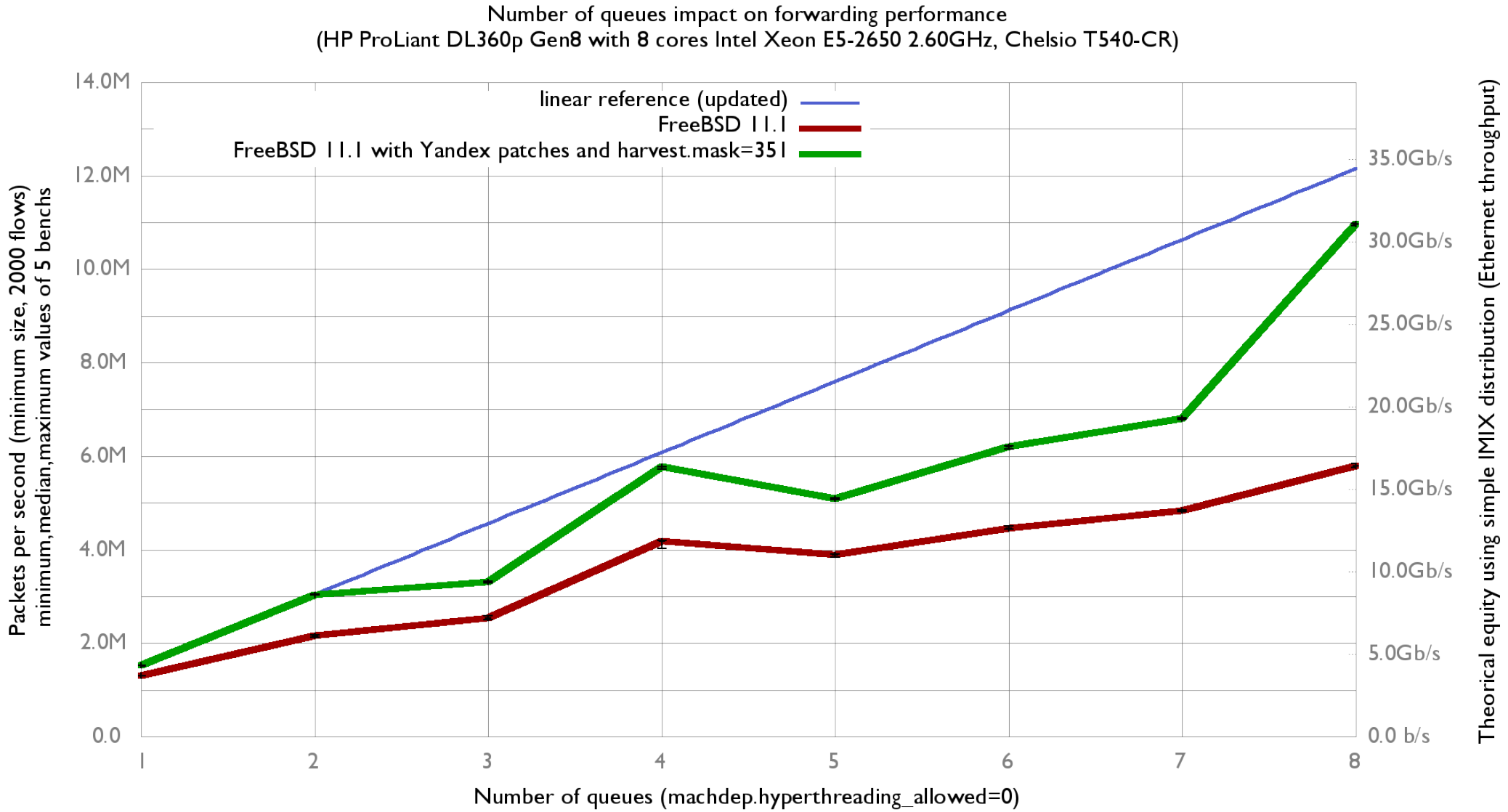
Tips 3: Use steroid patches from Russia

# Avoid some NIC

- 10G Emulex OneConnect (be3)
  - No configurable number of rx/tx queues (4)
  - No configurable Ethernet Flow control
  - 1.33Mpps is not even a gigabit line-rate

Tips 4: Use good NIC (Mellanox, Chelsio, Intel)

# Linear performance ? (single socket)




Notice the linear improvement in number of queue = power of 2

# Queue/IRQ pins to CPU ?

```
# grep -R bus_bind_intr src/sys/dev/*
```

- bxe:  QLogic NetXtreme II Ethernet 10Gb PCIe
- cxgbe: Chelsio T4-, T5-, and T6-based (into #ifdef RSS)
- e1000 (igb, em, lem) : Intel Gigabit
- ixgbe: Intel 10 Gigabit
- ixl: Intel XL710 Ethernet 40Gb
- qlnxe: Cavium 25/40/100 Gigabit Ethernet
- sfxge: Solarflare 10Gb
- vxge: Neterion X3100 10Gb

# Queue/IRQ pins to CPU

- Config 1: Default
- Config 2: Queue/IRQ pining

```
chelsio_affinity_enable="YES"


~# service chelsio_affinity start
Bind t5nex0:0a IRQ 284 to CPU 0
Bind t5nex0:0a IRQ 285 to CPU 1
Bind t5nex0:0a IRQ 286 to CPU 2
Bind t5nex0:0a IRQ 287 to CPU 3
Bind t5nex0:0a IRQ 288 to CPU 4
Bind t5nex0:0a IRQ 289 to CPU 5
Bind t5nex0:0a IRQ 290 to CPU 6
Bind t5nex0:0a IRQ 291 to CPU 7
(...)
```

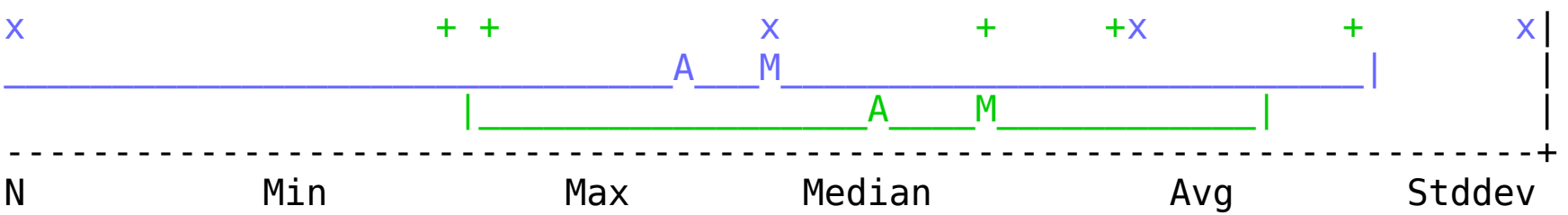
# Queue/IRQ pins to CPU

```
x Xeon E5-2650v2 & cxgbe, default: inet4 packets-per-second
+ Xeon E5-2650v2 & cxgbe, IRQ pinned to CPU: inet4 packets-per-second
```

```
    N           Min           Max        Median           Avg        Stddev
x   5      10939210      10969716      10952795      10951860     12056.937
+   5      11132364      11161395      11151483      11146670     12273.277
Difference at 95.0% confidence
        194810 +/- 17742.8
        1.77878% +/- 0.163429%
        (Student's t, pooled s = 12165.6)
```

Small benefit and only if pps >10Mpps

```
x Atom C2750 & cxgbe, default: inet4 packets-per-second
+ Atom C2750 & cxgbe, IRQ pinned to CPU: inet4 packets-per-second
```



```
    N           Min           Max        Median           Avg        Stddev
x   5       4059502       4232479       4149250       4139666     76051.798
+   5     4112849.5       4212811       4173030     4160909.7     43836.876
No difference proven at 95.0% confidence
```

# Increasing RX queues number

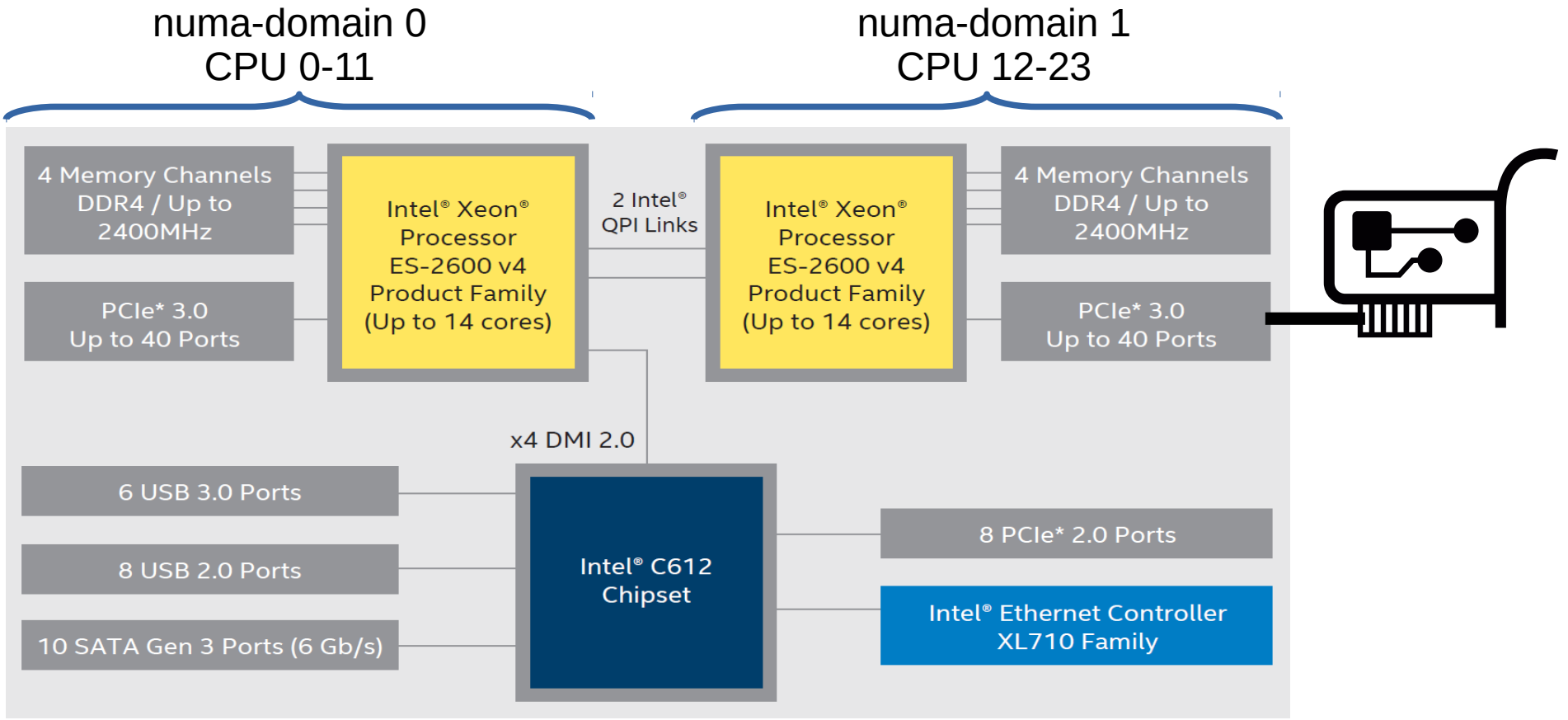
| Setup<br>E5-2650v4 (2x12 cores) | 8 queues<br>(default for ixgbe & cxgbe) | 24 queues<br>(default for mlx5en) | ministat |
|---|---|---|---|
| ixgbe<br>Intel 82599ES | 6.72 Mpps | 8.07 Mpps | 21.34% +/- 4.96% |
| cxgbe<br>Chelsio T520 | 9.59 Mpps | 12.40 Mpps | 29.45% +/- 0.37% |
| mlx5en<br>Mellanox ConnectX-4 Lx | *7.26 Mpps* | 14.64 Mpps | |

Tips 5: Check default maximum of queues and increase it if ncpu > 8

*mlx4en drivers didn't allow to changes number of queue (16 here)*

| | |
|---|---|
| 10Gb/s full duplex IMIX | 7 Mpps |
| 1Gb/s full duplex IMIX | 700 Kpps |

# NUMA affinity




Intel Xeon Processor E5-2600 v4 Product Family: Platform Brief

```
t5nex0: <Chelsio T520-CR> mem 0xc9200000-0xc927ffff,0xc8000000-
0xc8ffffff,0xc9684000-0xc9685fff irq 50 at device 0.4 numa-domain 1 on pci14
```

# Default: NO NUMA affinity

- Default CPU load with 12 RX queues:

```
last pid:  1080;  load averages:  7.13,  3.04,  1.30
273 processes: 35 running, 125 sleeping, 113 waiting
CPU 0:   0.0% user,  0.0% nice,  0.0% system,  0.4% interrupt, 99.6% idle
CPU 1:   0.0% user,  0.0% nice,  0.0% system,  0.4% interrupt, 99.6% idle
CPU 2:   0.0% user,  0.0% nice,  0.0% system,  0.0% interrupt,  100% idle
CPU 3:   0.0% user,  0.0% nice,  0.0% system,  0.0% interrupt,  100% idle
CPU 4:   0.0% user,  0.0% nice,  0.0% system, 89.8% interrupt, 10.2% idle
CPU 5:   0.0% user,  0.0% nice,  0.0% system,  100% interrupt,  0.0% idle
CPU 6:   0.0% user,  0.0% nice,  0.0% system, 94.9% interrupt,  5.1% idle
CPU 7:   0.0% user,  0.0% nice,  0.0% system, 89.8% interrupt, 10.2% idle
CPU 8:   0.0% user,  0.0% nice,  0.0% system, 84.6% interrupt, 15.4% idle
CPU 9:   0.0% user,  0.0% nice,  0.0% system, 92.1% interrupt,  7.9% idle
CPU 10:  0.0% user,  0.0% nice,  0.0% system, 84.6% interrupt, 15.4% idle
CPU 11:  0.0% user,  0.0% nice,  0.0% system, 83.9% interrupt, 16.1% idle
CPU 12:  0.0% user,  0.0% nice,  0.0% system, 85.8% interrupt, 14.2% idle
CPU 13:  0.0% user,  0.0% nice,  0.0% system, 92.1% interrupt,  7.9% idle
CPU 14:  0.0% user,  0.0% nice,  0.0% system, 85.0% interrupt, 15.0% idle
CPU 15:  0.0% user,  0.0% nice,  0.0% system, 78.0% interrupt, 22.0% idle
CPU 16:  0.0% user,  0.0% nice,  0.4% system,  0.0% interrupt, 99.6% idle
CPU 17:  0.0% user,  0.0% nice,  0.0% system,  0.0% interrupt,  100% idle
CPU 18:  0.0% user,  0.0% nice,  0.0% system,  0.0% interrupt,  100% idle
CPU 19:  0.0% user,  0.0% nice,  0.0% system,  0.0% interrupt,  100% idle
CPU 20:  0.0% user,  0.0% nice,  0.0% system,  0.0% interrupt,  100% idle
CPU 21:  0.0% user,  0.0% nice,  0.0% system,  0.0% interrupt,  100% idle
CPU 22:  0.0% user,  0.0% nice,  0.0% system,  0.0% interrupt,  100% idle
CPU 23:  0.0% user,  0.0% nice,  0.0% system,  0.0% interrupt,  100% idle
Mem: 13M Active, 13M Inact, 1170M Wired, 6393K Buf, 248G Free
```

Scheduler or drivers not NUMA aware

Numa-domain 0

Numa-domain 1

# NUMA affinity

- cxgbe configured with 12 RX queues, plugged on PCI-E belonging to numa-domain 1 (cores 12-23)

- Config 1: no-affinity (default)
- Config 2: cxgbe queues pined to core 0-11

```
chelsio_affinity_enable="YES"
```

- Config 3: cxgbe queues pined to core 12-23

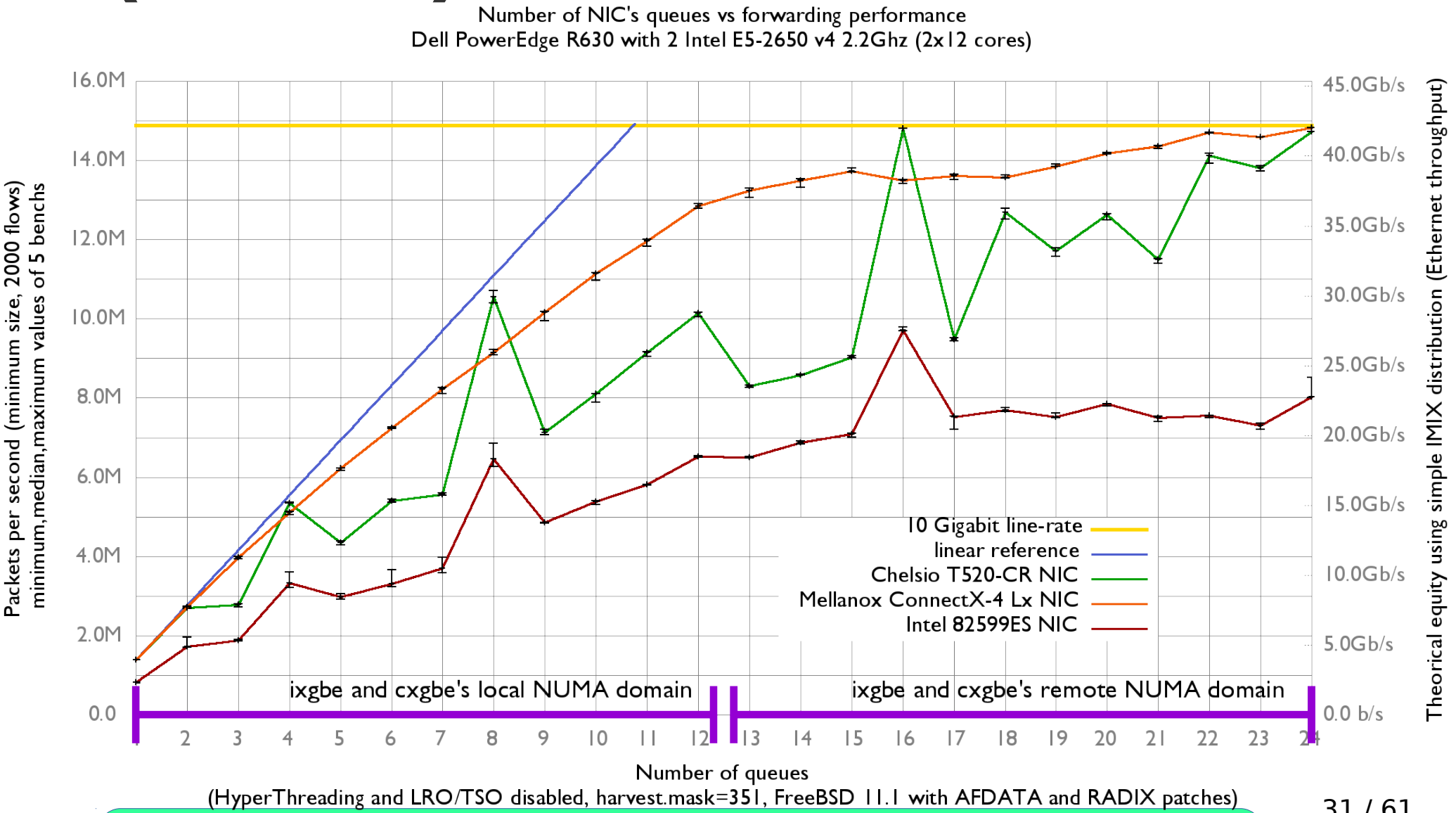
```
chelsio_affinity_enable="YES"

chelsio_affinity_firstcpu="12"
```

# NUMA affinity

```
x Xeon 2xE5-2650v4 & cxgbe, default: inet4 packet-per-seconds
+ Xeon 2xE5-2650v4 & cxgbe, affinity-numa0: inet4 packet-per-seconds
* Xeon 2xE5-2650v4 & cxgbe, affinity-numa1: inet4 packet-per-seconds
+-----------------------------------------------------------------------+
|                  +x                                                 *  |
|+      x     x + +x+                                                ** **|
|         |____A__M_|                                                    |
|      |_______A__M____|                                                |
|                                                                  |MA_||
+-----------------------------------------------------------------------+
    N           Min           Max        Median           Avg        Stddev
x   5       9351036       9580847       9571249       9510859     98839.328
+   5       9220385       9603697       9557225     9493098.6      154964.3
No difference proven at 95.0% confidence
*   5      10584085      10670945      10617361      10629374     35170.165
Difference at 95.0% confidence
        1.11851e+06 +/- 108191
        11.7604% +/- 1.25701%
        (Student's t, pooled s = 74182.7)
```

Tips 6: Take care of NUMA affinity with queue to CPU pining

# Linear performance ? (NUMA)




Notice that mlx5en didn't required number of queue = power of 2
cxgbe reaches line-rate with only 16 queues

# NIC hardware acceleration features

- Checksum offload: rxcsum, txcsum, ...
- VLAN offload: vlanmtu, vlanhwtag, vlanhwfilter, vlanhwcsum,...
- TSO :TCP Segmentation Offload
  - NIC split large segment into MTU-sized packets
  - MUST be disabled on a router (and incompatible with ipfw nat)
- LRO: Large Received Offload
  - Breaks the end-to-end principle on a router: MUST be disabled
- Hardware resources reservation

# Disabling LRO & TSO

| Server<br>CPU (cores) & NIC | Enabled<br>(default) | Disabled | ministat |
|---|---|---|---|
| E5-2650v4 (2x12) & ixgbe<br>Xeon & Intel 82599ES | 7.97 Mpps | 8.07 Mpps | No difference proven at 95.0% confidence |
| E5-2650v4 (2x12) & cxgbe<br>Xeon & Chelsio T520 | 12.40 Mpps | 12.40 Mpps | No difference proven at 95.0% confidence |
| E5-2650v4 (2x12) & ml4en<br>Xeon & Mellanox ConnectX-3 Pro | 8.05 Mpps | 7.85 Mpps | No difference proven at 95.0% confidence |
| E5-2650v4 (2x12) & ml5en<br>Xeon & Mellanox ConnectX-4 Lx | 14.65Mpps | 14.83 Mpps | 1.3% +/- 0.1% |
| E5-2650v2 (8) & cxgbe<br>Xeon & Chelsio T540 | 10.84 Mpps | 10.92 Mpps | 0.74% +/- 0.26% |
| C2758 (8) & cxgbe<br>Atom & Chelsio T540 | 4.20 Mpps | 4.18 Mpps | No diff. proven at 95.0% confidence |
| C2758 (8) & ixgbe<br>Atom & Intel 82599ES | 3.06 Mpps | 3.06 Mpps | No diff. proven at 95.0% confidence |
| C2558 (4) & igb<br>Atom & Intel I354 | 1.2 Mpps | 1.2 Mpps | No diff. proven at 95.0% confidence |
| GX412 (4) & igb<br>AMD & intel I210 | 729 Kpps | 727 Kpps | No diff. proven at 95.0% confidence |

Tips 6: You can disable LRO & TSO on your router/firewall

# hw.igb|ix.rx_process_limit

| Server<br>CPU (cores) & NIC | 100(igb), 256(ix), default median | -1 (disabled) median | ministat |
|---|---|---|---|
| E5-2650v4 (2x12) & ixgbe<br>Xeon & Intel 82599ES | 8.04 Mpps | 8.34 Mpps | 3.75% +/- 0.73% |
| C2758 (8) & ixgbe<br>Atom & Intel 82599ES | 3.12 Mpps | 3.85 Mpps | 22.66% +/- 2.14% |
| C2558 (4) & igb<br>Atom & Intel I354 | 1.10 Mpps | 1.13 Mpps | 1.65% +/- 0.9% |
| GX412 (4) & igb<br>AMD & Intel I210 | 730 Kpps | 735 Kpps | No diff. proven at 95.0% conf. |

Tips 6: Disable rx_process_limit with igb & ixgbe

# Disabling unused features

“Disallowing capabilities provides a hint to the driver and firmware to not reserve hardware resources for that feature”

/boot/loader.conf:

```
hw.cxgbe.toecaps_allowed="0"
hw.cxgbe.rdmacaps_allowed="0"
hw.cxgbe.iscsicaps_allowed="0"
hw.cxgbe.fcoecaps_allowed="0"
```

# Disabling unused features

```
x Xeon 2xE5-2650v4 & cxgbe, default caps enabled: inet4 packet-per-seconds
+ Xeon 2xE5-2650v4 & cxgbe, caps disabled: inet4 packet-per-seconds
+-------------------------------------------------------------------------+
|x                                                                       +|
|x                                                                       +|
|x                                                                       +|
|x                                                                       +|
|x                                                                       +|
|A                                                                        |
|                                                                       A|
+-------------------------------------------------------------------------+
    N           Min           Max        Median           Avg        Stddev
x   5      12411366      12413439      12411915      12412289     901.22767
+   5      14796094      14800927      14799082      14798629     2169.6179
Difference at 95.0% confidence
        2.38634e+06 +/- 2422.83
        19.2256% +/- 0.0201158%
        (Student's t, pooled s = 1661.24)
```

Tips 7: Disable unused caps with cxgbe

# Forwarding tuning summary

- **Yandex‘s patches: AFDATA and RADIX locks**
- **Increase Intel & Chelsio NIC queues if ncpu > 8, but kept power-of-two number**
- boot/loader.conf

```
machdep.hyperthreading_allowed="0"
hw.igb.rx_process_limit="-1"
hw.em.rx_process_limit="-1"
hw.ix.rx_process_limit="-1"
hw.cxgbe.toecaps_allowed="0"
hw.cxgbe.rdmacaps_allowed="0"
hw.cxgbe.iscsicaps_allowed="0"
hw.cxgbe.fcoecaps_allowed="0"
```

Intel drivers

Chelsio drivers (useful starting at 10Mpps, so with Yandex’s patches)

- etc/rc.conf

```
harvest_mask="351"
```

# Before vs after tuning (IPv4)

| Setup<br>CPU (cores) & NIC | Generic 11.1 | Yandex patched<br>& tuned 11.1 | ministat |
|---|---|---|---|
| E5-2650v4 (2x12) & ixgbe<br>Xeon & Intel 82599ES | 3.74 Mpps | 8.61 Mpps | 127.93% +/- 8.44% |
| E5-2650v4 (2x12) & cxgbe<br>Xeon & Chelsio T520 | 4.83 Mpps | 14.8 Mpps | 204.3% +/- 4.80% |
| E5-2650v4 (2x12) & ml4en<br>Xeon & Mellanox ConnectX-3 Pro | 3.92 Mpps | 8.06 Mpps | 126.9% +/- 7.77% |
| E5-2650v4 (2x12) & ml5en<br>Xeon & Mellanox ConnectX-4 Lx | 0 Mpps | 14.64 Mpps | *NA* |
| E5-2650v2 (8) & cxgbe<br>Xeon & Chelsio T540 | 5.75 Mpps | 11.15 Mpps | 139.8% +/- 5.0% |
| E5-2650v2 (8) & oce<br>Xeon & Emulex be3 | 1.33 Mpps | 1.33 Mpps | No diff. proven at 95.0% confidence |
| C2758 (8) & cxgbe<br>Atom & Chelsio T540 | 2.83 Mpps | 4.19 Mpps | 50.49% +/- 5.33% |
| C2758 (8) & ixgbe<br>Atom & Intel 82599ES | 2.29 Mpps | 3.85 Mpps | 66.97% +/- 2.7% |
| C2558 (4) & igb<br>Atom & Intel I354 | 951 Kpps | 1.13 Mpps | 18.58% +/- 1.17% |
| GX412 (4) & igb<br>AMD & Intel I210 | 726 Kpps | 735 Kpps | 1.03% +/- 0.56% |

# IPv4 vs IPv6 performance

| Setup<br>CPU (cores) & NIC | inet4 | inet6 | ministat |
|---|---|---|---|
| E5-2650v4 (2x12) & ixgbe<br>Xeon & Intel 82599ES | 8.35 Mpps | 8.12 Mpps | -3.25% +/- 1.7% |
| E5-2650v4 (2x12) & cxgbe<br>Xeon & Chelsio T520 | 14.8 Mpps | 14.47 Mpps | -2.18% +/- 0.02% |
| E5-2650v4 (2x12) & ml4en<br>Xeon & Mellanox ConnectX-3 Pro | 8.06 Mpps | 7.71 Mpps | -3.35% +/- 3.26% |
| E5-2650v4 (2x12) & ml5en<br>Xeon & Mellanox ConnectX-4 Lx | 14.84 Mpps | 14.29 Mpps | -3.70% +/- 0.02% |
| E5-2650v2 (8) & cxgbe<br>Xeon & Chelsio T540 | 10.94 Mpps | 9.18 Mpps | -16.12% +/- 0.19% |
| C2758 (8) & cxgbe<br>Atom & Chelsio T540 | 4.29 Mpps | 3.43 Mpps | -19.08% +/- 1.61% |
| C2758 (8) & ixgbe<br>Atom & Intel 82599ES | 3.81 Mpps | 3.43 Mpps | -9.84% +/- 1.3% |
| C2558 (4) & igb<br>Atom & Intel I354 | 1.23 Mpps | 1.08 Mpps | -11.79% +/- 0.5% |
| GX412 (4) & igb<br>AMD & Intel I210 | 734 Kpps | 709 Kpps | -3.6% +/- 0.70% |

Notice the difference between Chelsio and Intel NIC on C2758
(bottleneck no more in the drivers but in the Kernel)

# Configuration impact

- VLAN tagging
- VIMAGE & VNET jail
- Bridge

# VLAN tagging

- Config 1: No VLAN

```
ifconfig_cxl0="inet 198.18.0.10/24"
ifconfig_cxl1="inet 198.19.0.10/24"
```

- Config 2: VLAN tagging

```
vlans_cxl0="2"
ifconfig_cxl0="up"
ifconfig_cxl0_2="inet 198.18.0.10/24"
vlans_cxl1="4"
ifconfig_cxl1="up"
ifconfig_cxl1_4="inet 198.19.0.10/24"
```

# VLAN tagging

```
x Xeon E5-2650v2 & cxgbe, no VLAN tagging: inet4 packets-per-second
+ Xeon E5-2650v2 & cxgbe, VLAN tagging: inet4 packets-per-second
+-------------------------------------------------------------------------+
|+                                                                        |
|+                                                                      xx|
|+++                                                                   xxx|
|                                                                     |A||
|MA|                                                                      |
+-------------------------------------------------------------------------+
    N           Min           Max        Median           Avg        Stddev
x   5      10917371      10970686      10945136      10946743     22298.313
+   5       9056449       9104195       9064032     9075563.7     21531.387
Difference at 95.0% confidence
        -1.87118e+06 +/- 31966.4
        -17.0935% +/- 0.267353%
        (Student's t, pooled s = 21918.2)
```

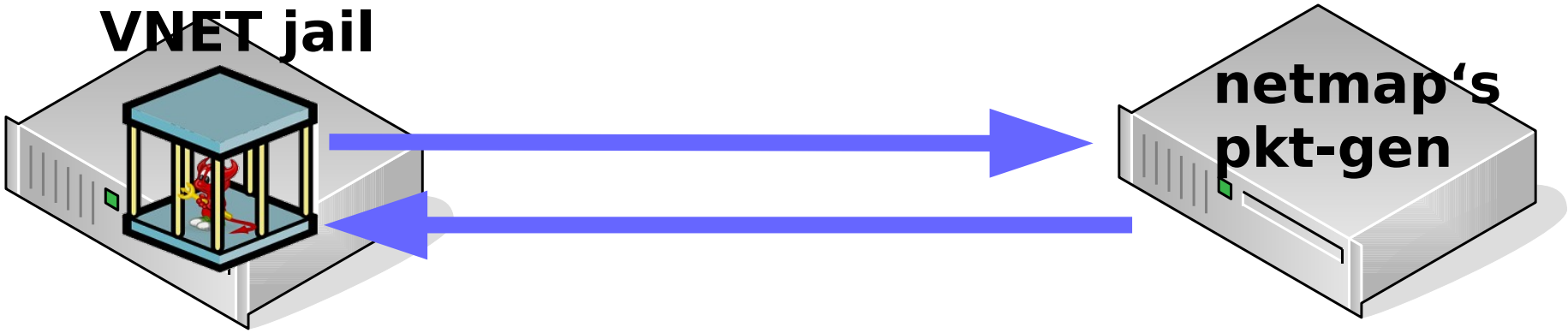
-17% with tagging: Known problem
Yet another patch from Yandex
ixgbe: https://reviews.freebsd.org/D12040
mlx5en: https://reviews.freebsd.org/D12041

# Adding VIMAGE support

```
options                 VIMAGE
```

| E5-2650v2 & cxgbe<br>Xeon & Chelsio T540 | GENERIC (median) Mpps | VIMAGE (median) Mpps | ministat |
|---|---|---|---|
| inet 4 forwarding | 10.9 | 10.2 | -6.25% +/- 0.29% |
| inet 6 forwarding | 9.18 | 9.39 | 2.24% +/- 0.33 |

# Multi-tenant router




```
                host /etc/rc.conf
ifconfig_cxl0="up -tso4 -tso6 -lro -vlanhwtso"
ifconfig_cxl1="up -tso4 -tso6 -lro -vlanhwtso"
jail_enable="YES"
jail_list="jrouter"
```

```
             Jail jrouter /etc/rc.conf
gateway_enable=YES
ipv6_gateway_enable=YES
ifconfig_cxl0="inet 198.18.0.10/24"
ifconfig_cxl1="inet 198.19.0.10/24"
static_routes="generator receiver"
route_generator="-net 198.18.0.0/16 198.18.0.108"
route_receiver="-net 198.19.0.0/16 198.19.0.108"
```

# VNET jail: impact on PPS

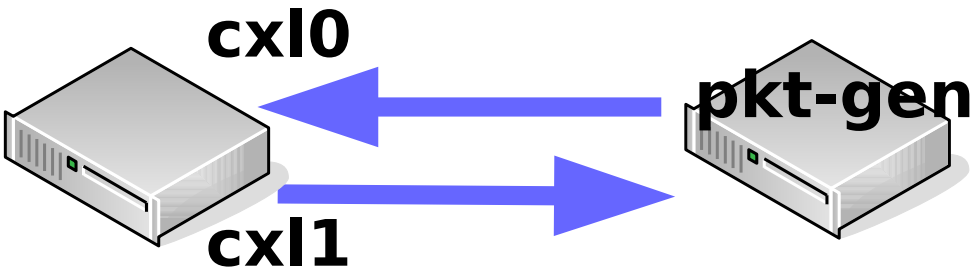
| E5-2650v2 & cxgbe<br>Xeon & Chelsio T540 | No Jail | VNET-Jail | Ministat |
|---|---|---|---|
| inet 4 forwarding | 10.8 Mpps | 11.0 Mpps | No diff. proven at 95.0% confidence |
| inet 6 forwarding | 10.0 Mpps | 10.0 Mpps | No diff. proven at 95.0% confidence |

VNET-jail rocks!

# if_bridge

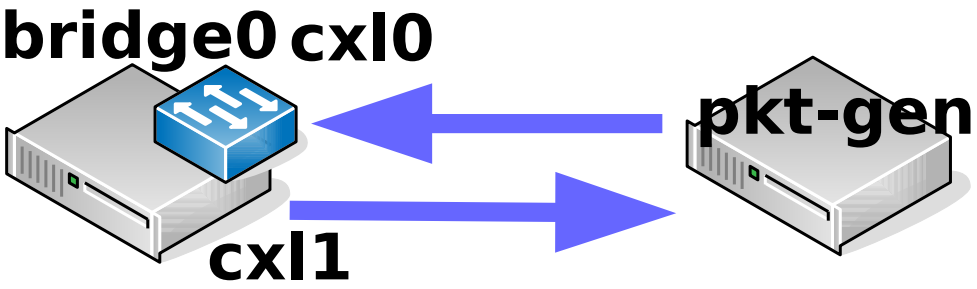
- Config 1: No bridge

```
ifconfig_cxl0="inet 198.18.0.10/24"
ifconfig_cxl1="inet 198.19.0.10/24"
```




- Config 2: Dummy bridge

```
cloned_interfaces="bridge0"
ifconfig_bridge0="inet 198.18.0.8/24 addm cxl0 up"
ifconfig_cxl0="up"
ifconfig_cxl1="inet 198.19.0.10/24"
```



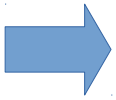
# if_bridge

```
x Xeon E5-2650v2 & cxgbe, NO bridge: inet4 packets-per-second
+ Xeon E5-2650v2 & cxgbe, bridge: inet4 packets-per-second
+------------------------------------------------------------------------+
|  +                                                                   x|
|++++                                                                 xx|
|                                                                     |A||
|  |AM|                                                                 |
+------------------------------------------------------------------------+
    N           Min           Max        Median           Avg        Stddev
x   5      11102006      11179490      11155098      11149783     28766.212
+   5       4040161       4322481     4201494.5     4178806.5     113801.03
Difference at 95.0% confidence
        -6.97098e+06 +/- 121051
        -62.5212% +/- 1.05729%
        (Student's t, pooled s = 83000.5)
```

-62% with bridge interface involved
bridge_input() include lot's of LOCK_

# Firewalls: Disclaimer!

None of the following benches can conclude a firewall is better than another.

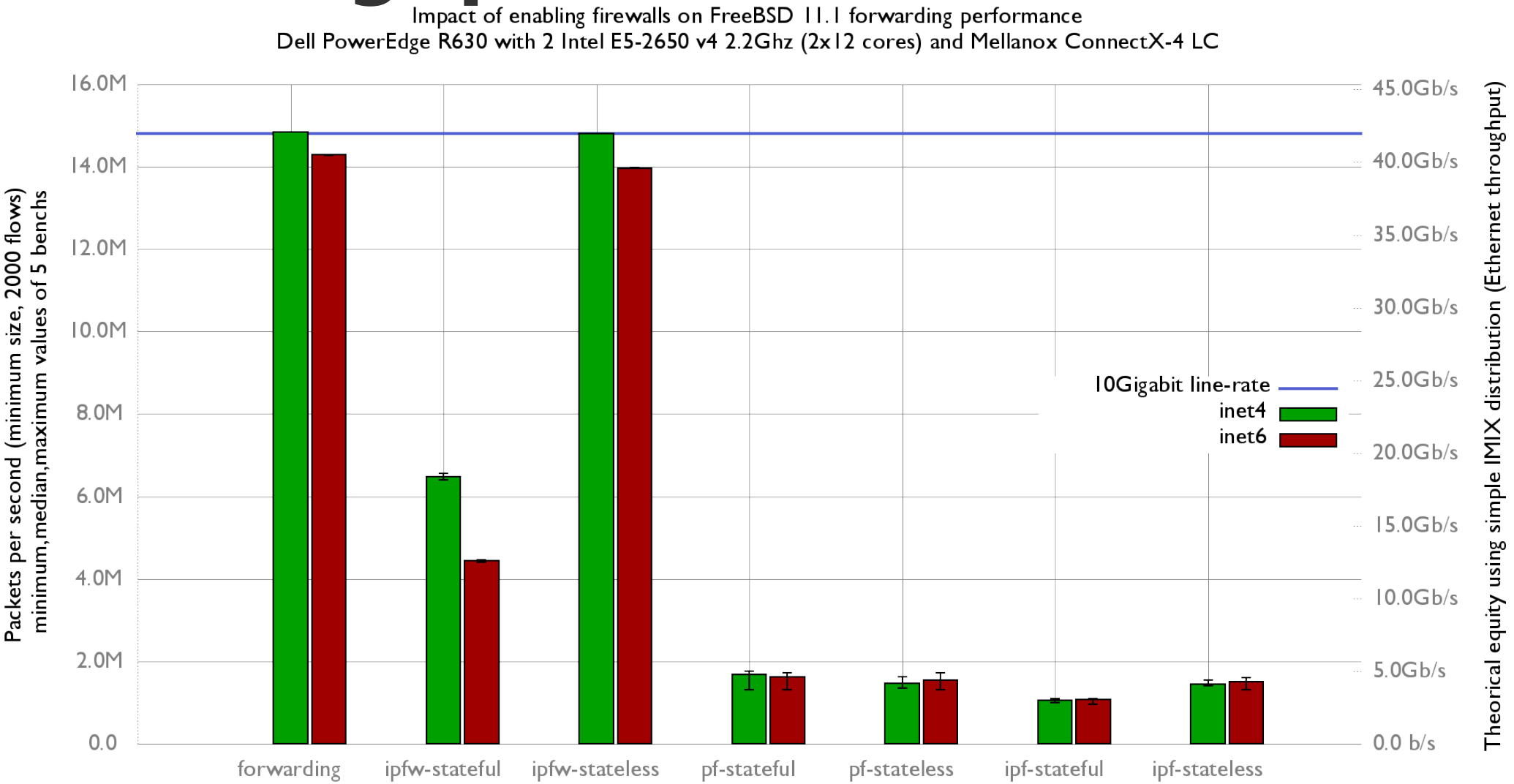


A firewall can't be reduced to its only forwarding performance impact

# Firewalls

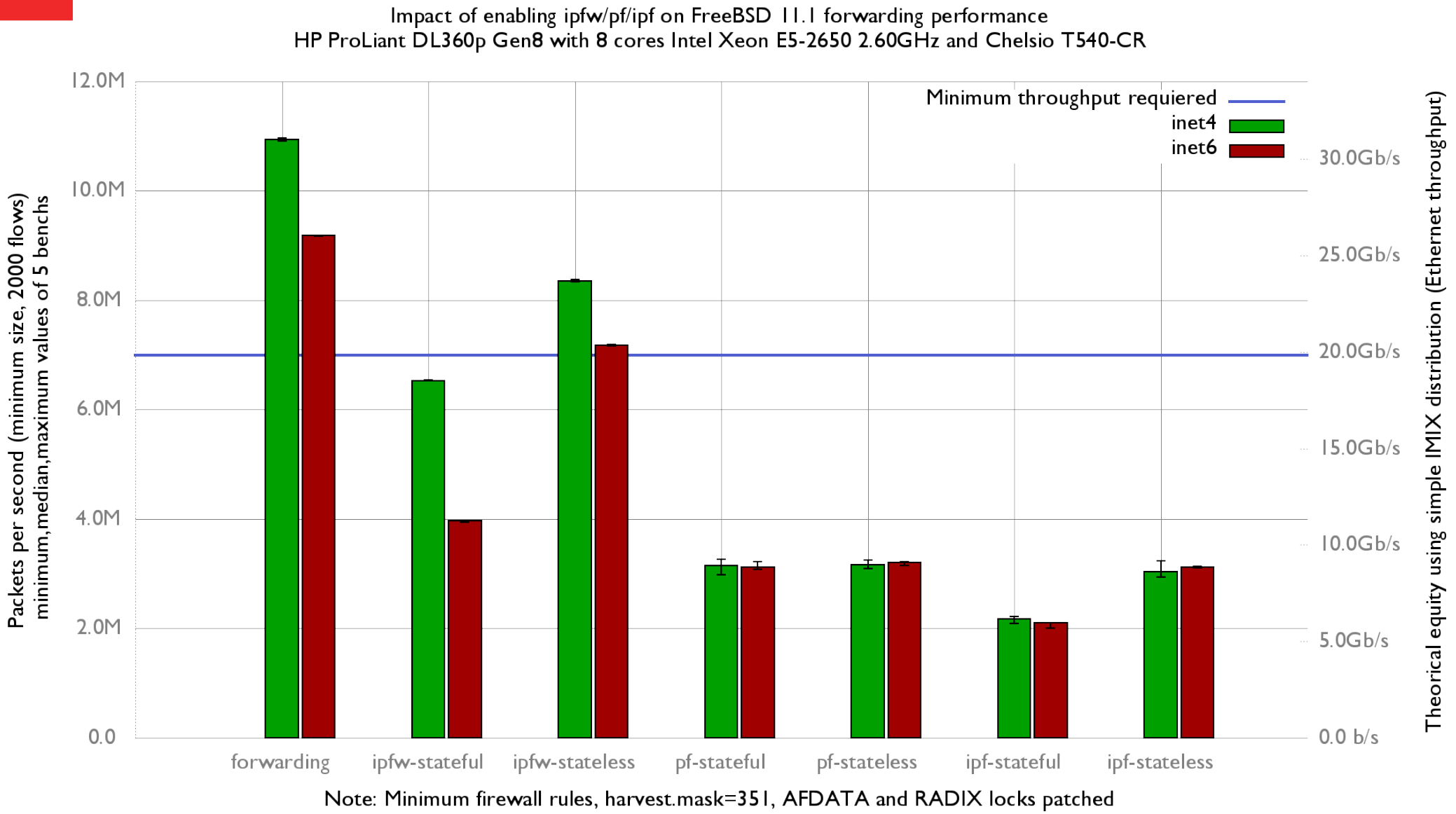
- How these impact throughput (PPS):
  - Enabling ipfw / pf / ipf with inet4 & inet6
  - Number of rules
  - Table size
  - Number of UDP flows

# Firewalls impact on throughput




Warning: do not conclude a firewall is better than another with this result!

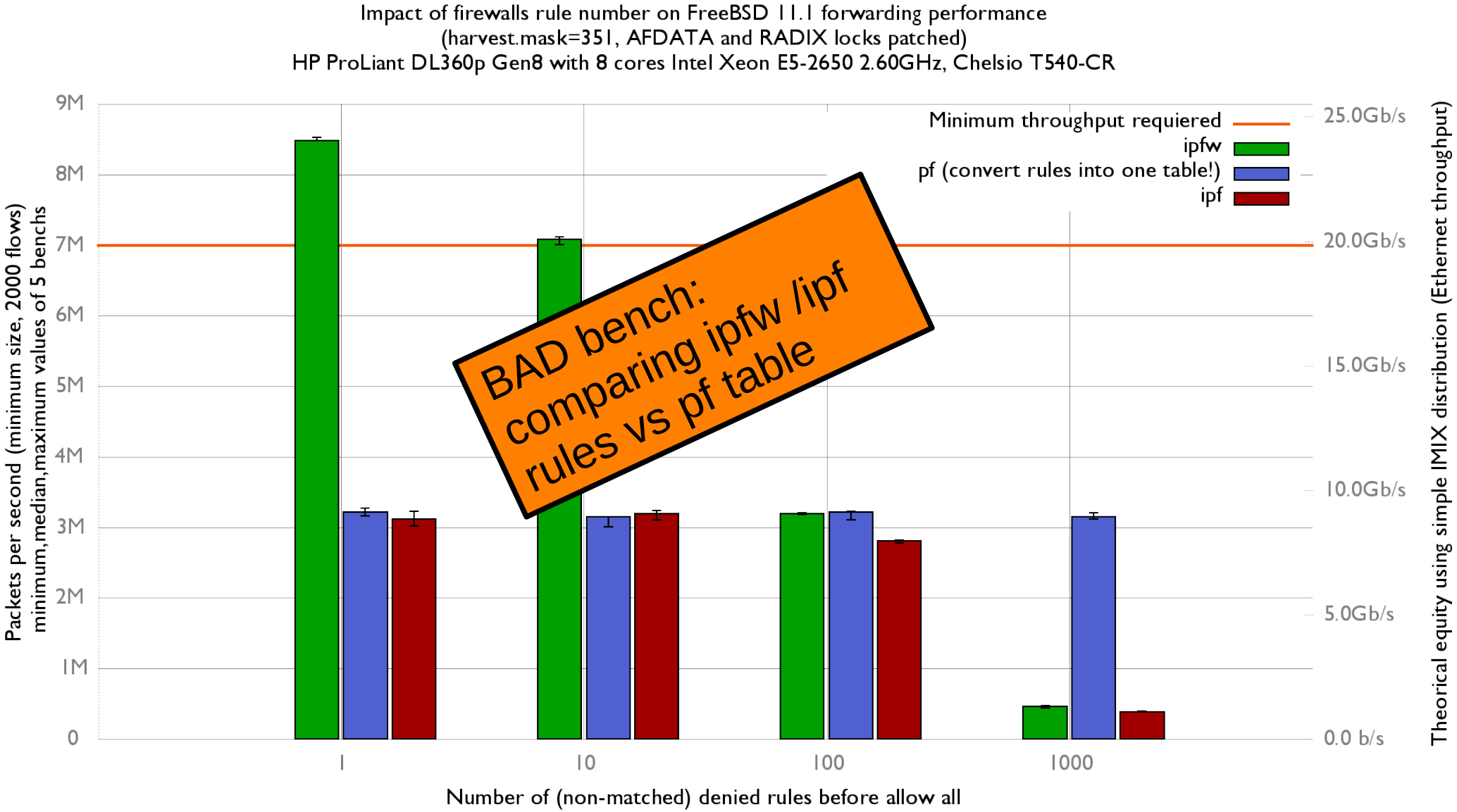
# Firewalls impact on throughput





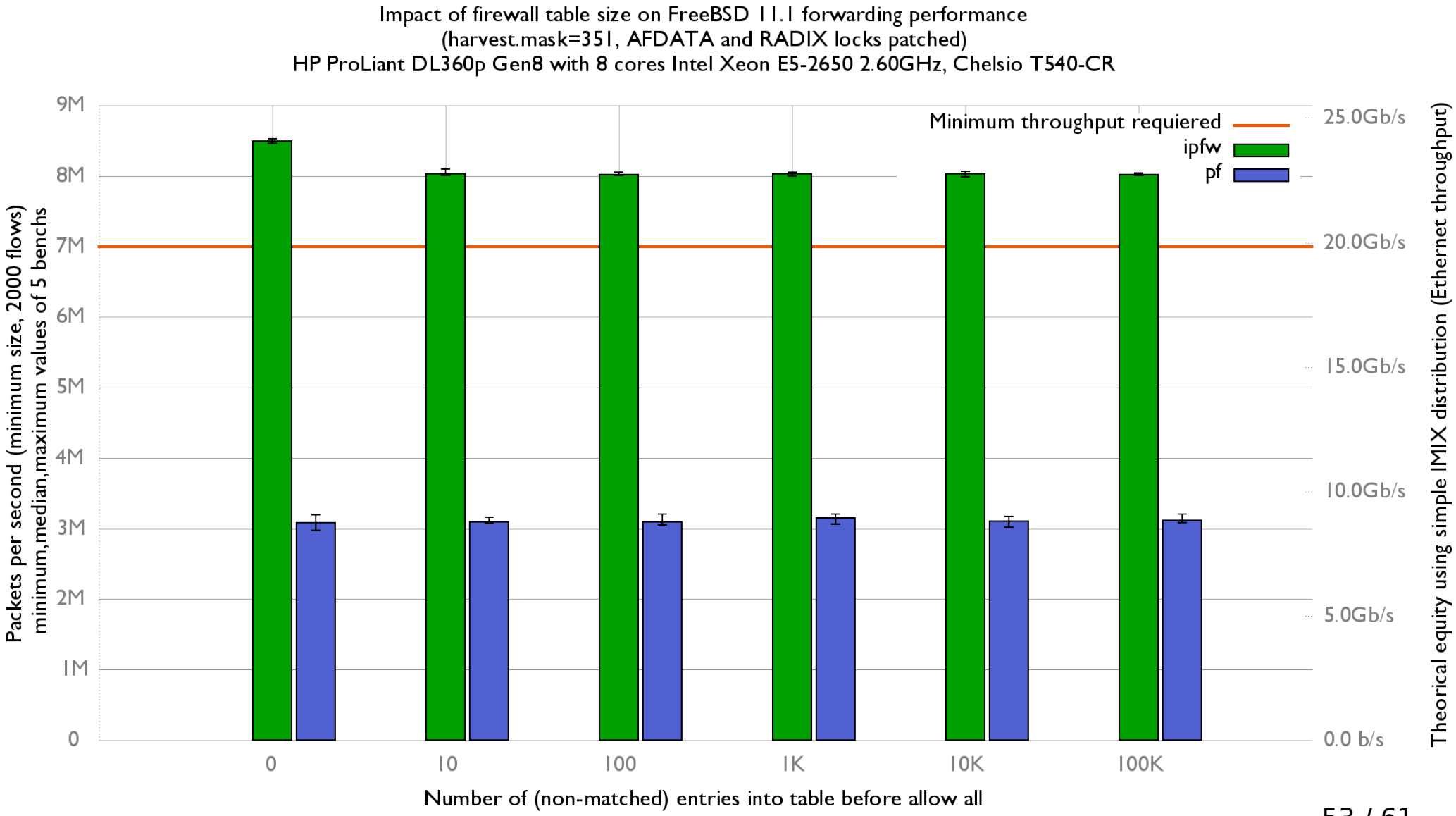

Warning: do not conclude a firewall is better than another with this result!

# Stateless: rules impact




Keep MINIMUM numbers of rules with ipfw/ipf

# Stateless: Table size impact




Use table

# Stateful ipfw: number of states

- One UDP flow create 1 state (dynamic rule)

```
check-state

ipfw add allow ip from any to any keep-state
```

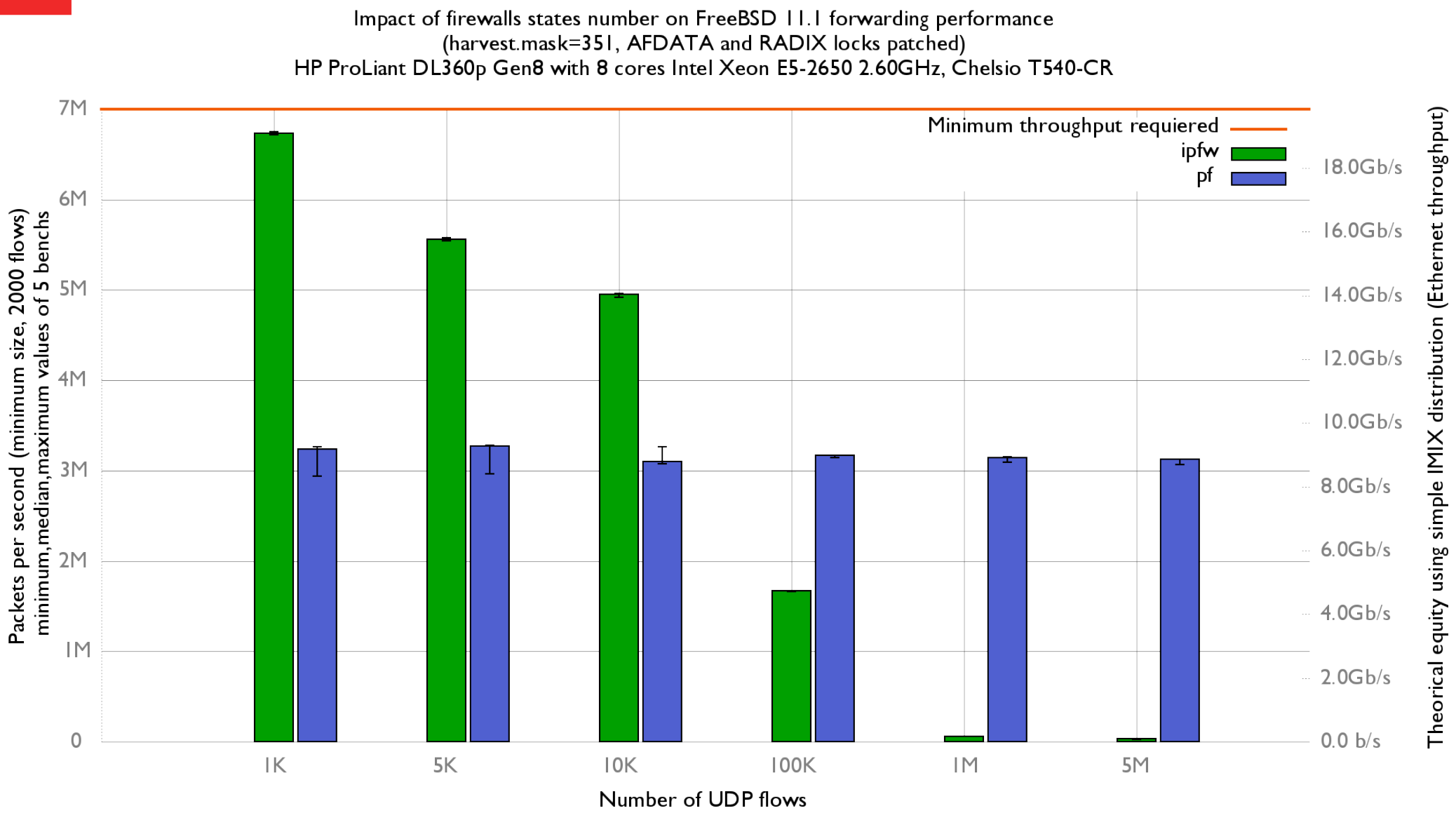
| keys | Default value | Increased value |
|---|---|---|
| dynamic rules<br>`net.inet.ip.fw.dyn_max` | 16 384 | 5 000 000 |
| hash table size<br>[max_dyn / 64 ?]<br>(power of 2)<br>`net.inet.ip.fw.dyn_buckets` | 256 | 65 536 (max) |

# Stateful pf: number of state

- One UDP flow consumes 2 pf states
- Linear relationship between maximum number of states and hash table size

| keys | Default value | Increased value |
|---|---|---|
| states limit<br>`set limit { states X }` | 10 000 | 10 000 000 |
| Hash table size<br>= state x 3<br>(power of 2)<br>`net.pf.pf_states_hashsize` | 32 768 | 33 554 432<br>*(max with 8GB RAM)* |
| RAM consummed<br>(hashsize x 80)<br>`vmstat -m \| grep pf_hash` | 2.5Mb | 2.5Gb |

# stateful: Number of state

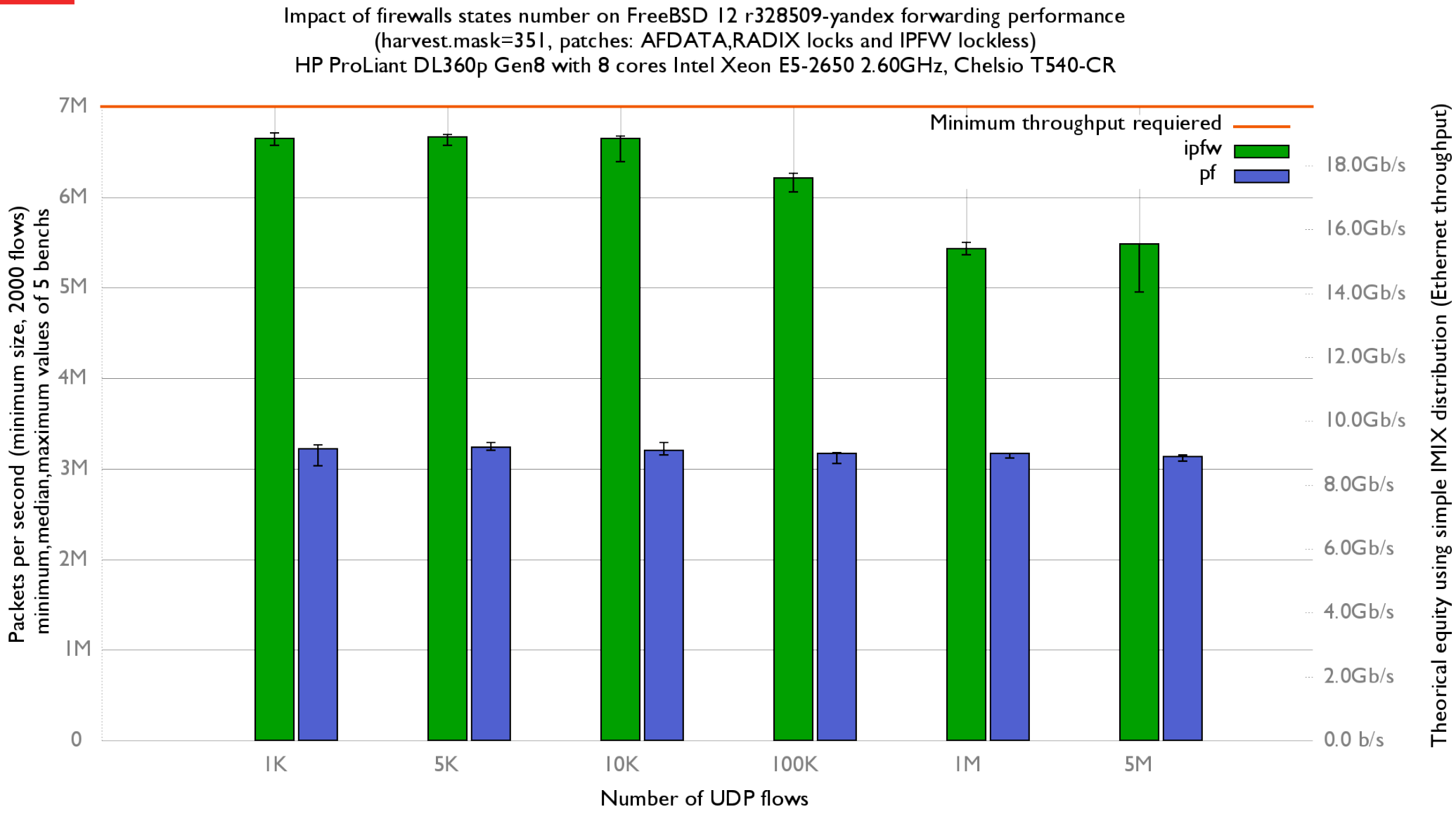



Note: For a stateful firewall with more than 100K… use pf on FreeBSD 11.1

# ipfw stateful lockless

- Andrey V. Elsukov (ae)’s reaction to the previous bench:
  - “Rework ipfw dynamic states implementation to be lockless on fast path”
  - Brings lot’s of performance improvement
  - Use ConcurrencyKit
  - Committed on head as r328988

# ipfw stateful lockless




For a fast stateful firewall… try IPFW on -head

# Resources

- Benches scripts, configurations, RAW results, flamegraph

https://github.com/ocochard/netbenches

- BSD Router Project (nanoBSD based on FreeBSD)

https://bsdrp.net

# Questions ?

# Thanks !