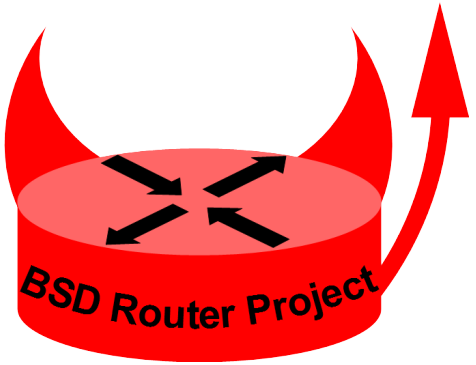
EuroBSDcon 2017

Tuning FreeBSD for routing and firewalling

Olivier Cochard-Labbé

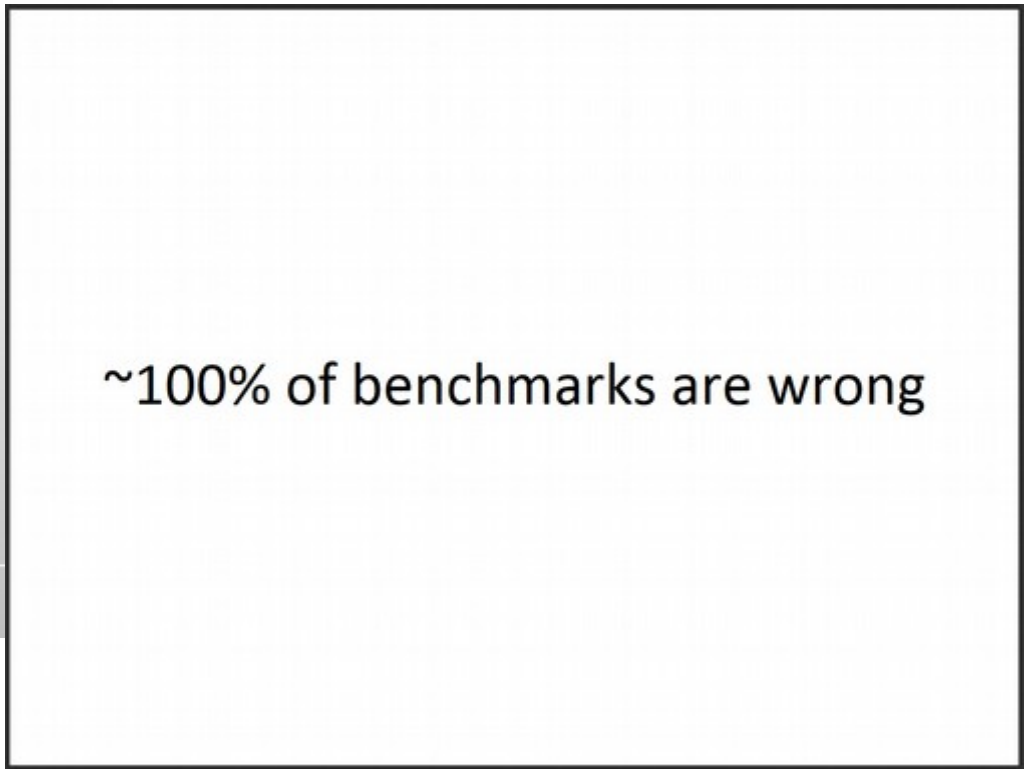
whoami(1)

- olivier.cochard@ 

- olivier@ 


FreeBSD®

Disclaimer



http://www.brendangregg.com/Slides/MeetBSD2014_Performance.pdf

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!

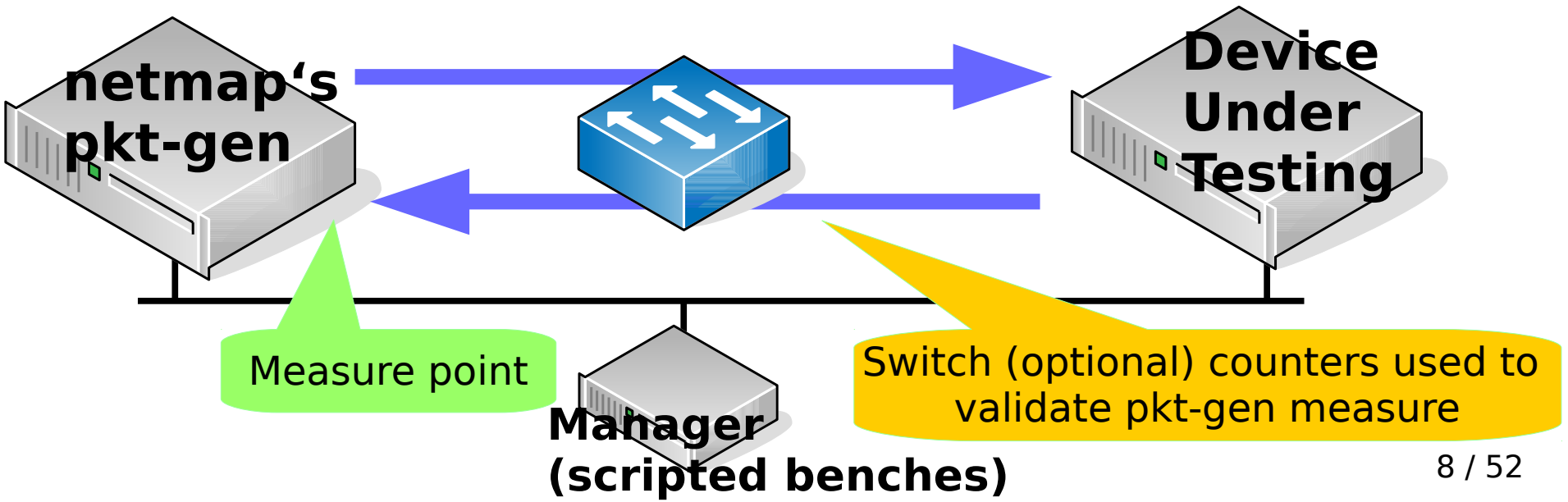
- Packets-per-second * Packets-size
- Estimated using Simple Internet Mix (IMIX) packet size trimodal reference distribution
- IPv4 layer:
  - PPS*(7*40 + 4*576 + 1500)/12*8
- Ethernet layer (switch counters):
  - PPS*(7*(40+14)+4*(576+14)+(1500+14))/12*8
- *Since about 2004, Internet packets size distribution 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



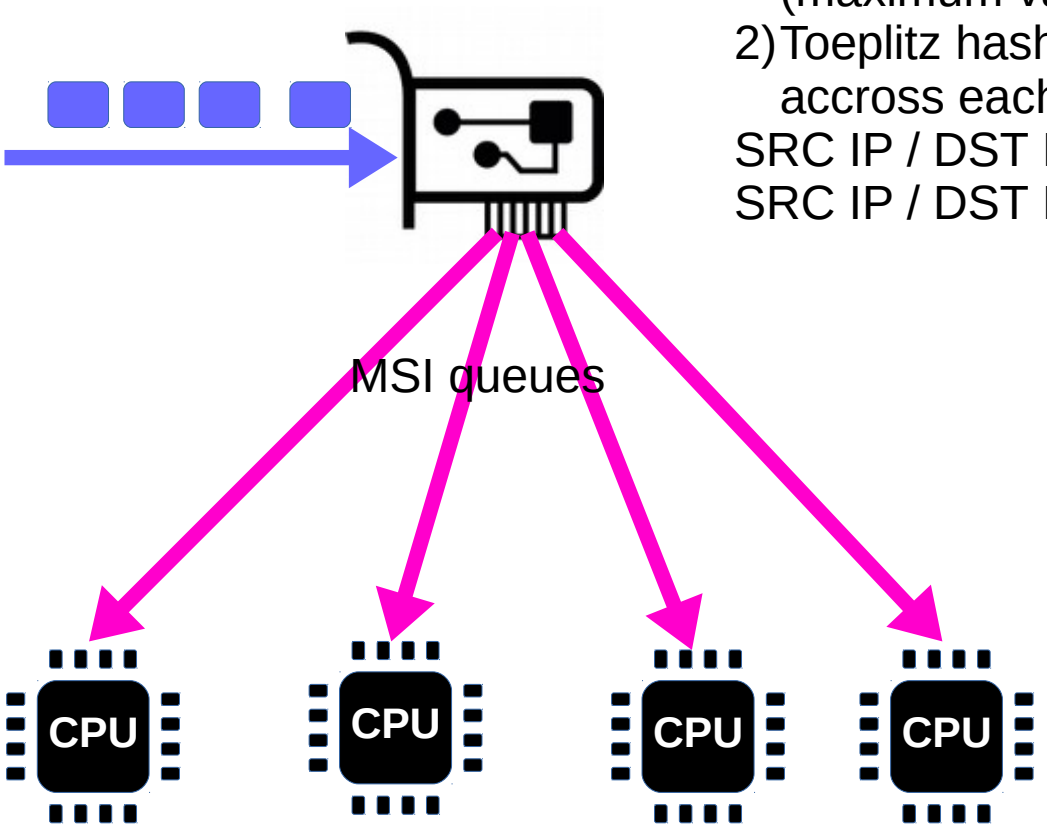
Hardware details

Servers	CPU	cores	GHz	Network card (driver name)
HP ProLiant DL360p Gen8	Intel **E5-2650** v2	8x2	2.6	10G Chelsio T540-CR (**cxl**) 10G Emulex OneConnect be3 (**oce**)
SuperMicro 5018A-FTN4	Intel Atom **C2758**	8	2.4	10G Chelsio T540-CR (**cxl**)
SuperMicro 5018A-FTN4	Intel Atom **C2758**	8	2.4	10G Intel 82599 (**ix**)
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 Mellanox NIC
- No 16 cores-in-one-socket CPUs
- No multi-socket

Same optical SFP model on all 10G: QFX-SFP-10GE-SR

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: cxl, ix, igb, bxe 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

HyperThreading & cxgbe(4)

```
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

HyperThreading & cxgbe(4)

- 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`

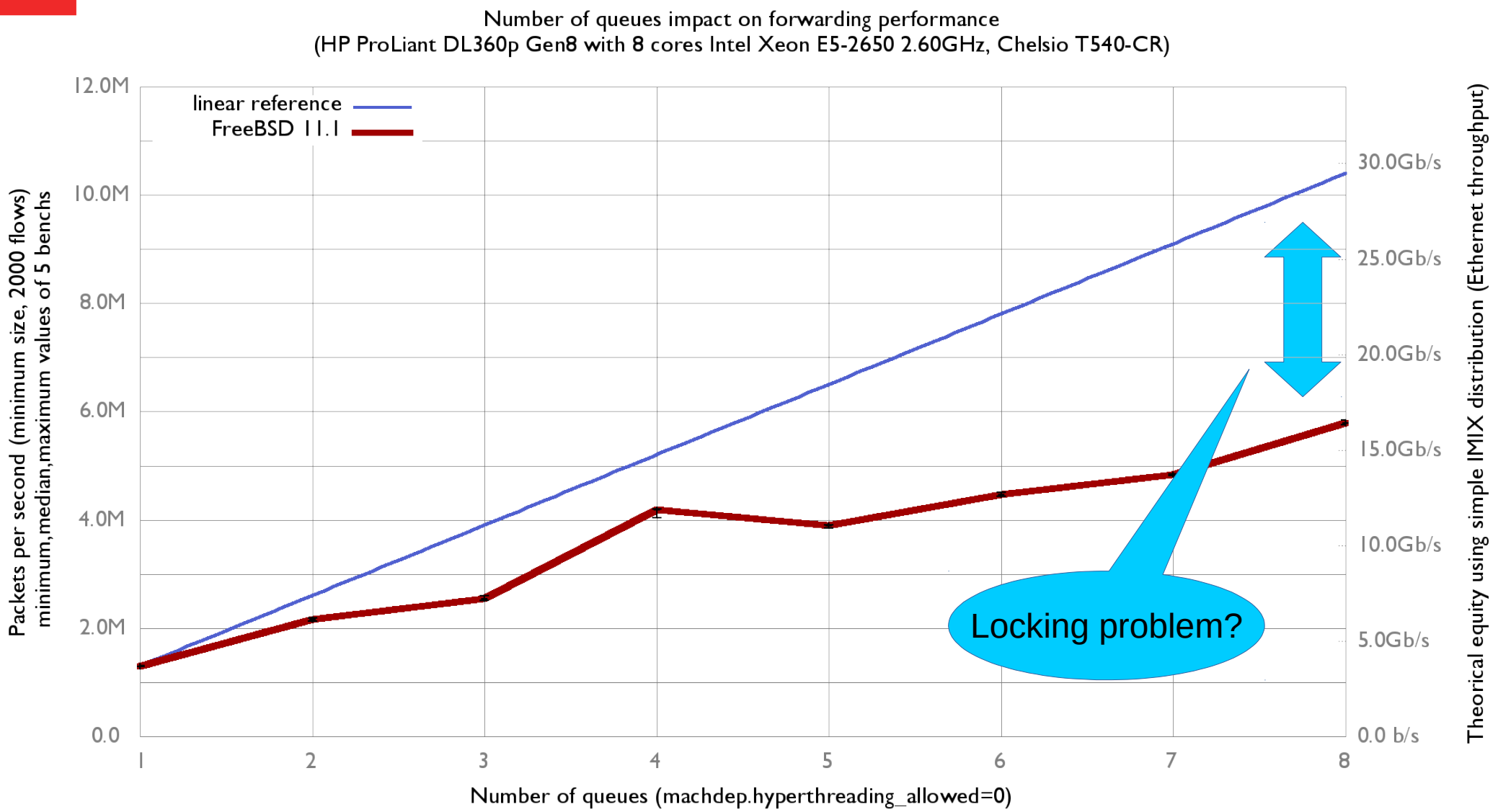
Disabling Hyper-Threading

ministat(1) is my friend

```
x Xeon E5-2650-cxgbe, HT-enabled & 8rxq(default): inet4 packets-per-second
+ Xeon E5-2650-cxgbe, HT-enabled & 16rxq: inet4 packets-per-second
* Xeon E5-2650-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)
```

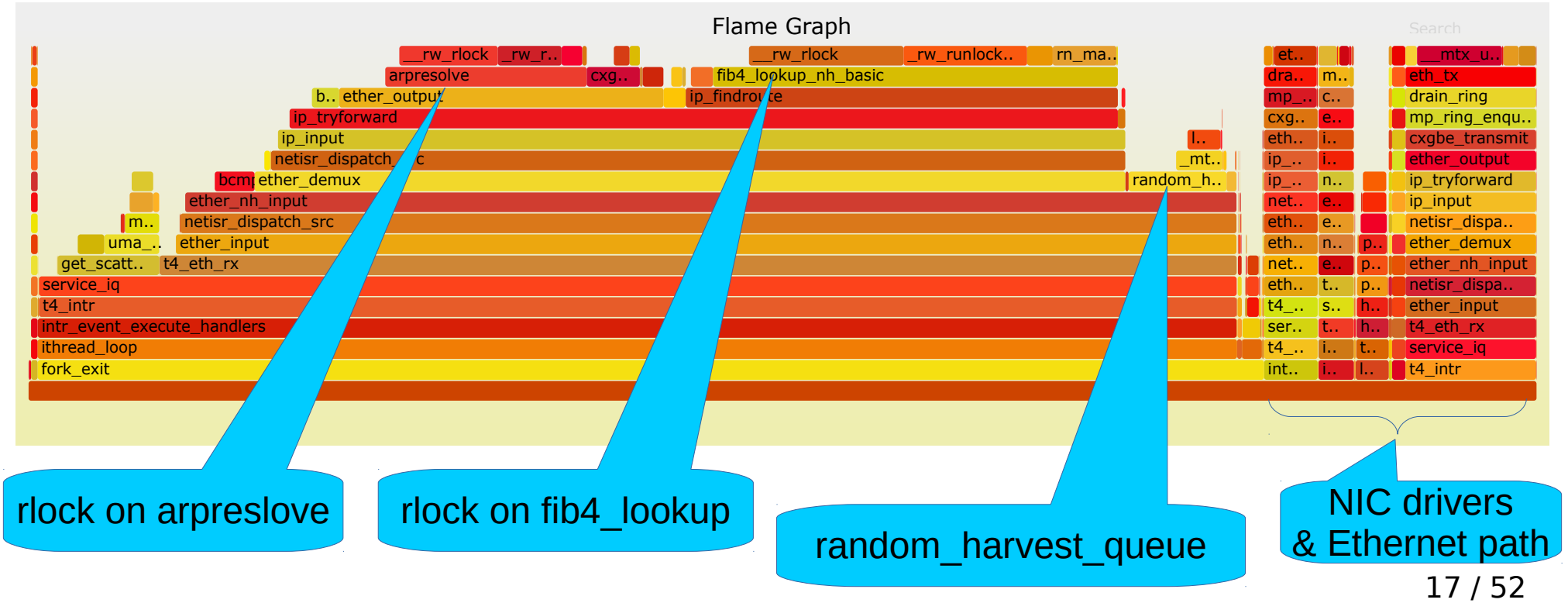
Tips 1: Disable Hyper-threading

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	511 (default) median	351 median	ministat
E5_2650-cxl Xeon & Chelsio NIC	5.76 Mpps	5.79 Mpps	No diff. proven at 95.0% confidence
E5_2650-oce Xeon & Emulex NIC	1.33 Mpps	1.33 Mpps	No diff. proven at 95.0% confidence
C2758-cxl Atom & Chelsio NIC	2.83 Mpps	3.17 Mpps	12.52% +/- 1.82%
C2758-ix Atom & Intel NIC	2.3 Mpps	2.43 Mpps	6.14% +/- 1.84%
C2558-igb Atom & Intel NIC	951 Kpps	1 Mpps	4.75% +/- 1.08%
GX412-igb AMD & Intel NIC	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"
(but ask to your security officer first)

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_2650-cxl Xeon & Chelsio NIC	5.75 Mpps	10.9 Mpps	90.56% +/- 1.24
E5_2650-oce Xeon & Emulex NIC	1.33 Mpps	1.33 Mpps	No diff. proven at 95.0% confidence
C2758-cxl Atom & Chelsio NIC	3.15 Mpps	4.2 Mpps	34.4% +/- 2.9%
C2758-ix Atom & Intel NIC	2.43 Mpps	3.08 Mpps	26% +/- 1.18
C2558-igb Atom & Intel NIC	1 Mpps	1.2 Mpps	20.17% +/- 2.56%
GX412-igb AMD & Intel NIC	747 Kpps	729 Kpps	-2.37% +/- 0.58%

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

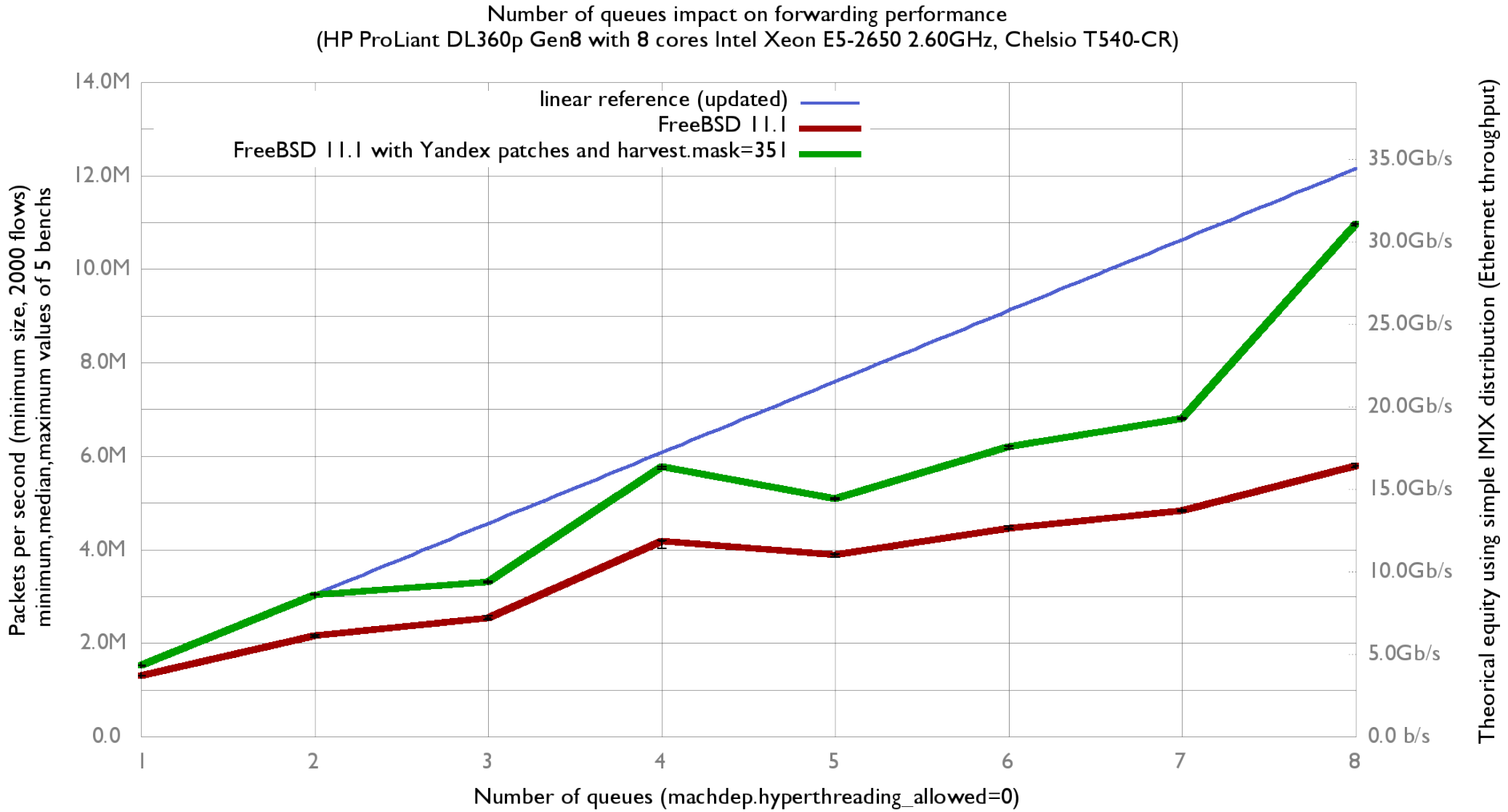
Tips 3: Use patches from Russia

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 (Chelsio, Intel, Mellanox)

Linear performance ?



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

Disabling LRO & TSO

Server	Enabled (default) median	Disabled median	ministat
E5_2650-cxl Xeon & Chelsio NIC	10.84 Mpps	10.92 Mpps	0.74% +/- 0.26%
C2758-cxl Atom & Chelsio NIC	4.20 Mpps	4.18 Mpps	No diff. proven at 95.0% confidence
C2758-ix Atom & Intel NIC	3.06 Mpps	3.06 Mpps	No diff. proven at 95.0% confidence
C2558-igb Atom & Intel NIC	1.2 Mpps	1.2 Mpps	No diff. proven at 95.0% confidence
GX412-igb AMD & intel NIC	729 Kpps	727 Kpps	No diff. proven at 95.0% confidence

Tips 5: Disable LRO & TSO on your router/firewall

hw.igb|ix.rx_process_limit

Server	100(igb), 256(ix), default median	-1 (disabled) median	ministat
C2758-ix Atom & Intel NIC	3.12 Mpps	3.85 Mpps	22.66% +/- 2.14%
C2558-igb Atom & Intel NIC	1.10 Mpps	1.13 Mpps	1.65% +/- 0.9%
GX412-igb AMD & Intel NIC	730 Kpps	735 Kpps	No diff. proven at 95.0% conf.

Tips 6: Disable rx_process_limit on igb&ix

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

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

- 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_2650-cxl, default: inet4 packets-per-second
+ Xeon E5_2650-cxl, IRQ pinned to CPU: inet4 packets-per-second
+-------------------------------------------------------------------------+
|                                                                       +   |
|xx  xx    x                                                  + +     +  +|
||___A___|                                                               |
|                                                             |___A_M_| |
+-------------------------------------------------------------------------+
    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-cxl, default: inet4 packets-per-second
+ Atom C2750-cxl, IRQ pinned to CPU: inet4 packets-per-second
+-------------------------------------------------------------------------+
|x  x                    + +             x         +     +x          +      x|
|   |_____________________________________A___M_____________________________|  |
|                         |_____________________A____M_____________|         |
+-------------------------------------------------------------------------+
    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
```

Tuning summary

- **Yandex‘s patches: AFDATA and RADIX locks**
- 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"
- etc/rc.conf
  - **harvest_mask="351"**
  - ifconfig_X="YYY -tso4 -tso6 -lro -vlanhwtso"

Before vs after tuning (IPv4)

setup	Default 11.1 (median)	Patched & tuned 11.1 (median)	ministat
E5_2650-cxl Xeon & Chelsio NIC	4.64 Mpps	11.15 Mpps	139.8% +/- 5.0%
E5_2650-oce Xeon & Emulex NIC	1.33 Mpps	1.33 Mpps	No diff. proven at 95.0% conf.
C2758-cxl Atom & Chelsio NIC	2.83 Mpps	4.19 Mpps	50.49% +/- 5.33%
C2758-ix Atom & Intel NIC	2.29 Mpps	3.85 Mpps	66.97% +/- 2.7%
C2558-igb Atom & Intel NIC	951 Kpps	1.13 Mpps	18.58% +/- 1.17%
GX412-igb AMD & Intel NIC	726 Kpps	735 Kpps	1.03% +/- 0.56%

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

inet4 vs inet6 performance

setup	inet4 (median)	inet6 (median)	ministat
E5_2650-cxl Xeon & Chelsio NIC	10.94 Mpps	9.18 Mpps	-16.12% +/- 0.19%
C2758-cxl Atom & Chelsio NIC	4.29 Mpps	3.43 Mpps	-19.08% +/- 1.61%
C2758-ix Atom & Intel NIC	3.81 Mpps	3.43 Mpps	-9.84% +/- 1.3%
C2558-igb Atom & Intel NIC	1.23 Mpps	1.08 Mpps	-11.79% +/- 0.5%
GX412-igb AMD & Intel NIC	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

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_2650-cxl, no VLAN tagging: inet4 packets-per-second
+ Xeon E5_2650-cxl, 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)
```

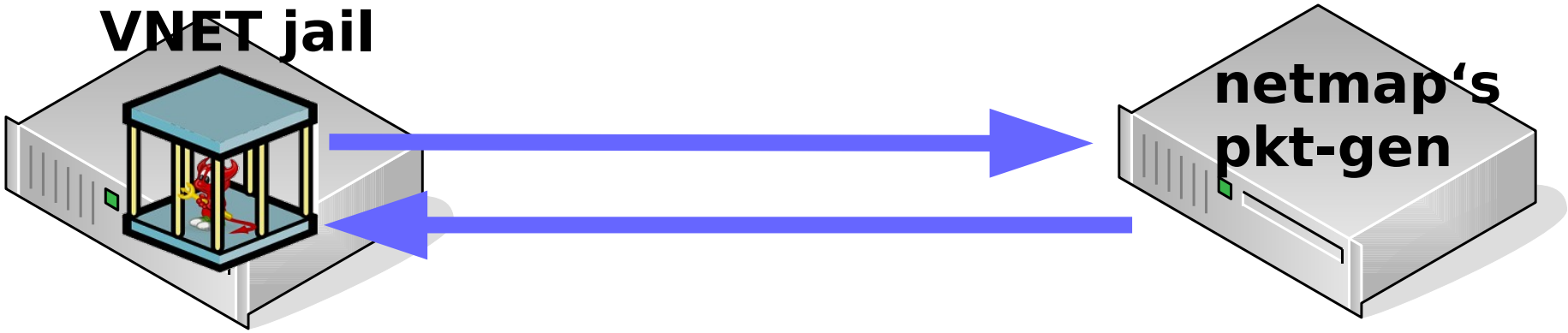
-17% with tagging: Known problem
Yet another patch from Yandex
https://reviews.freebsd.org/D12040

Adding VIMAGE support

```
options                 VIMAGE
```

E5_2650-cxl Xeon & Chelsio NIC	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

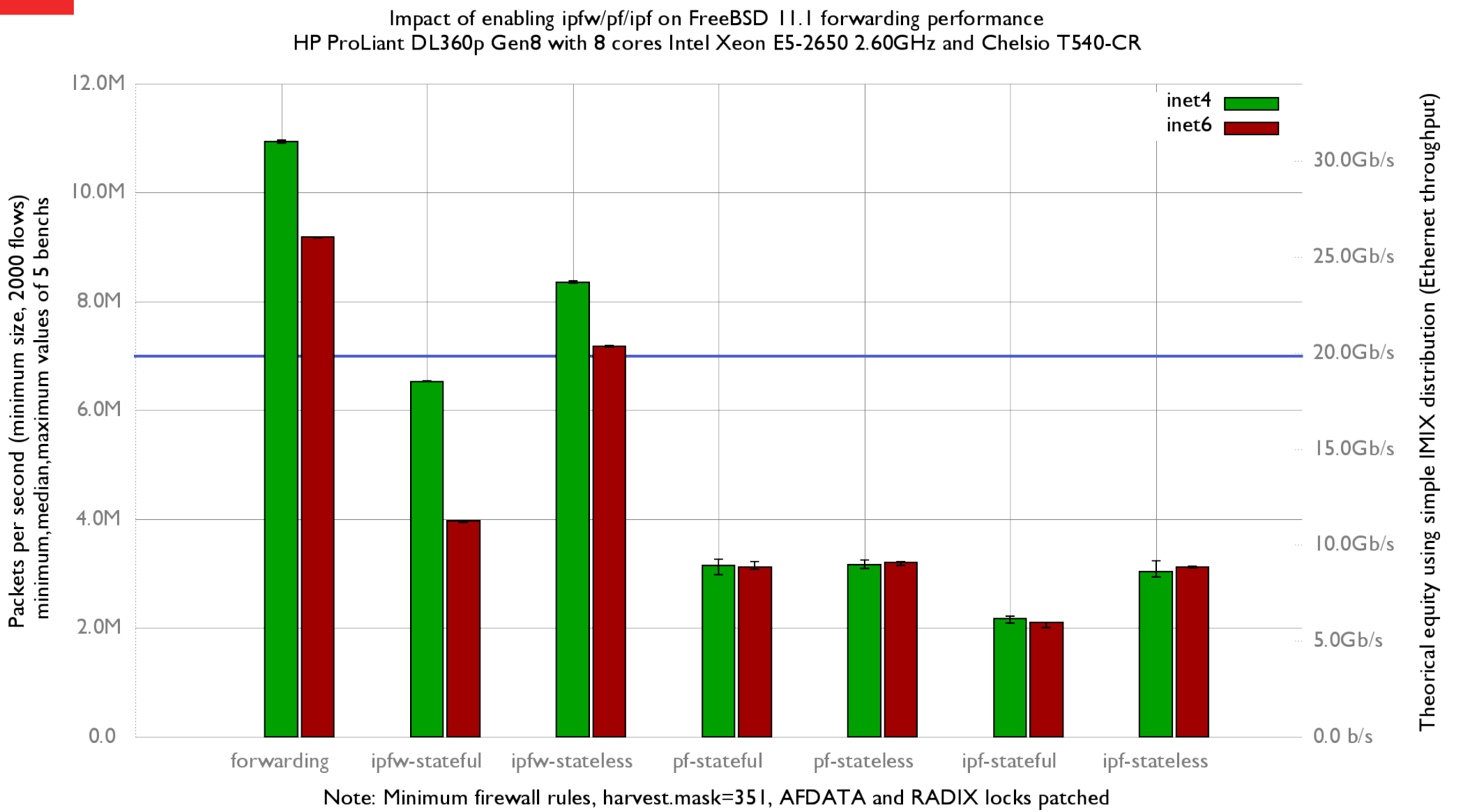
E5_2650-cxl Xeon & Chelsio NIC	No Jail (median) Mpps	VNET-Jail (median) Mpps	Ministat
inet 4 forwarding	10.8	11.0	No diff. proven at 95.0% confidence
inet 6 forwarding	10.0	10.0	No diff. proven at 95.0% confidence

VNET-jail rocks!

Firewalls

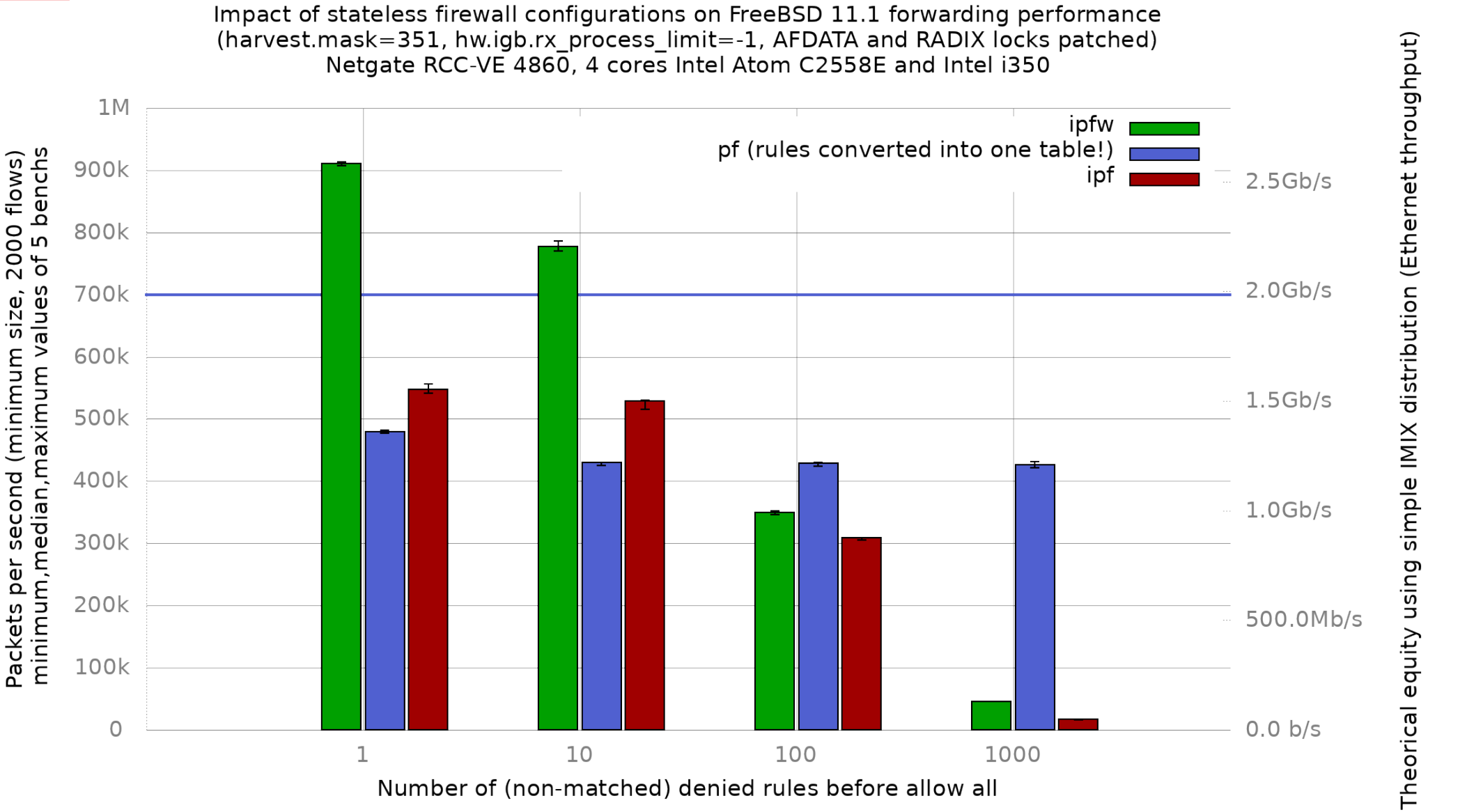
- How these impact PPS:
  - Enabling ipfw / pf / ipf with inet4 & inet6
  - Number of rules
  - Table size
  - Number of UDP flows

Impact of firewalls on PPS



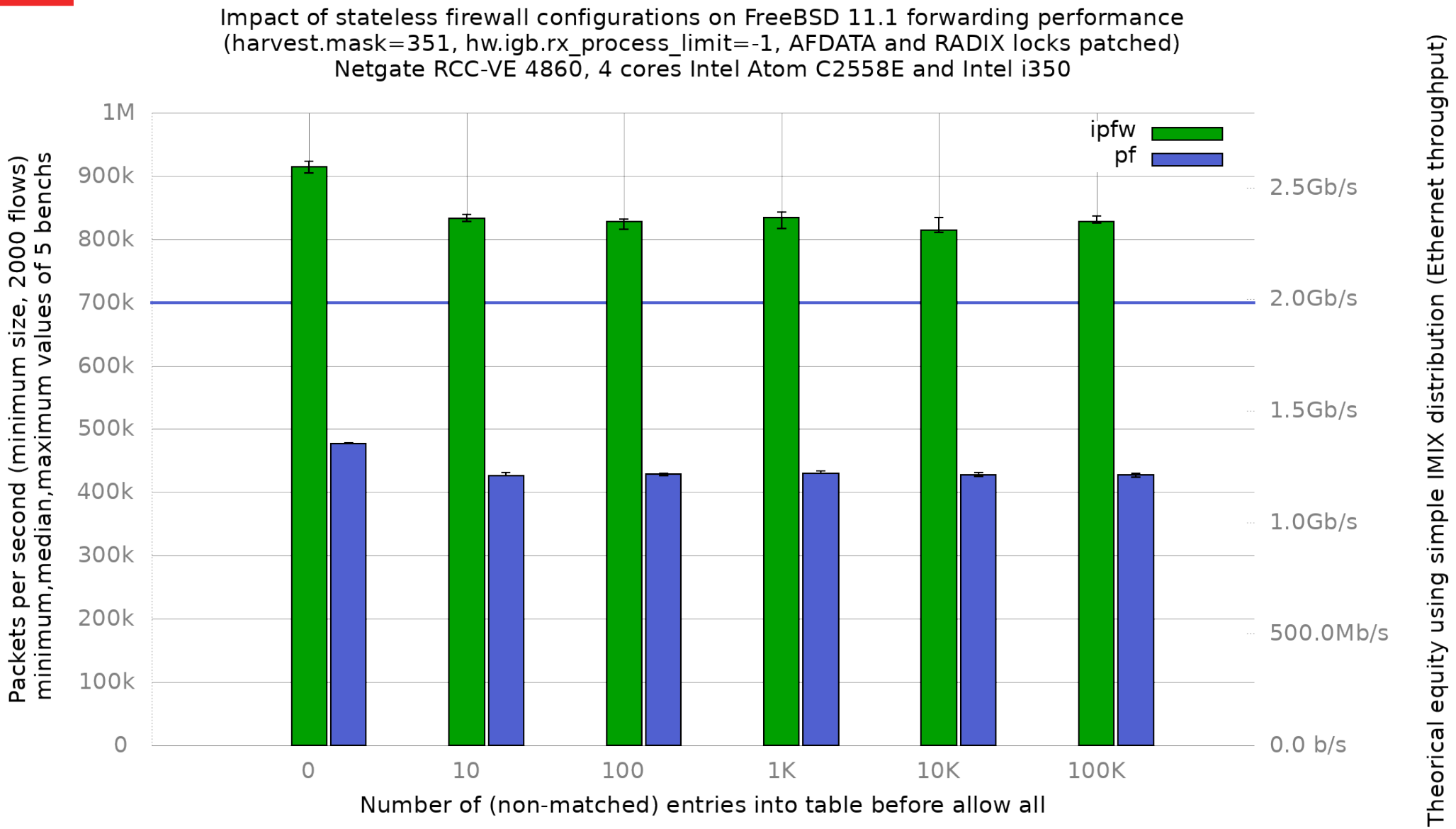
Warning: do not conclude a firewall is "better" than another with this bench

Stateless: rules impact



Keep MINIMUM numbers of rules with ipfw/ipf

Stateless: Table size impact



Use table

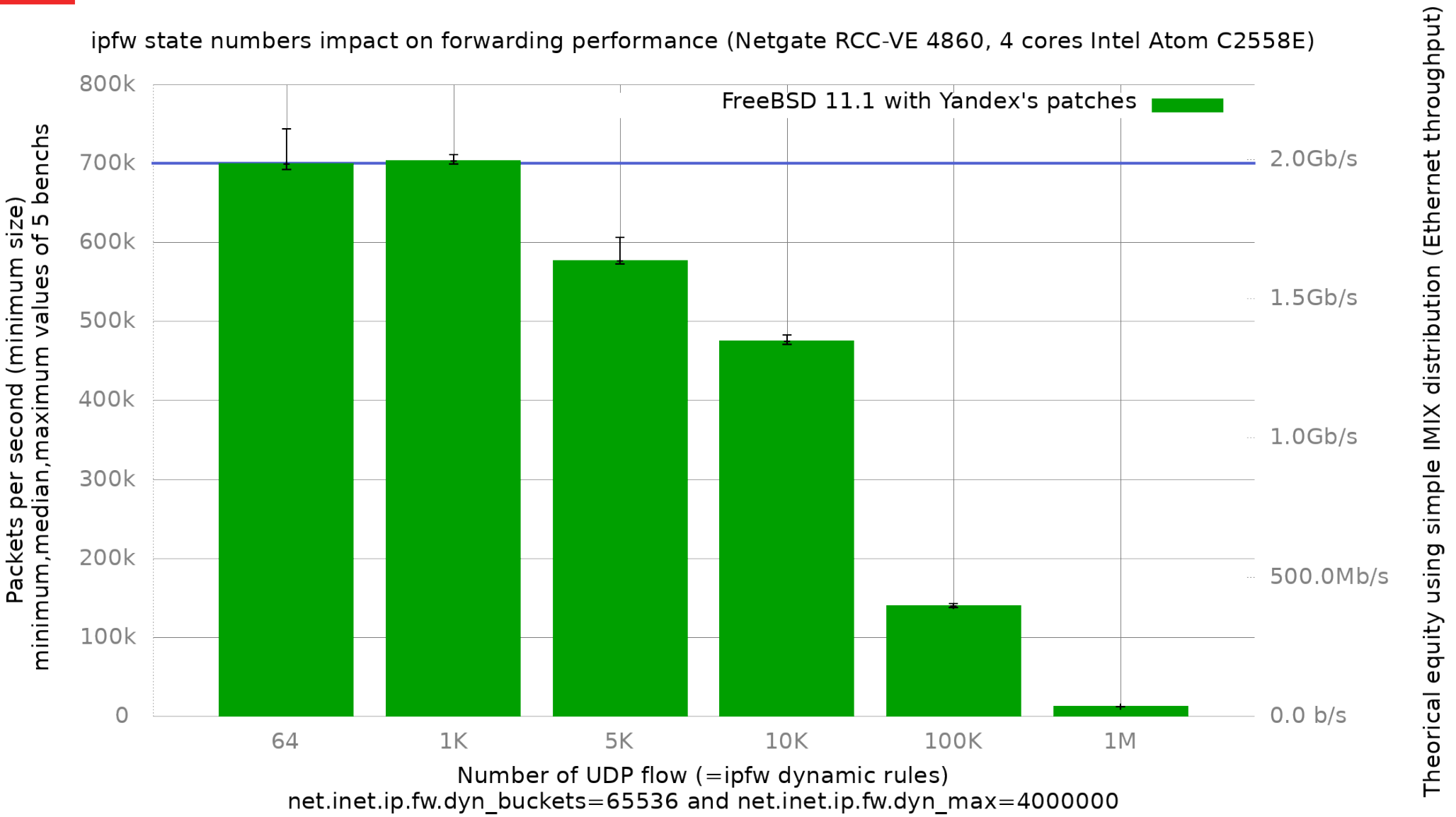
ipfw stateful: states impact

- One UDP flow create 1 state (dynamic rule)

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

keys	Default value	Maximum value
dynamic rules `net.inet.ip.fw.dyn_max`	16 384	*4 194 304*
hash table size [max_dyn / 64 ?] (power of 2) `net.inet.ip.fw.dyn_buckets`	256	65 536 (max)

ipfw stateful: states impact

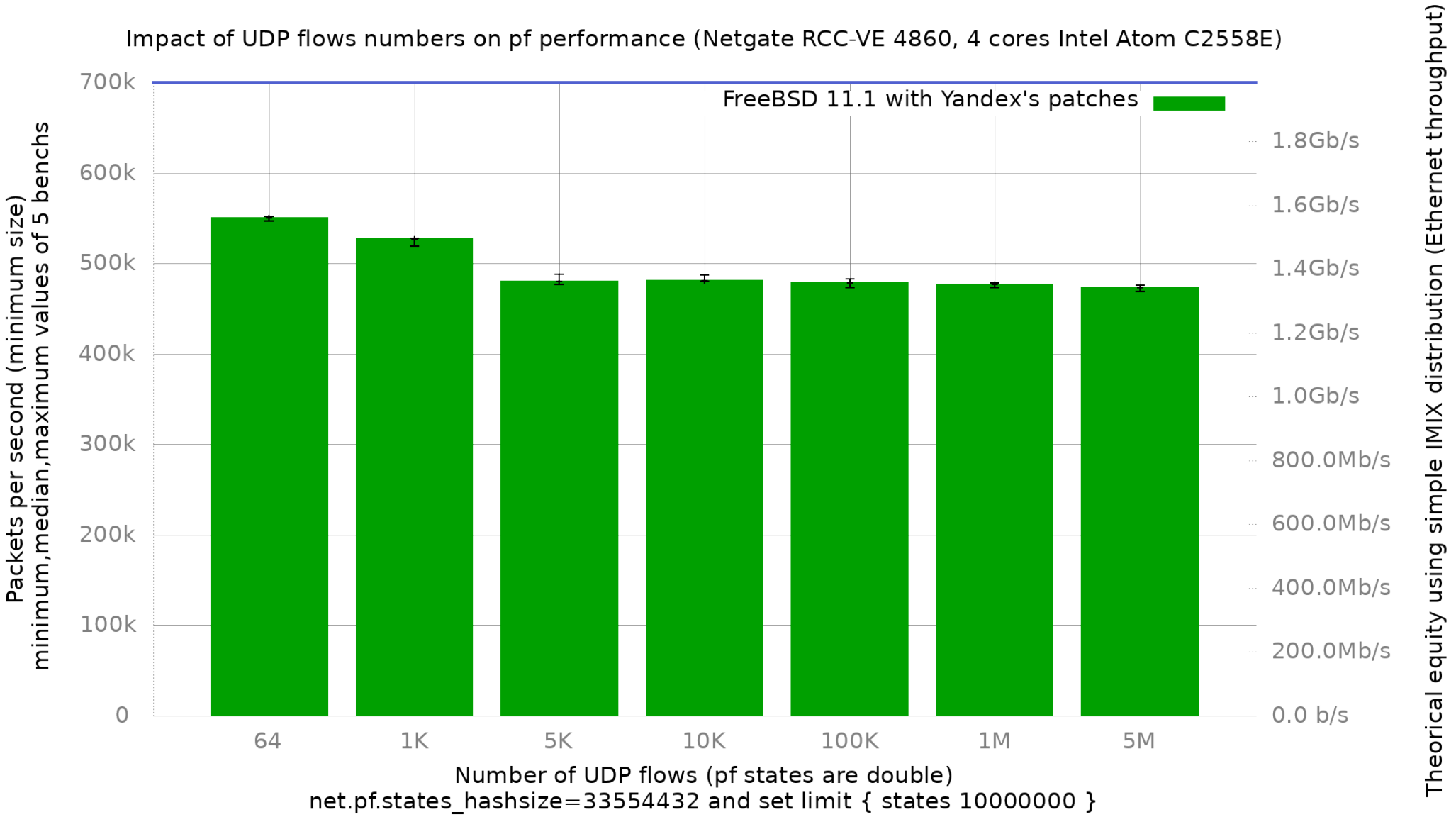


pf stateful: States impact

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

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

pf stateful: States impact



Note: For a stateful firewall… use pf

Back in 2012

- EuroBSDcon 2012 dev summit:

Changes since

- Still no MPLS
- User space forwarding solutions based on Intel‘s DPDK & Cisco‘s FD.io Vector Packet Processing lead the market... on Linux (12Mpps/core)
- And no "production ready" DPDK/Netmap forwarding solution on FreeBSD

Conclusion

- But kernel-space forwarding performance is still improved!

- Cf projects/routing

https://wiki.freebsd.org/ProjectsRoutingProposal

- Cf Nanako Momiyama‘s talk "IP Forwarding Fastpath" (EuroBSDCon 2016 & BSDCan 2017)

https://2016.eurobsdcon.org/PresentationSlides/NanakoMomiyama_TowardsFastIPForwarding.pdf

Resources

- Benches scripts, configurations, RAW results, flamegraph

https://github.com/ocochard/netbenches

- BSD Router Project (nanoBSD based on FreeBSD)

https://bsdrp.net

Questions ?

Thanks !