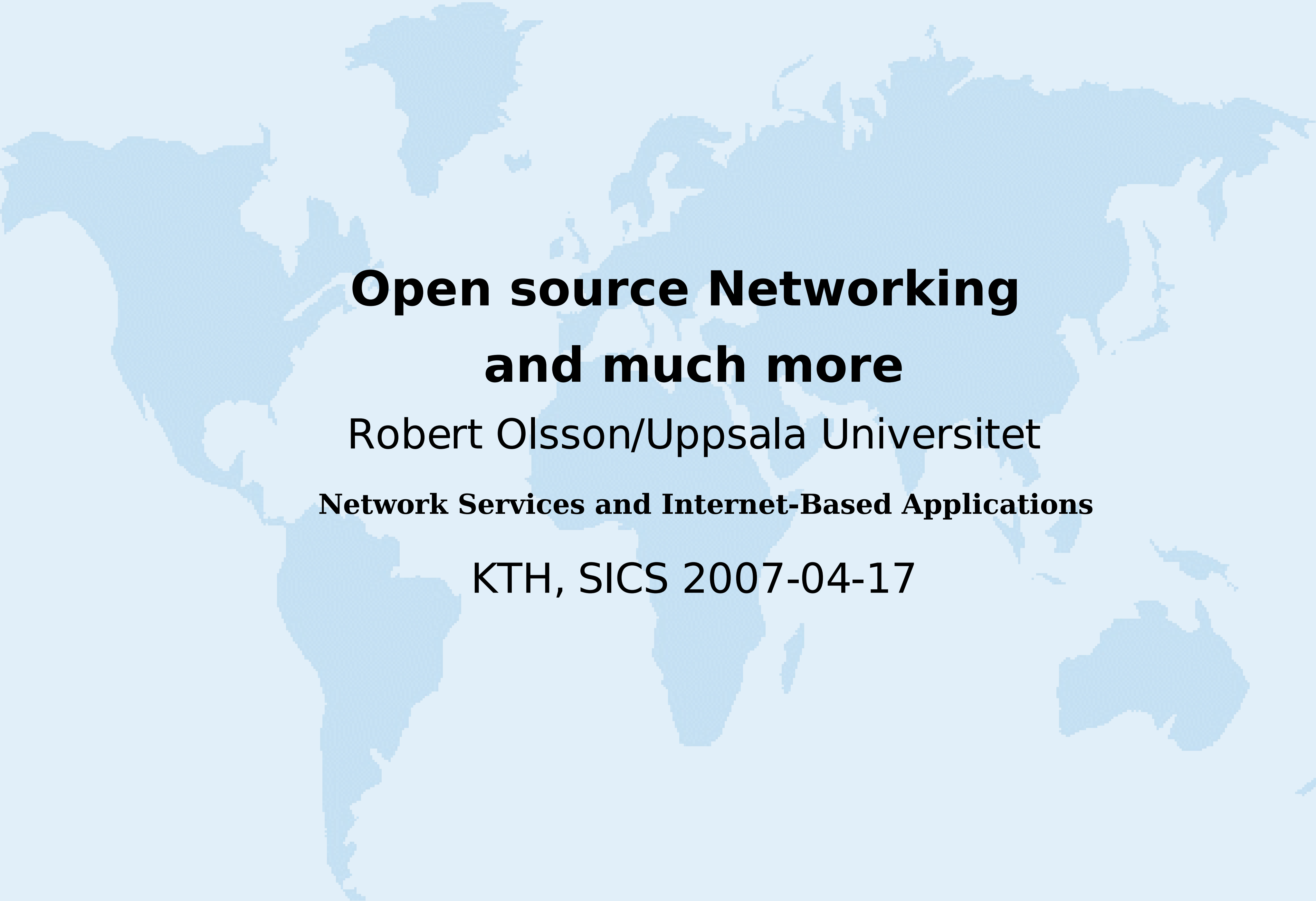


Open source Networking and much more

Robert Olsson/Uppsala Universitet

Network Services and Internet-Based Applications

KTH, SICS 2007-04-17

Fuskar i allt möjligt

Nätmanager. Alla möjliga tjänster plus IP-nät och routing,

Bildkodning

g3, g4, jpg (DCT) ISO etc.

Testverksamhet, Pilotprojekt, Linux – SUN SS5

Paketforwarding i Linux

IP-login/netlogin/nomad

IP-multicast kod PIM-SM, Zebra. Jens Låås.

IpInfusion köpte kod

MBGP för Zebra

IRDP för Zebra/Quagga

Koncept för bifrost svensk Linux distribution

Samarbete med Linux-utvecklare och industri

Fuskar i allt möjlig/forts

Alla sorts tester. Kreativ. El cheapo. Burkar vara referens.
Man måste veta vad man testar...
Pktgen används över hela världen. Säljs också.

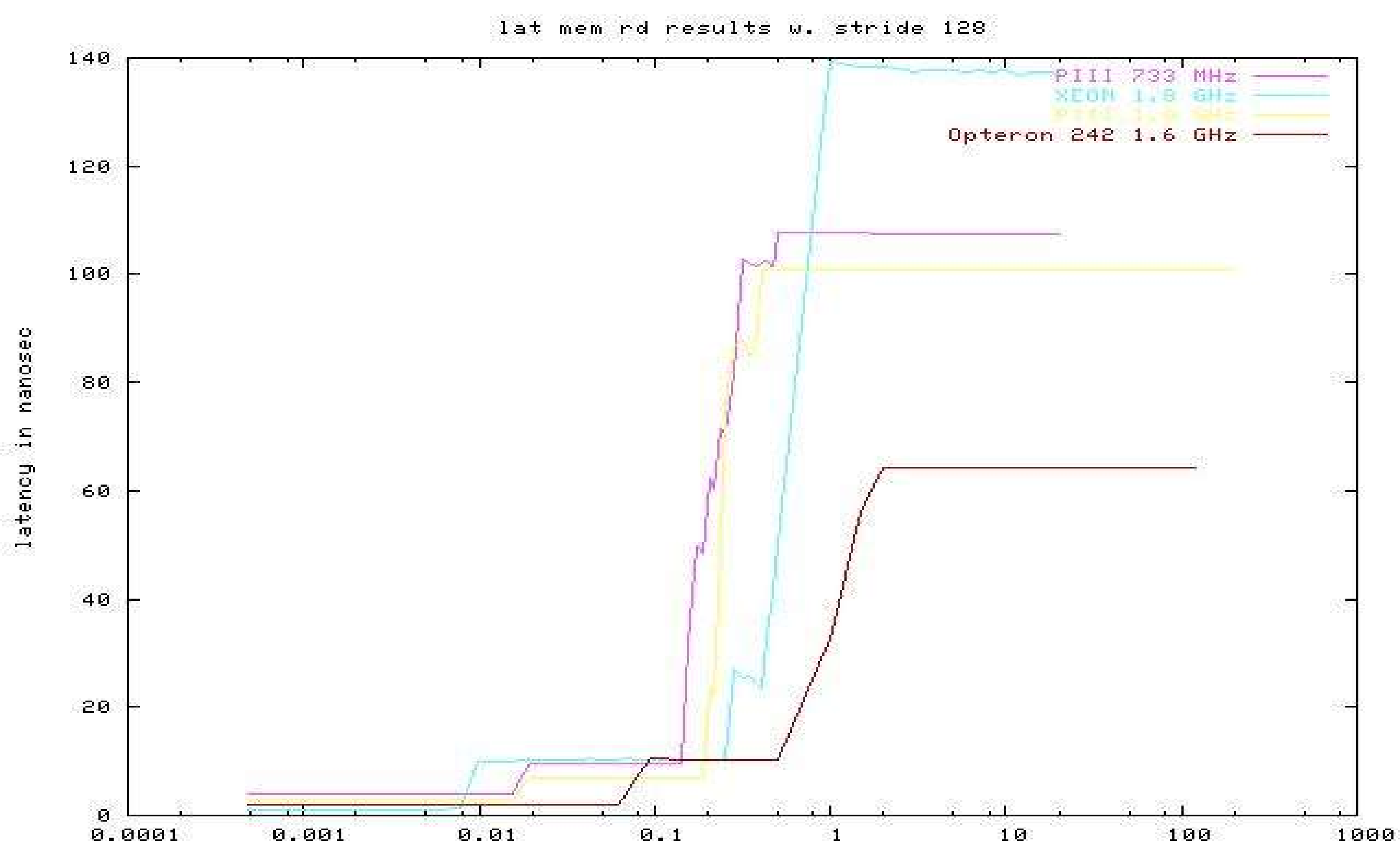
Routing/ Nätverksprestanda.
Polling. NAPI. 3 år. Startade i OLS Ottawa.
Multiprocessor prestanda för nätverk. Alexey Kuznetsov.
NordUsenix. Ip6 tester.
route cache tuneing, statistik, rtstat

Fuskar i allt möjligt/forts

Hårvarugenomgångar. Chipset etc etc.
fib_trie med Jens Låås, Hans Liss. 1 år
TRASH med Stefan Nilsson. Värt ett bättre öde.

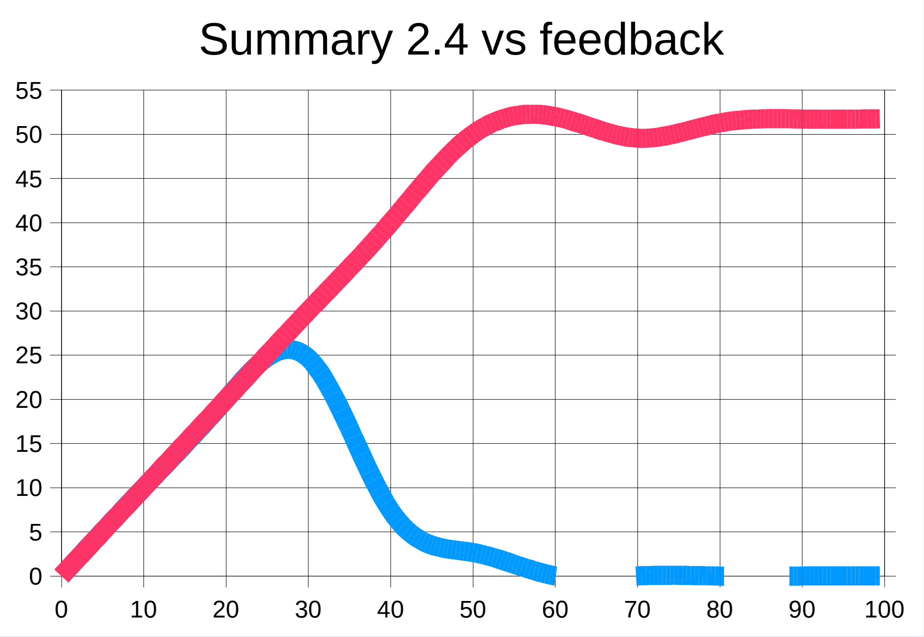
OpenWrt, WLAN tester.

Cache effect/Performance



Overall Effect

- Inelegant handling of heavy net loads
  - System collapse
- Scalabiity affected
  - System and number of NICS
    - A single hogger netdev can bring the system to its knees and deny service to others



March 15 report on lkml

Thread: "How to optimize routing perfomance"

reported by

Marten.Wikstron@framsfab.se

- Linux 2.4 peaks at 27Kpps

- Pentium Pro 200, 64MB RAM

Cache effect/Performance

Cache line 32 – 128 bytes

Optimize struct for cache and multiprocessors usage

PIO even worse then cache miss

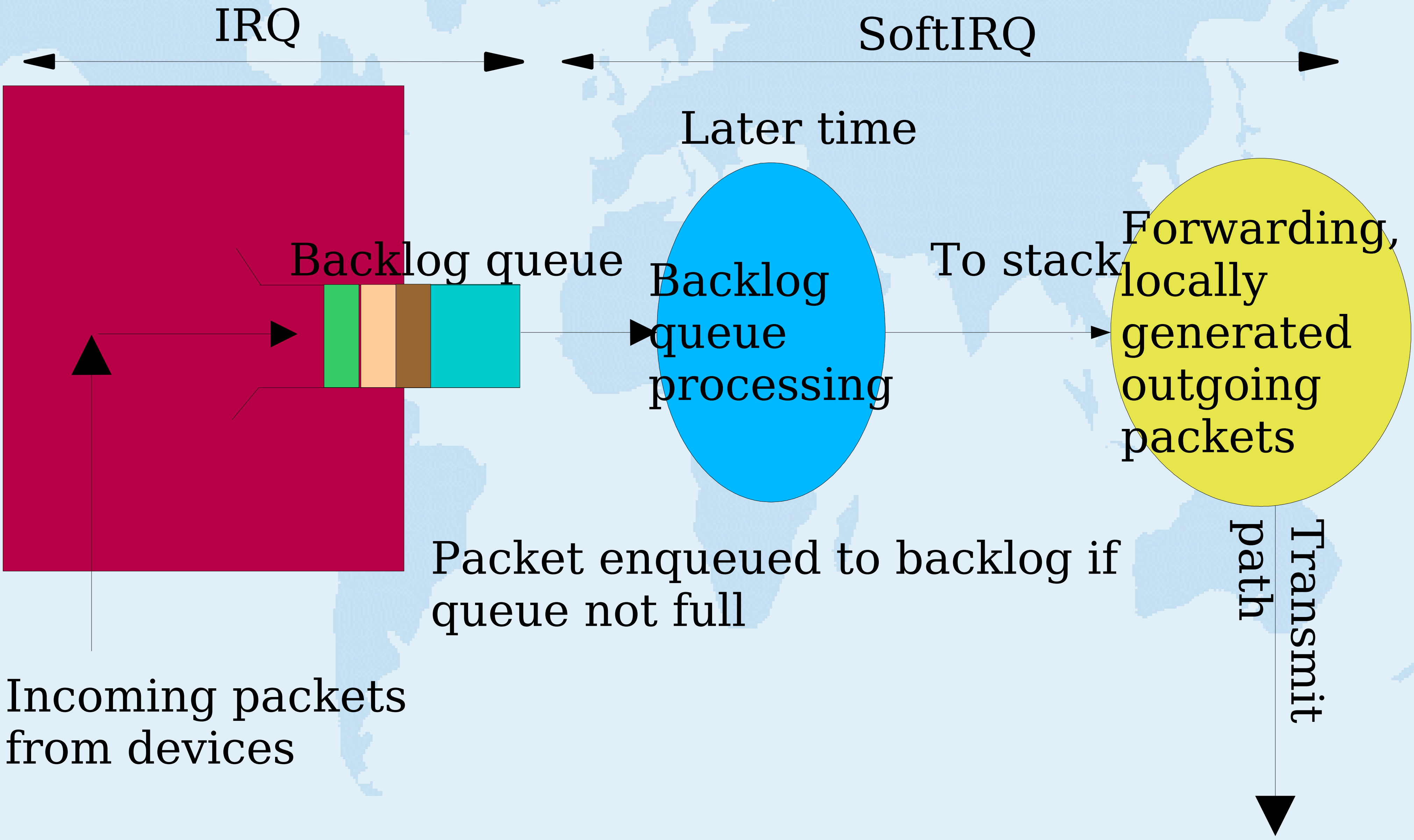
PIO READ stalls CPU

PIO WRITE can be posted

DMA copies of data into RAM

Does prefetch solve problems?

Looking inside the box



Packet enqueued to backlog if queue not full

Incoming packets from devices

BYE BYE Backlog queue

- Packet stays in original queue (eg DMA)
- Netrx softirq
  - **foreach dev in poll list**
    - Calls *dev->poll()* to grab upto *quota* packets
    - Device driver are polled from softirq and pkts are pulled and delivered to network stack.
    - Dev driver indicates done/ notdone.
      - Done ==> we go back to IRQ mode.
      - Nodone ==> device remain on polling list
      - Breakes the netrx softirq at one jiffie or netdev_max_backlog
      - This to ensure other taskes to run

Kernel support

NAPI kernel part was included in:
2.5.7 and back ported to 2.4.20

Current driver support:

e1000	Intel GIGE NIC's
tg3	BroadCom GIGE NIC's
dl2k	D-Link GIGE NIC's
tulip (pending)	100 Mbs

NAPI: observations & issues

Ooh I get even more interrupts.... with polling.

As we seen NAPI is an interrupt/polling hybrid. NAPI uses interrupts to guarantee low latency and at high loads interrupts never gets re-enabled. Consecutive polling occur.

Old scheme added interrupt delay to handle CPU from being killed by interrupts.

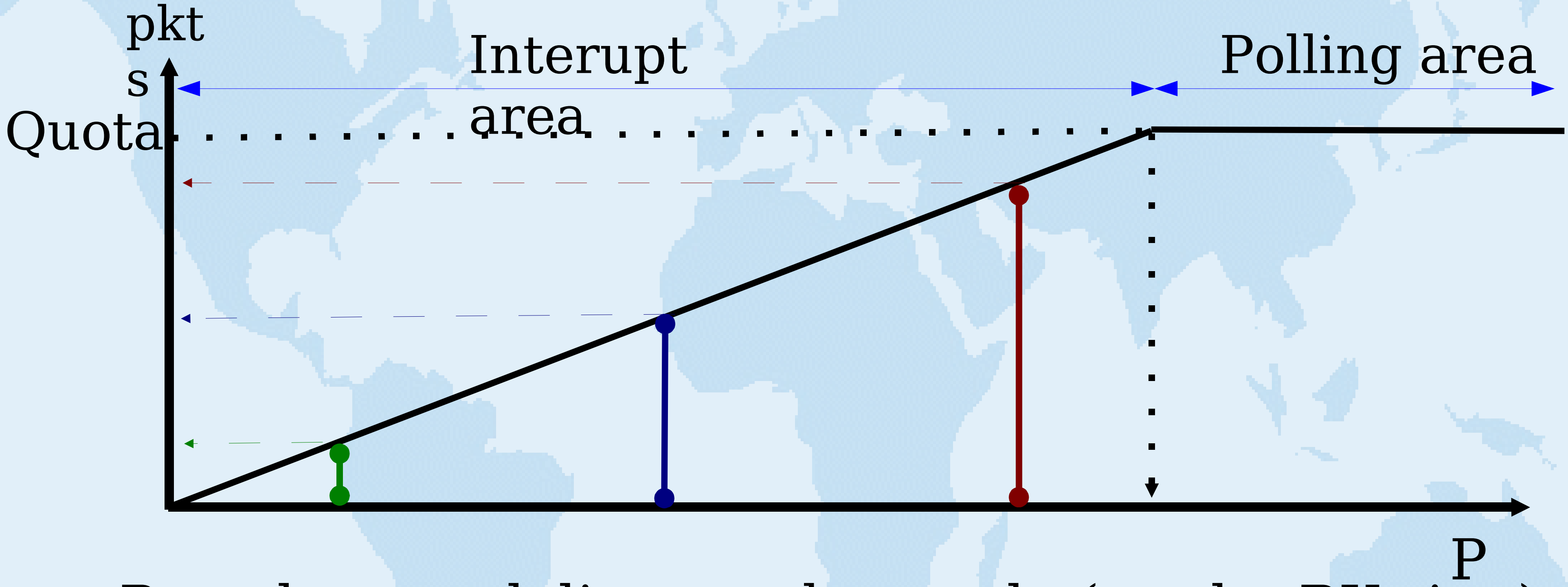
In the NAPI case we can do without this delay for the first time but it means more interrupts in low load situations.

Should we add interrupt delay just of old habit?

Core Problems

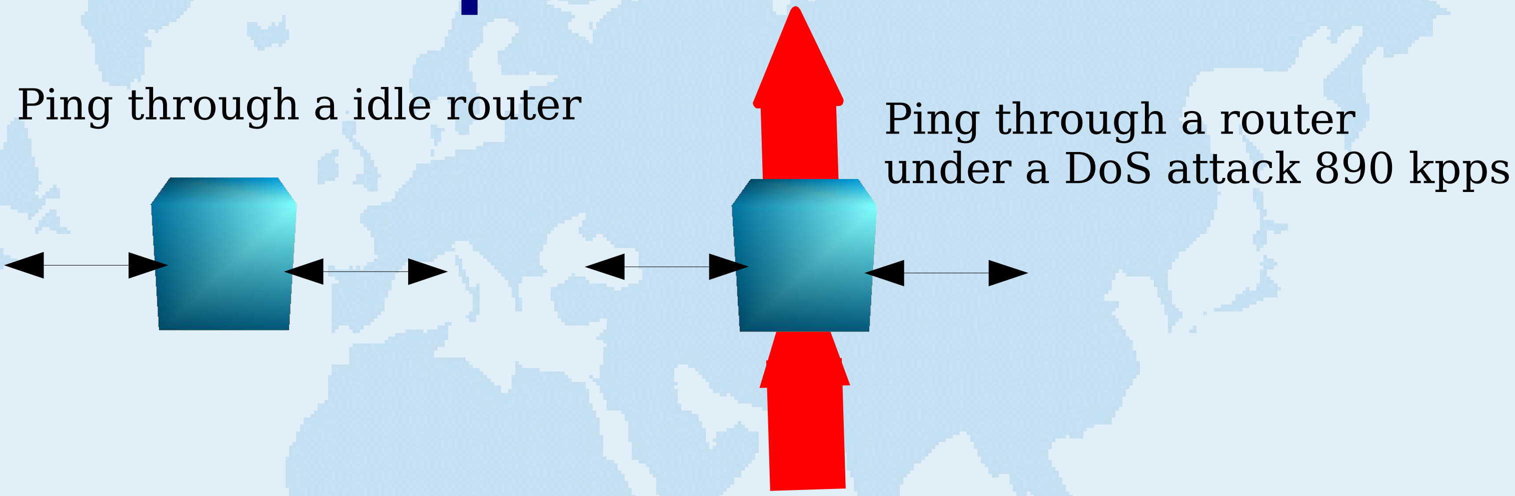
- heavy net load: system congestion collapse
  - High Interupt rates
    - Livelock and Cache locality effects
    - Interupts are just simply expensive
  - CPU
    - interupt driven: takes too long to drop bad packet
  - Bus (PCI)
    - Packets still being DMAed when system overloaded
  - Memory bandwidth
    - Continous allocs and frees to fill DMA rings
- Unfairness in case of a hogger netdev

A high level view of new system



- ➔P packets to deliver to the stack (on the RX ring)
- ➔Horizontal line shows different netdevs with different i
- ➔Area under curve shows how many packets before next
- ➔Quota enforces fair share

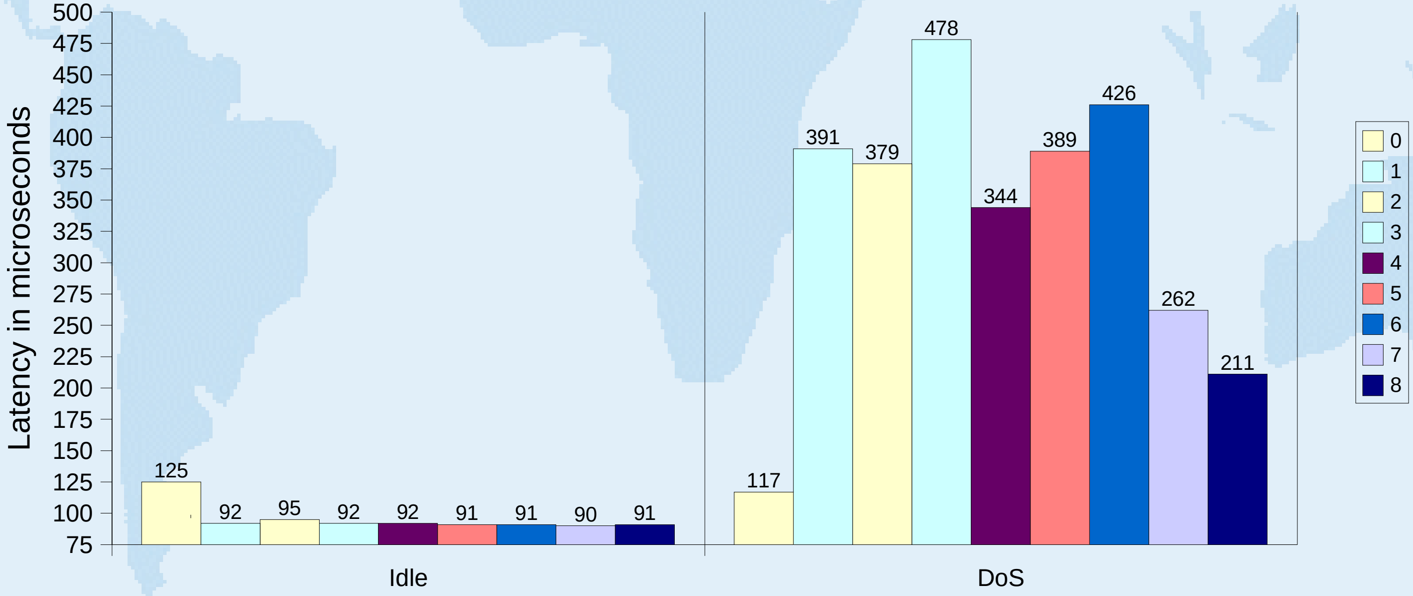
Some GIGE experiments/NAPI



Ping through a idle router

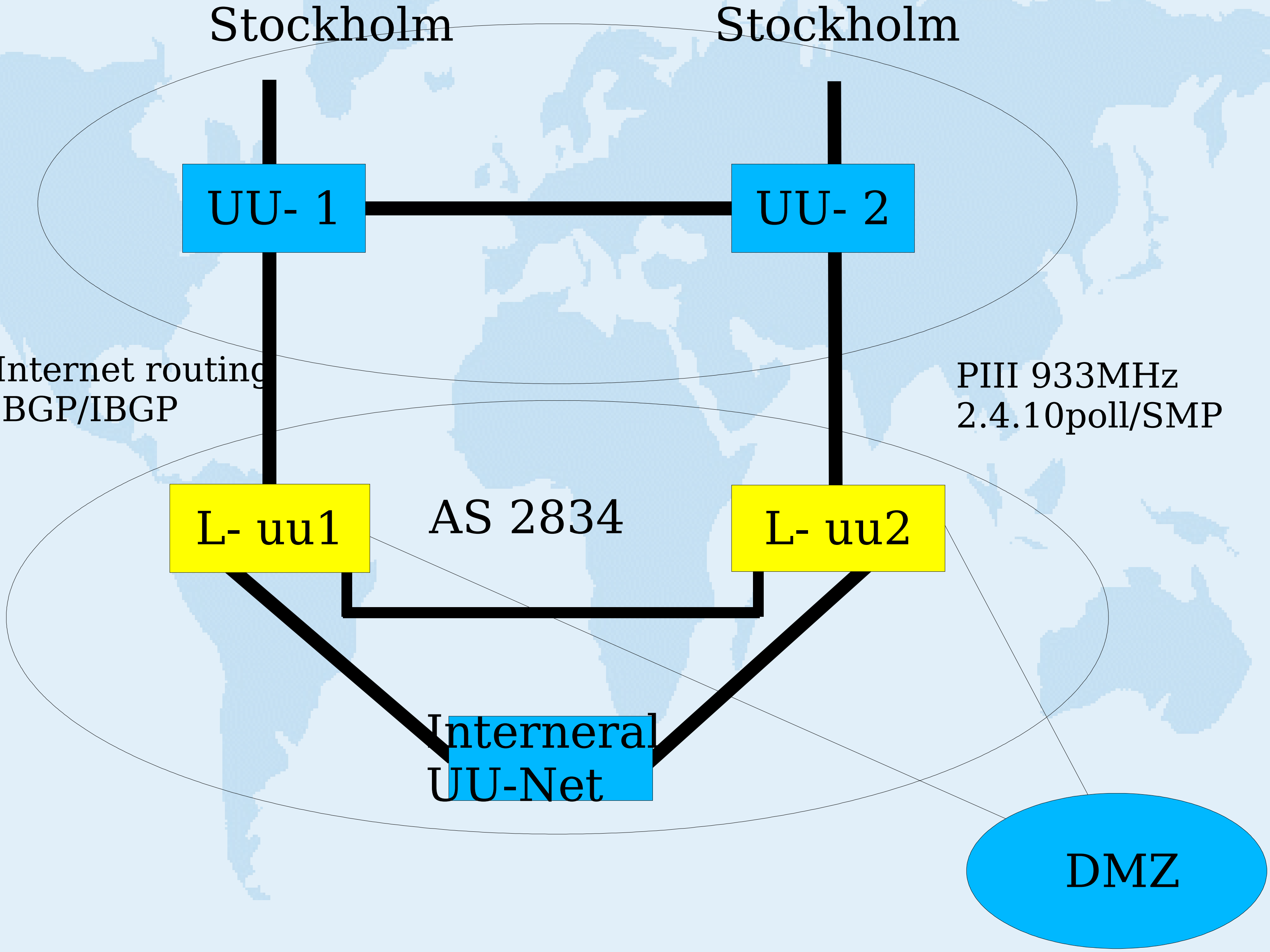
Ping through a router under a DoS attack 890 kpps

Ping latency/fairness under xtreme load/UP



Very well behaved just an increase a couple of 100 microsec !!

NAPI/SMP production in use: uu.se



Stockholm

Stockholm

UU- 1

UU- 2

Full Internet routing
via EBGP/IBGP

PIII 933MHz
2.4.10poll/SMP

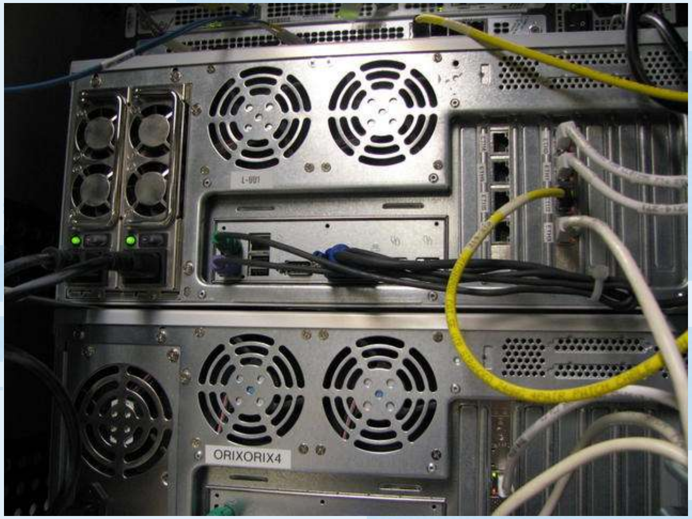
L- uu1

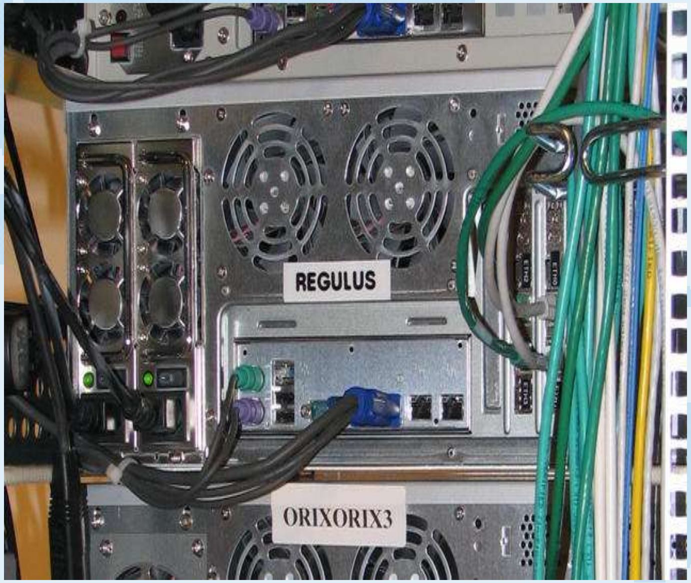
AS 2834

L- uu2

Interneral
UU-Net

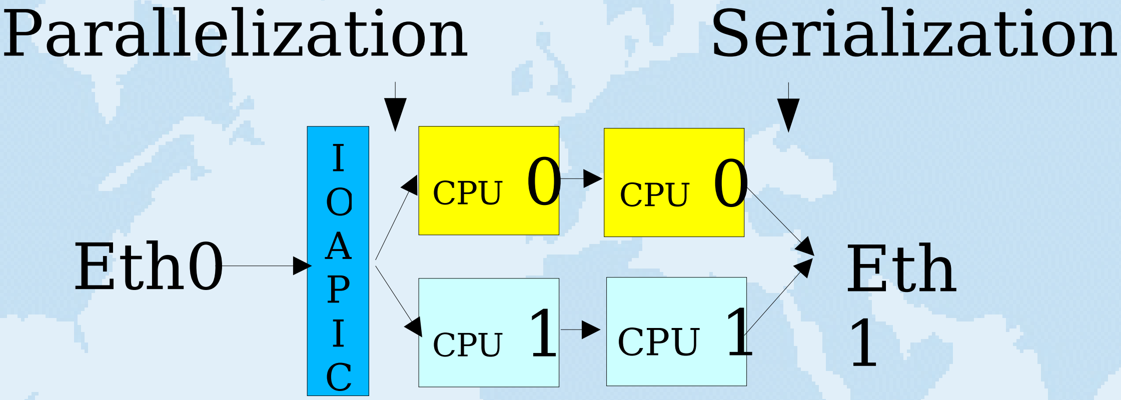
DMZ

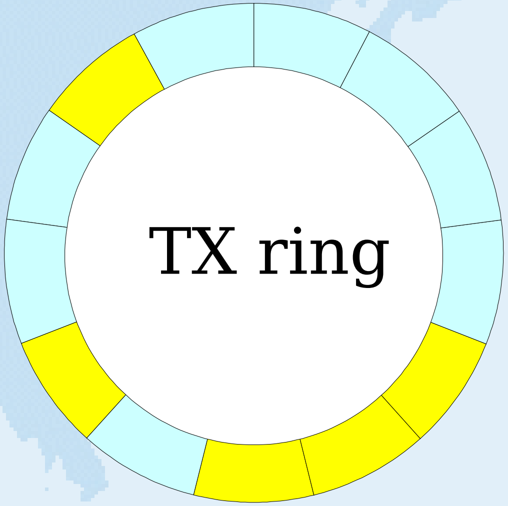

ORIXORIX4


REGULUS
ORIXORIX3

IBM
ARCHIVE-1
ARCHIVE-2

R&D





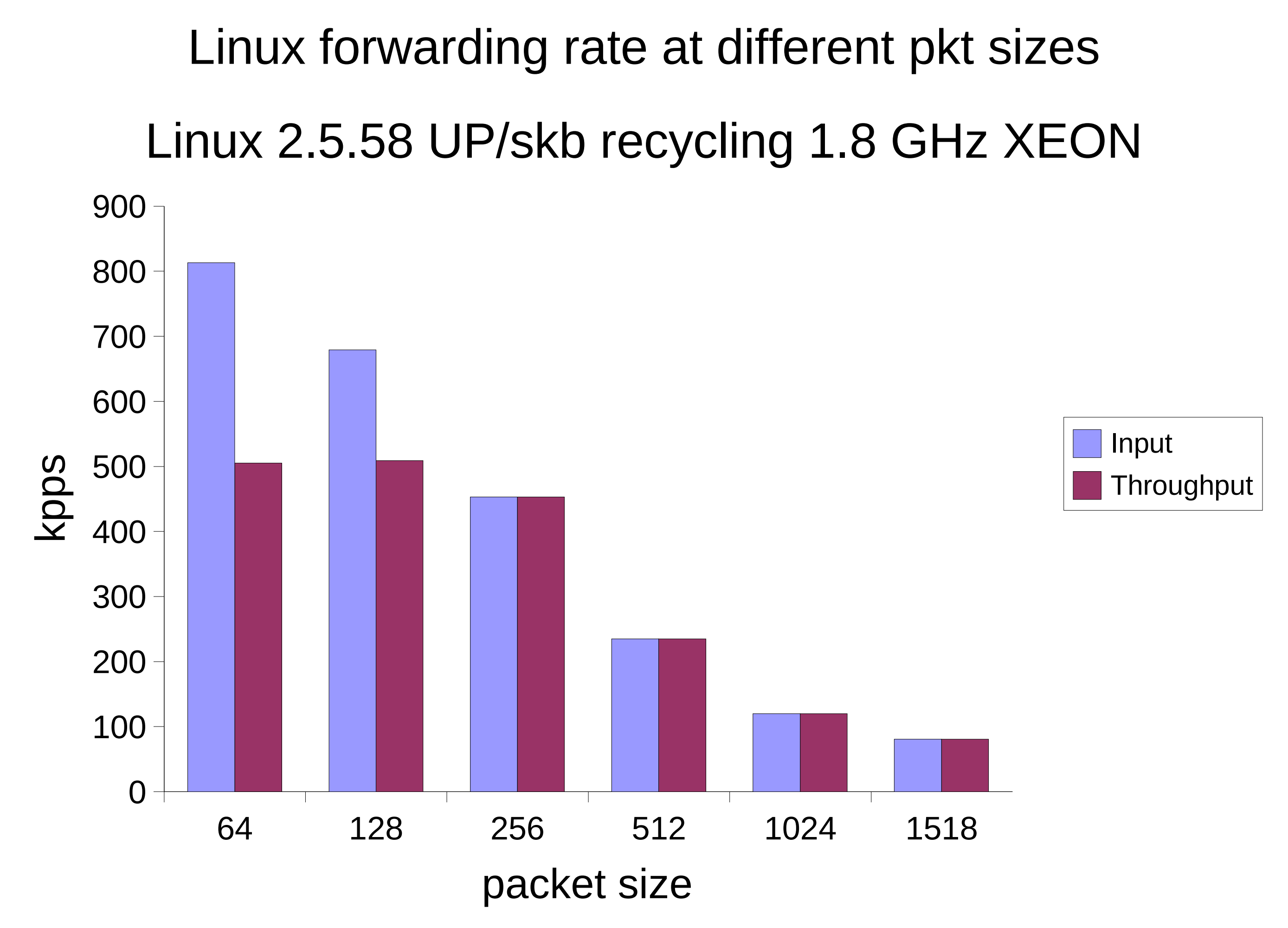
For user apps new scheduler
does affinty

Eth1 holds skb's
from different CPU's
Clearing TX-buff
releases cache bouncing

But for packet forwarding....
eth0->eth1 CPU0 (we can set
affinity eth1 -> CPU0)

But it would be nice to other
CPU for forwarding too. :-)

Forwarding performance



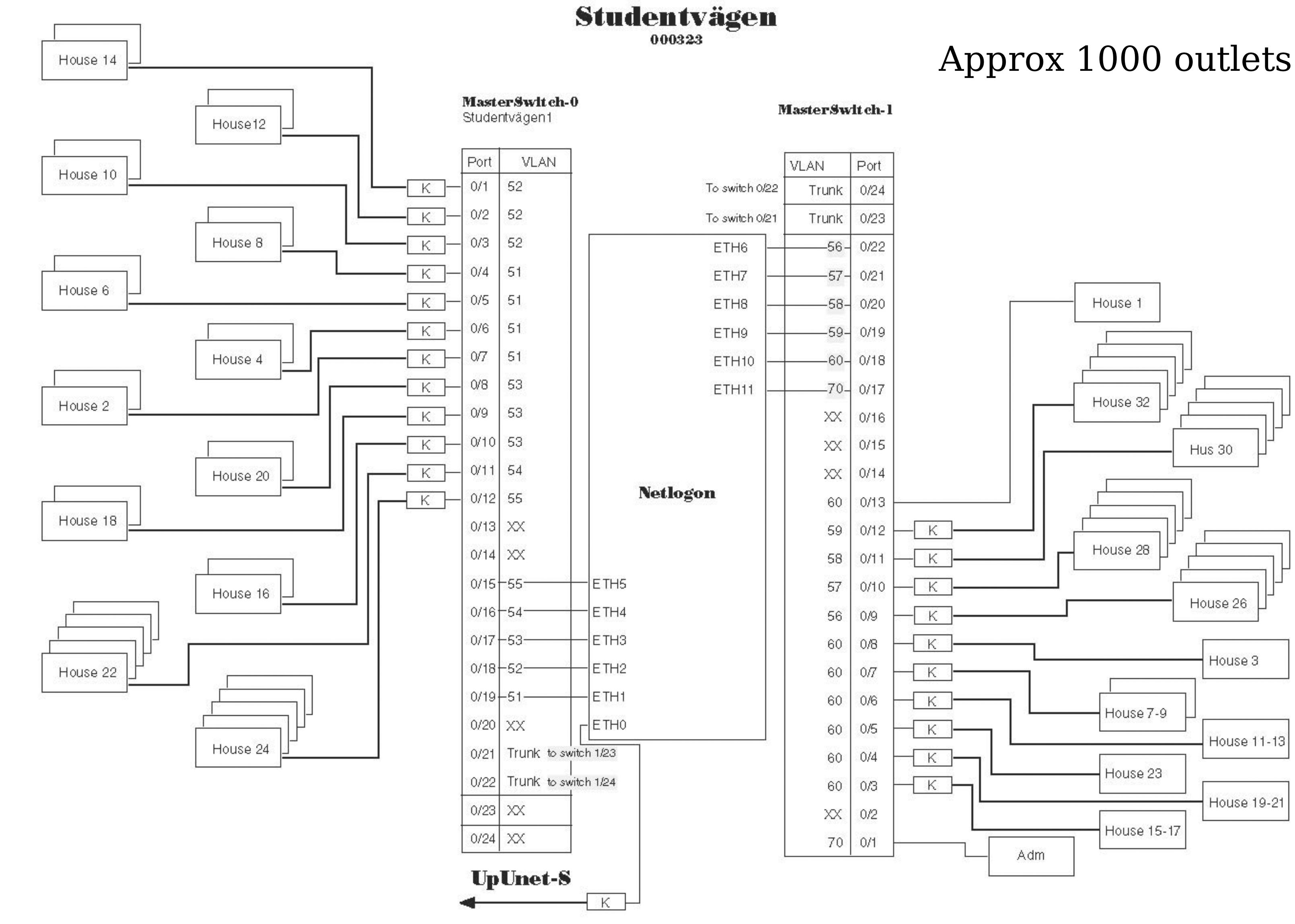
Fills a GIGE pipe -- starting from256byte pkts

Bifrost concept

- Linux kernel collaboration
  - FASTROUTE, HW_FLOWCONTROL, New NAPI for network stack.
- Performance testing, development of tools and testing techniques
- Hardware validation, support from big vendors
- Detect and cure problems in lab not in the network infrastructure.
- Test deploy (Often in own network)

IP-login installation

at Uppsala University



Studentvägen

000323

Approx 1000 outlets

MasterSwitch-0
Studentvägen1

Port	VLAN
0/1	52
0/2	52
0/3	52
0/4	51
0/5	51
0/6	51
0/7	51
0/8	53
0/9	53
0/10	53
0/11	54
0/12	55
0/13	XX
0/14	XX
0/15	55
0/16	54
0/17	53
0/18	52
0/19	51
0/20	XX
0/21	Trunk to switch 1/23
0/22	Trunk to switch 1/24
0/23	XX
0/24	XX

MasterSwitch-1

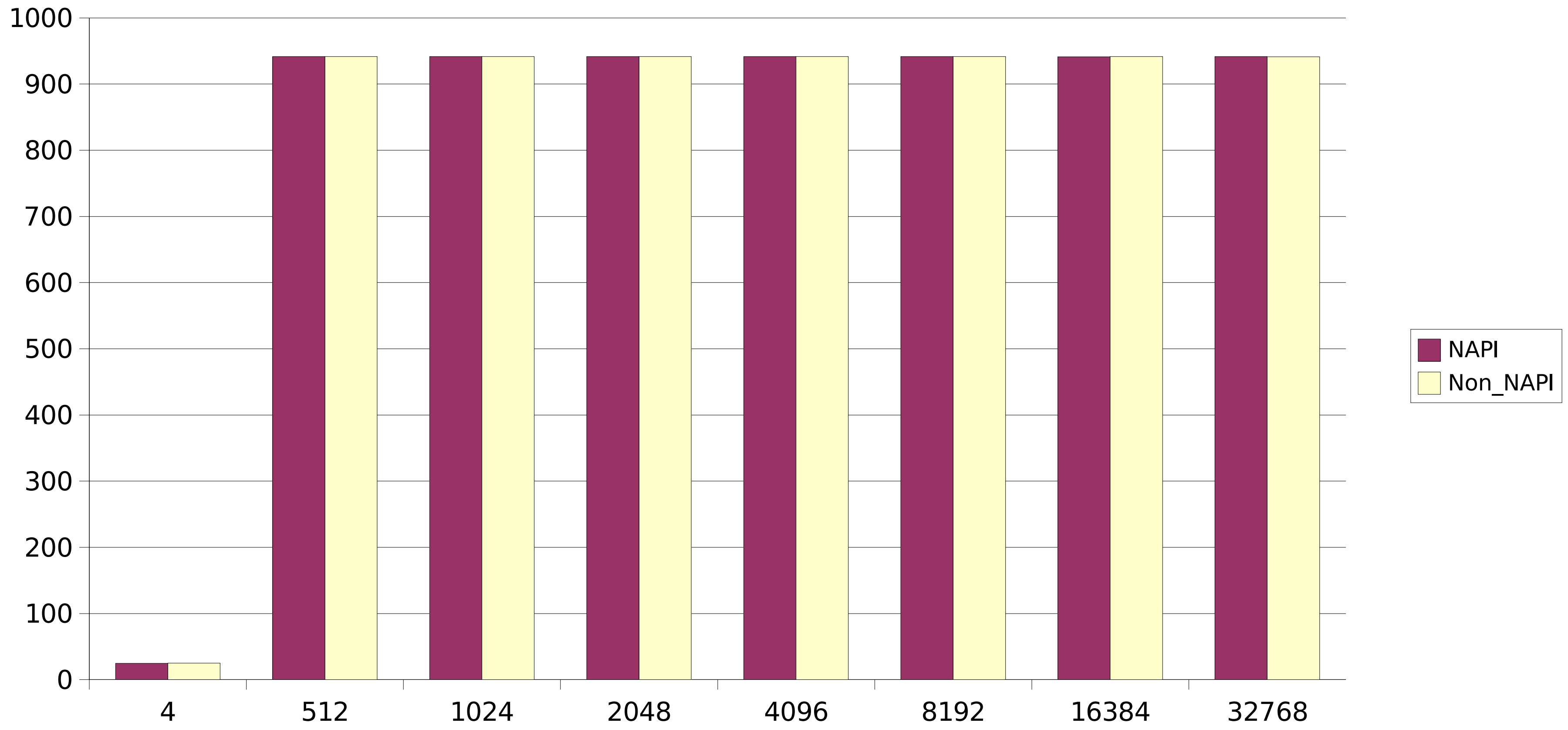
	VLAN	Port
To switch 0/22	Trunk	0/24
To switch 0/21	Trunk	0/23
	56	0/22
	57	0/21
	58	0/20
	59	0/19
	60	0/18
	70	0/17
	XX	0/16
	XX	0/15
	XX	0/14
	60	0/13
	59	0/12
	58	0/11
	57	0/10
	56	0/9
	60	0/8
	60	0/7
	60	0/6
	60	0/5
	60	0/4
	60	0/3
	XX	0/2
	70	0/1

Netconf 2005

Robert Olsson

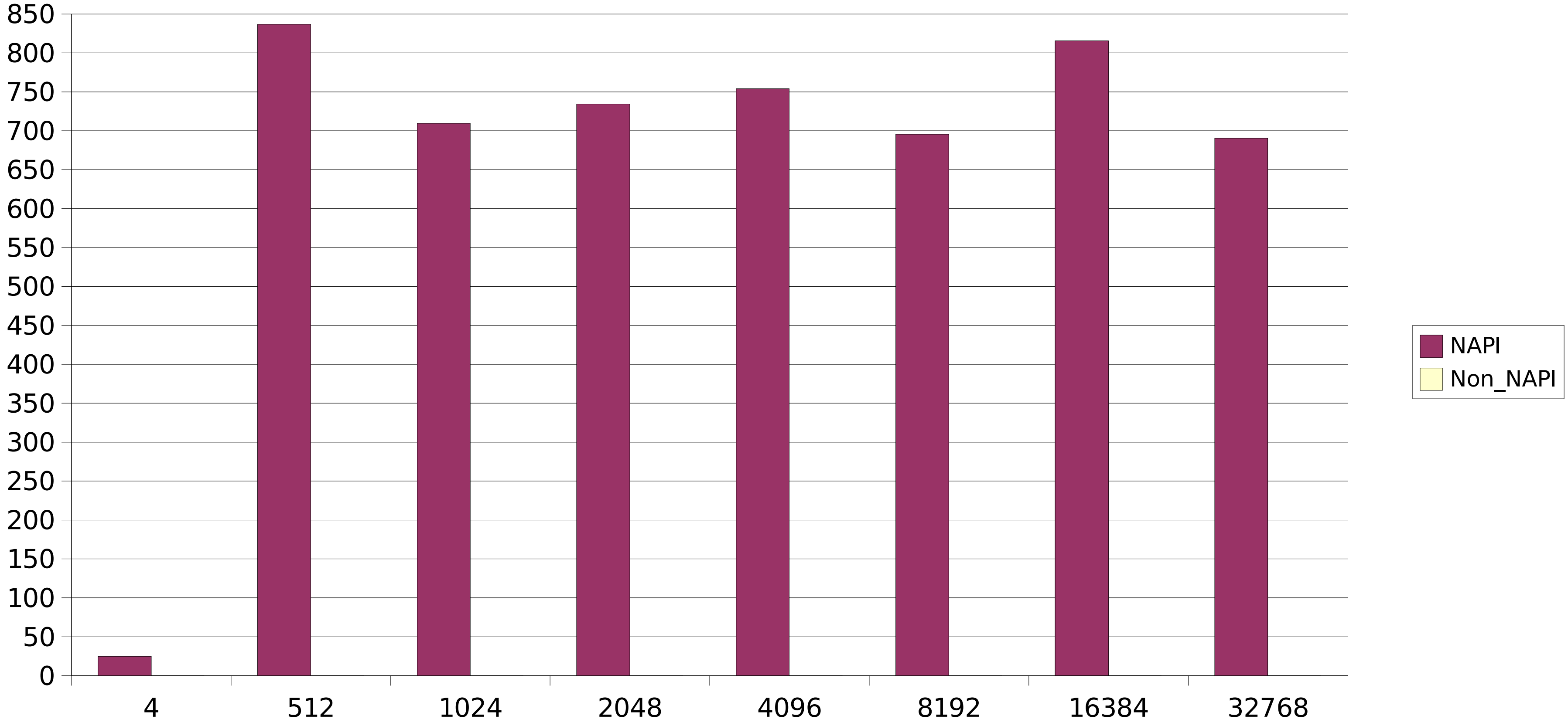
Experiments & Experiences
with FIB lookup and route cache

TCP performance



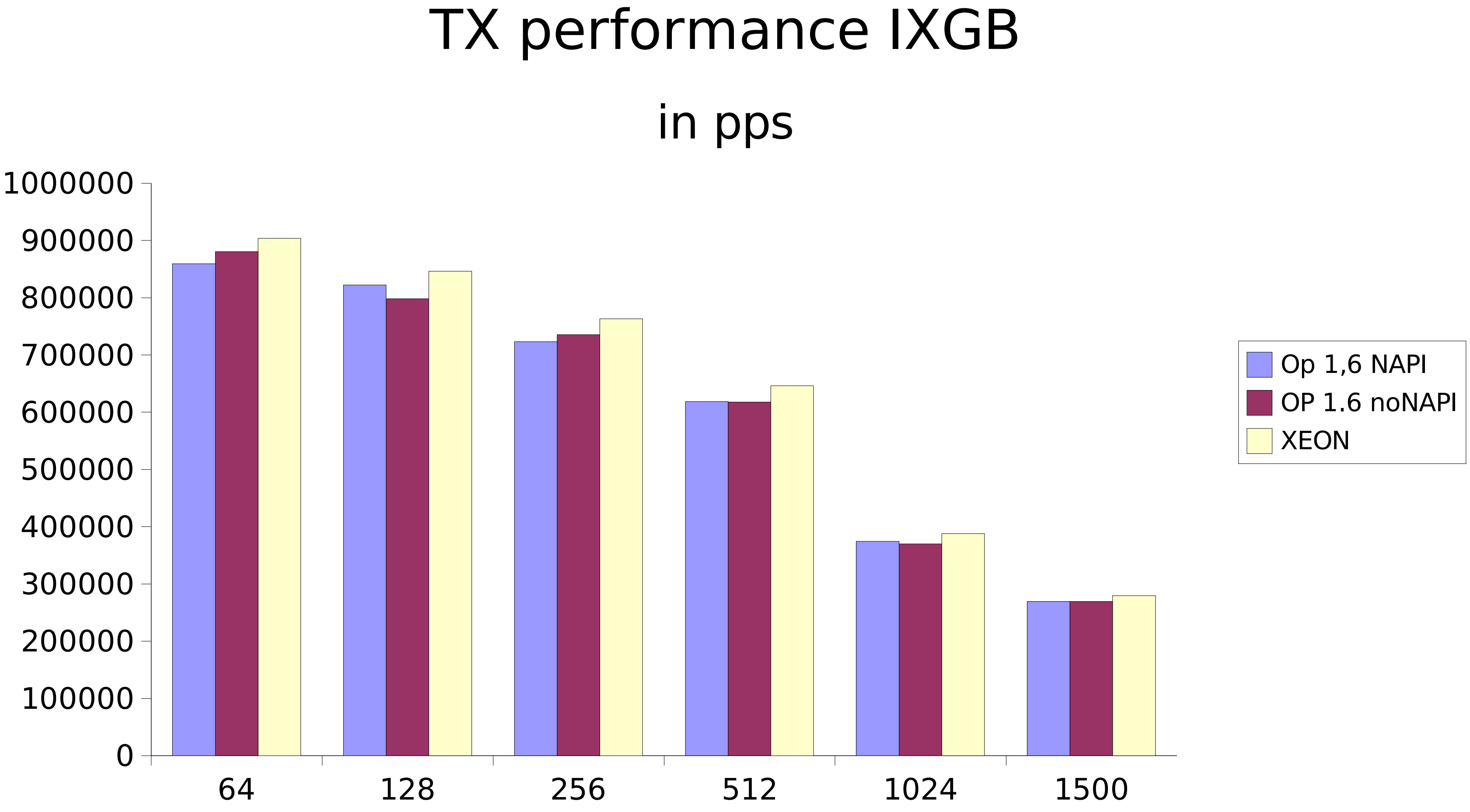
2.6.11.7 SMP kernel using one CPU driver e1000 NAPI - no-NAPI. Opteron 1.6 GHz e1000 w 82546GB.

TCP performancewhen receiving DoS on other NIC



2.6.11.7 SMP kernel using one CPU driver e1000 NAPI - no-NAPI. Opteron 1.6 GHz e1000 w 82546GB.

10 GbE early days



Other activities
informal linux agenda

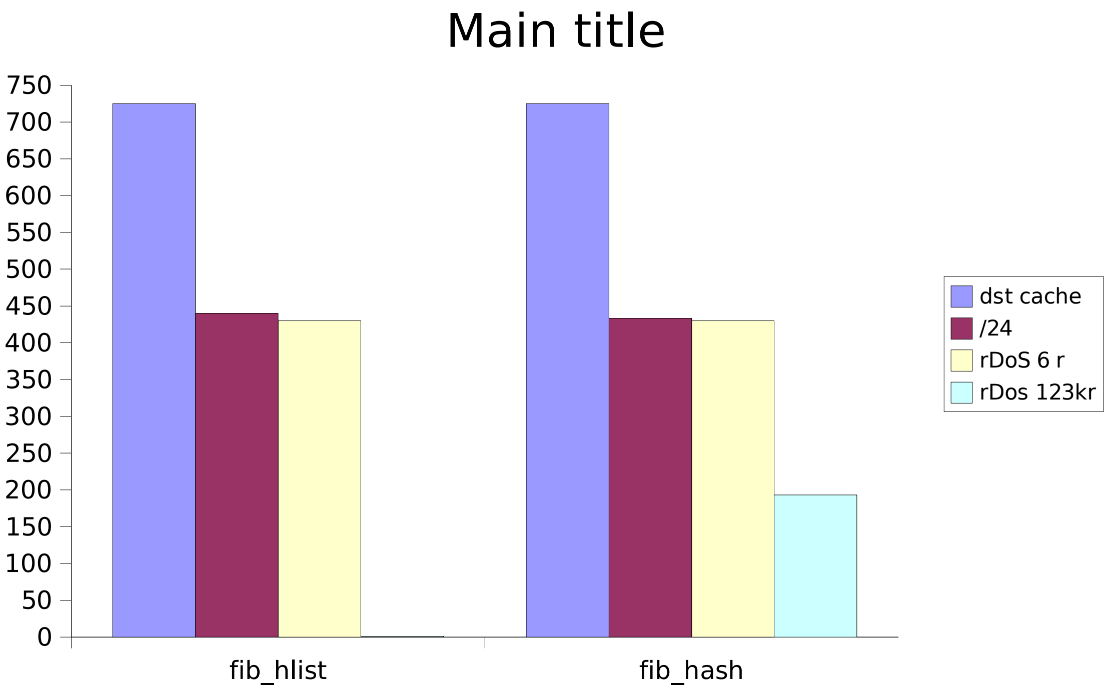
Ericsson is willing to open patent for Linux
Jamal have the contacs via Ericsson Montreal

DaveM has discussions with Washington university about who is willing to grant another patent for use with Linux

Discussed LC-trie with Alexey Kuznetsov.

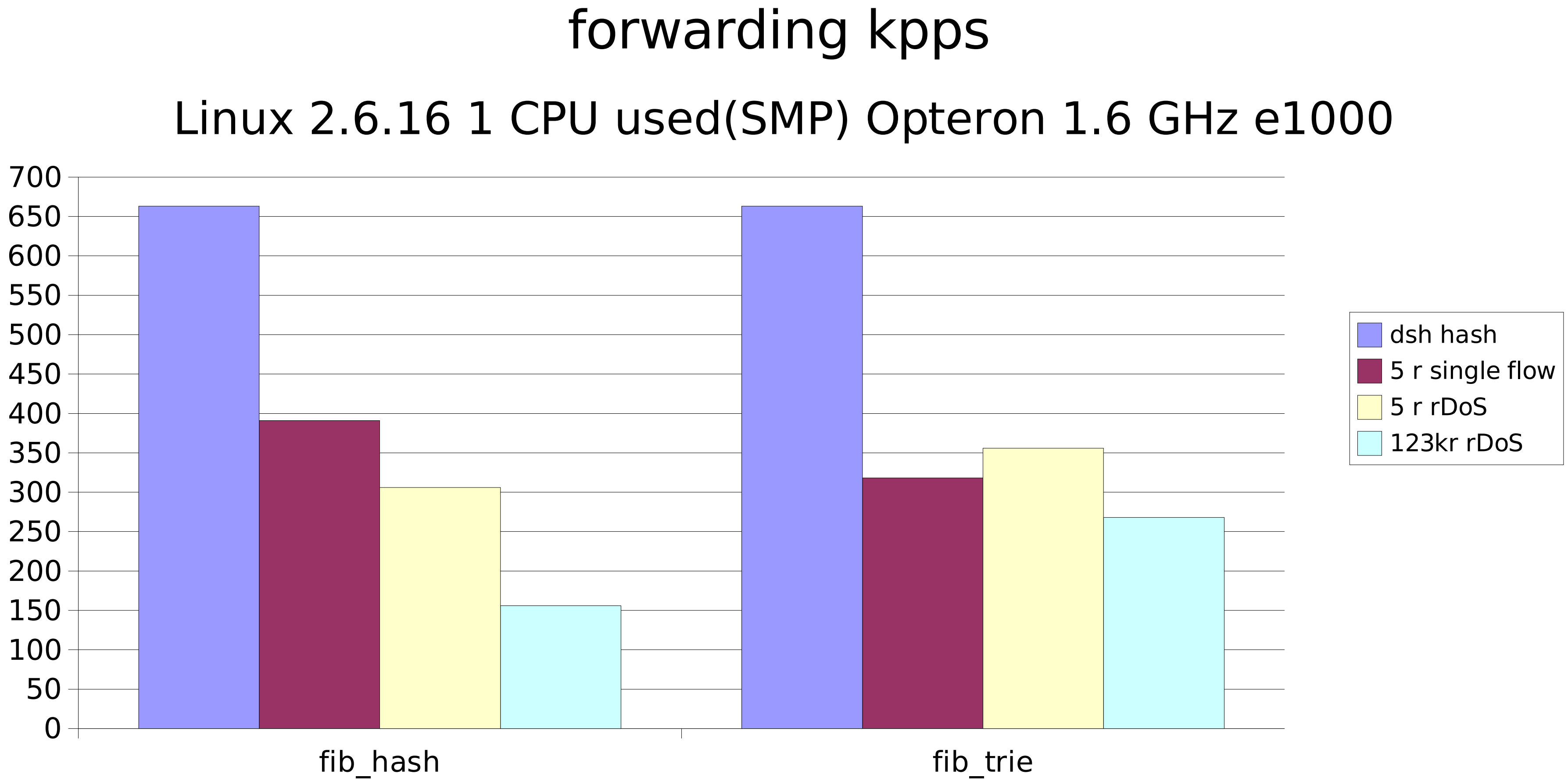
LC-trie investigations. Got GPL from authors.

fib_hlist performance



Note!
Zero for fib_hlist :) Still decent many apps.

fib_trie performance comparison

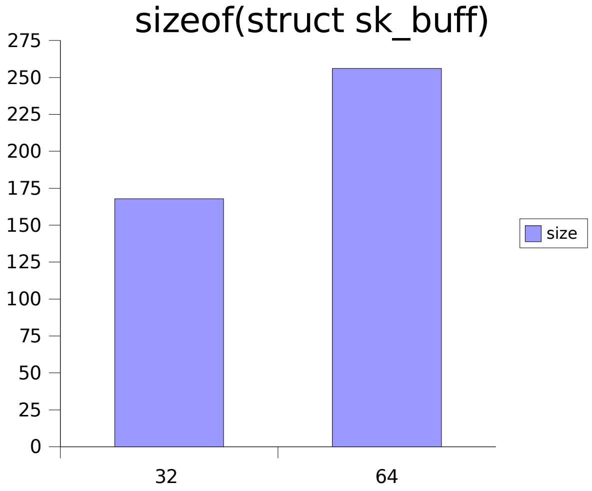


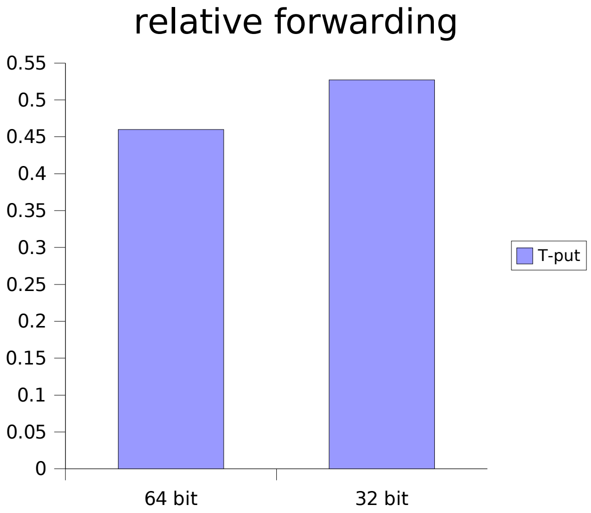
forwarding kpps

Linux 2.6.16 1 CPU used(SMP) Opteron 1.6 GHz e1000

Preroute pathes to disable route hash

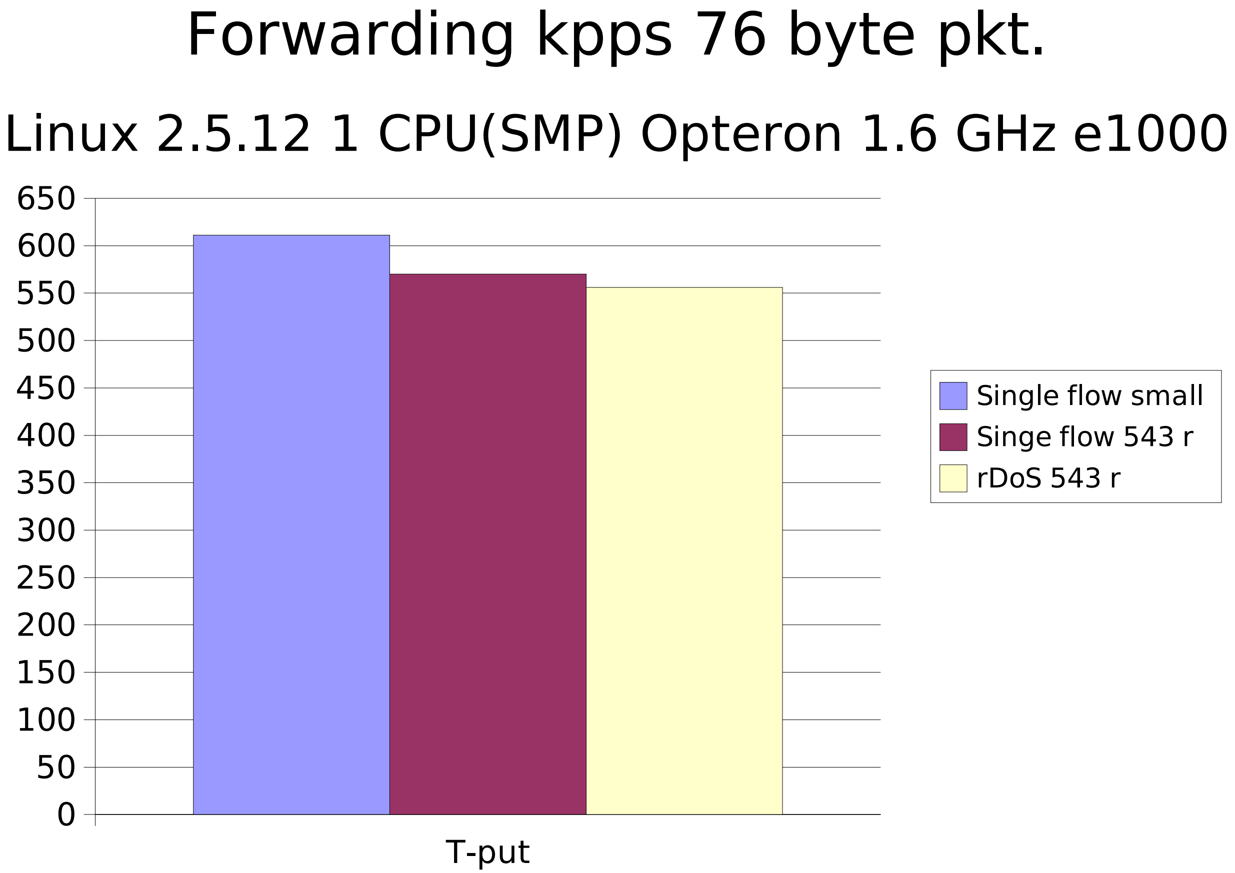
32/64 bit || sizeof(sk_buff)


sizeof(struct sk_buff)


relative forwarding

Gcc 3.4 x86_64 vs i686 on same HW

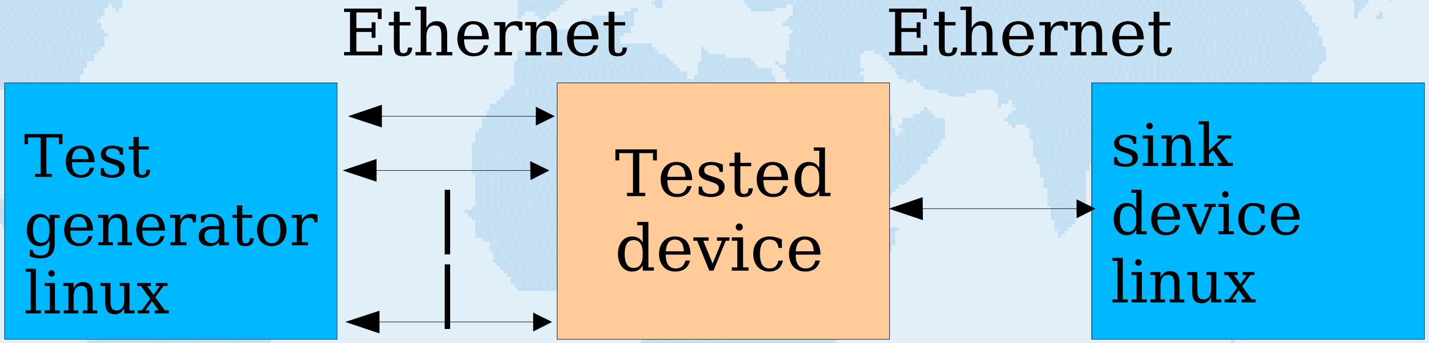
ipv6 performance



How rDoS work on sparse routing table?

Flexible netlab at Uppsala University

El cheapo-- High customable -- We write code :-)



* Raw packet performance
* TCP
* Timing
* Variants

Getting pktgen to run/1

Enable CONFIG_NET_PKTGEN
insmod pktgen if needed

One thread per CPU

- [pktgen/0],
- [pktgen/1]

/proc/net/pktgen/

- kpktgend_0,
- kpktgend_1,
- pgctrl

Getting pktgen to run/2

Adding devices to threads adds new files in /proc/

Example:

```
/proc/net/pktgen/
                  eth0
                  eth1
```

To be configured with device info

Getting pktgen to run/3

IP addresses, src, dst
counts
MAC adresses
Delay

Default:
UDP port 9 (discard) src and dst

Much more later....

Packet memory fastpath

Pktgen can do a trick to avoid full path for kmalloc and kfree when sending identical packets this increases performance. It's controlled by clone_skb

clone_skb=1000000 givs 1 master packet followed by one million clones

Results in only one full path malloc/kfree per million packets

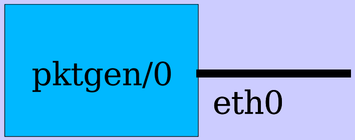
Delay

Gap between packets in nanoseconds.

Pktgen can insert an extra delay
For small delays pktgen busy-waits
Hard to get a specific rate
In most cases bursts are sent
Default 0

Needs experimentation

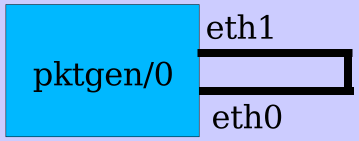
Setup Examples/1



Simple. Just send

Probably you need to keep link up

Setup Examples/2

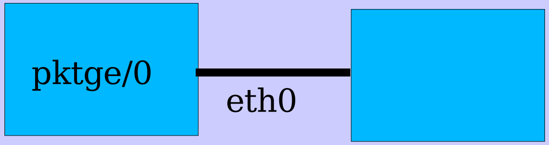


Just send.
Use another NIC on some box?

Set dst_mac correct if the pkts should be seen

Emulate incoming pkts with just a single box

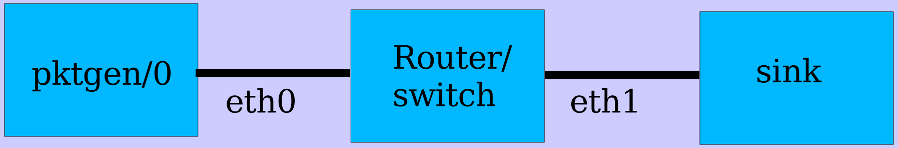
Setup Examples/3



Send to another device local or remote
Set dst_mac accordingly

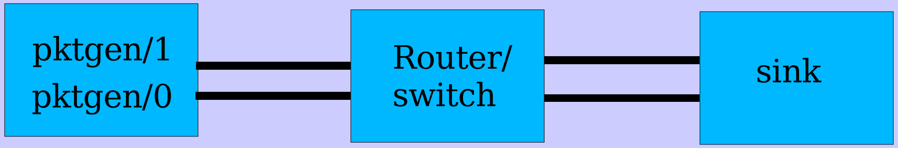
Dummy dev can be used to test forwarding

Setup Examples/4



Classical bridging/forwarding setup

Setup Examples/5



Bridging/forwarding in parallel setup
Can use multiple CPU's on sender(s)
(and on multiple CPU's onrouters)

Viewing pktgen threads

```
/proc/net/pktgen/kpktgend_0

Name: kpktgend_0
max_before_softirq: 10000
Running:
Stopped: eth1
Result: OK: max_before_softirq=10000
```

Configuring/1

Get a suitable script and modify

Next the glory details

Configuring/2

```
#! /bin/sh

#modprobe pktgen

function pgset() {
    local result

    echo $1 > $PGDEV

    result=`cat $PGDEV | fgrep "Result: OK:"`
    if [ "$result" = "" ]; then
         cat $PGDEV | fgrep Result:
    fi
}

function pg() {
    echo inject > $PGDEV
    cat $PGDEV
}
```

Configuring/3

```
# Config Start Here ---------------------------------------------------------

# thread config

PGDEV=/proc/net/pktgen/kpktgend_0
pgset "rem_device_all"
pgset "add_device eth1"

# device config

PGDEV=/proc/net/pktgen/eth1
pgset "count 1000000"
pgset "clone_skb 1000000"
pgset "pkt_size 60"
pgset "dst 10.10.11.2"
pgset "dst_mac  00:04:23:AE:05:16"

# Time to run
PGDEV=/proc/net/pktgen/pgctrl

echo "Running... ctrl^C to stop"
pgset "start"
echo "Done"

grep pps /proc/net/pktgen/eth1
```

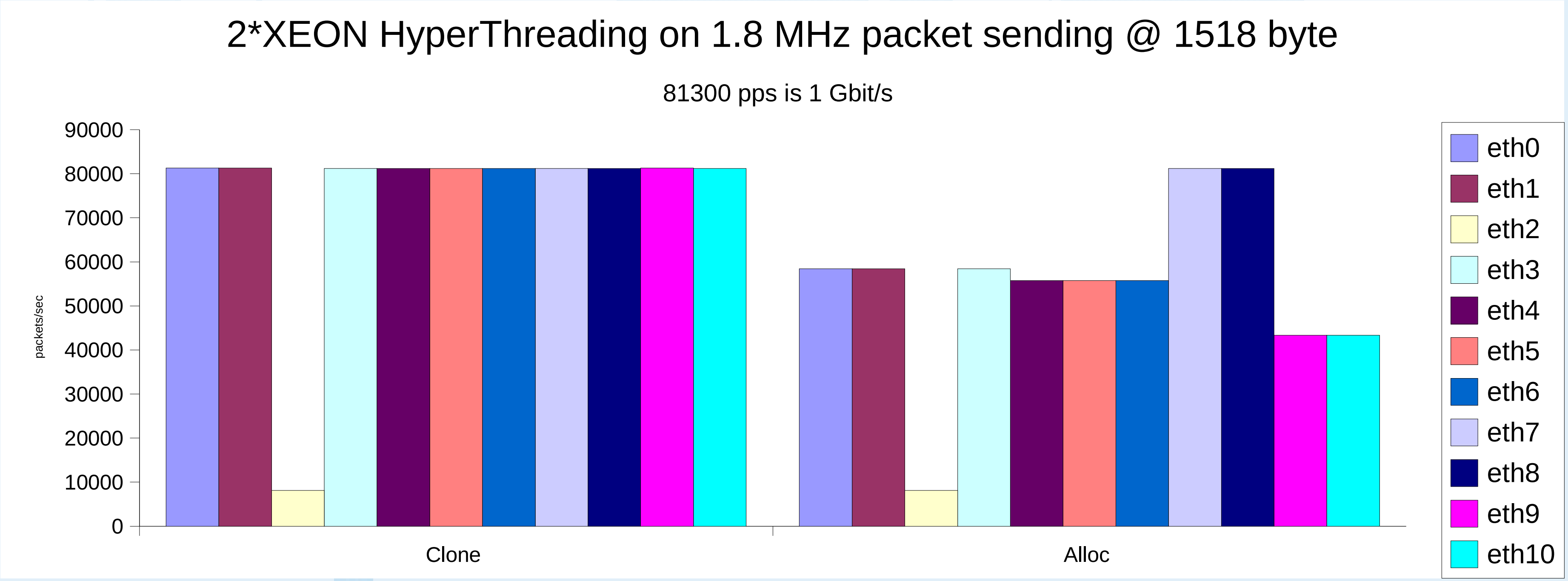
Viewing result

```
Cat /proc/net/pktgen/eth1

Params: count 10000000  min_pkt_size: 60  max_pkt_size: 60
     frags: 0  delay: 0  clone_skb: 1000000  ifname: eth1
     flows: 0 flowlen: 0
     dst_min: 10.10.11.2  dst_max:
     src_min:   src_max:
     src_mac: 00:04:23:AC:FD:82  dst_mac: 00:04:23:AE:05:16
     udp_src_min: 9  udp_src_max: 9  udp_dst_min: 9  udp_dst_max: 9
     src_mac_count: 0  dst_mac_count: 0
     Flags:
Current:
     pkts-sofar: 10000000  errors: 0
     started: 1119356264434801us  stopped: 1119356275792478us idle:
1434226us
     seq_num: 10000011  cur_dst_mac_offset: 0  cur_src_mac_offset: 0
     cur_saddr: 0x10a0a0a  cur_daddr: 0x20b0a0a
     cur_udp_dst: 9  cur_udp_src: 9
     flows: 0
Result: OK: 11357677(c9923451+d1434226) usec, 10000000 (60byte,0frags)
  880461pps 422Mb/sec (422621280bps) errors: 0
```

Some GIGE experiments

Pktgen sending test w. 11 GIGE interfaces
skb clone = 10.0 Gbit/s
skb alloc = 7.4 Gbit/s



SeverWorks X5DL8-GG Intel e1000

Trash datastucture

Interesting novel approc. Trie-Hash --> Trash

When extending the LC-trie

Paper with Stefan Nilsson/KTH

Expoits that keylen does not affect tree deepth

We lengthen the so key it can be better compressed.

Implemented in Linux forwarding patch as a replacement to the route hash.

Trash datastucture

Can do full key lookup. src/dst/sport/dport/proto/if etc and later socket.

For even ip6 with littele performnace degradation

Could be a candidate for the grand unified lookup

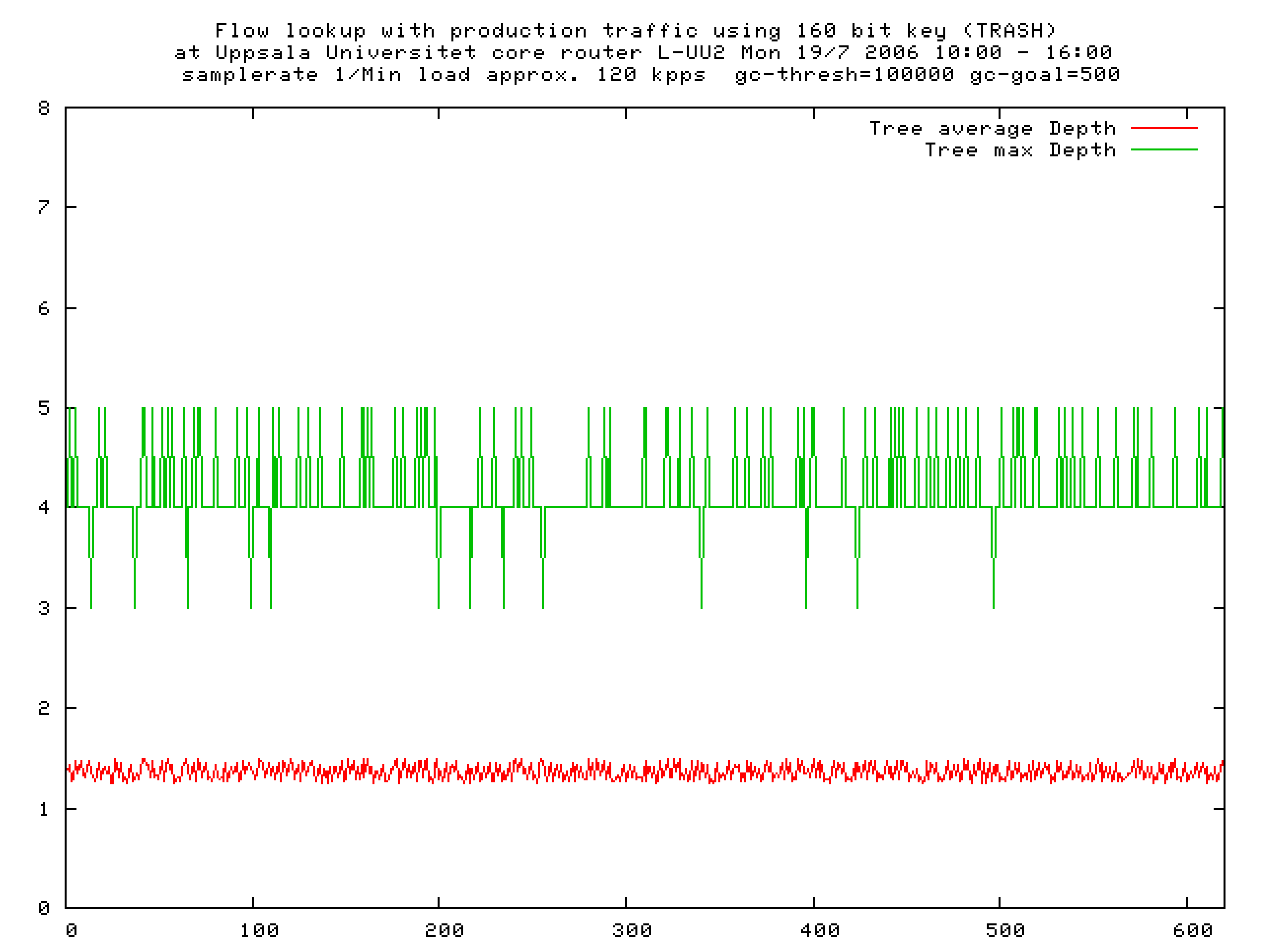
Full flow lookup can understand connections.

Free flow logging etc

New garbage collection (GC) possible. Active GC stat-ed
AGC in the paper. Listen to TCP SYN, FIN and RST
Show to be performance winner.

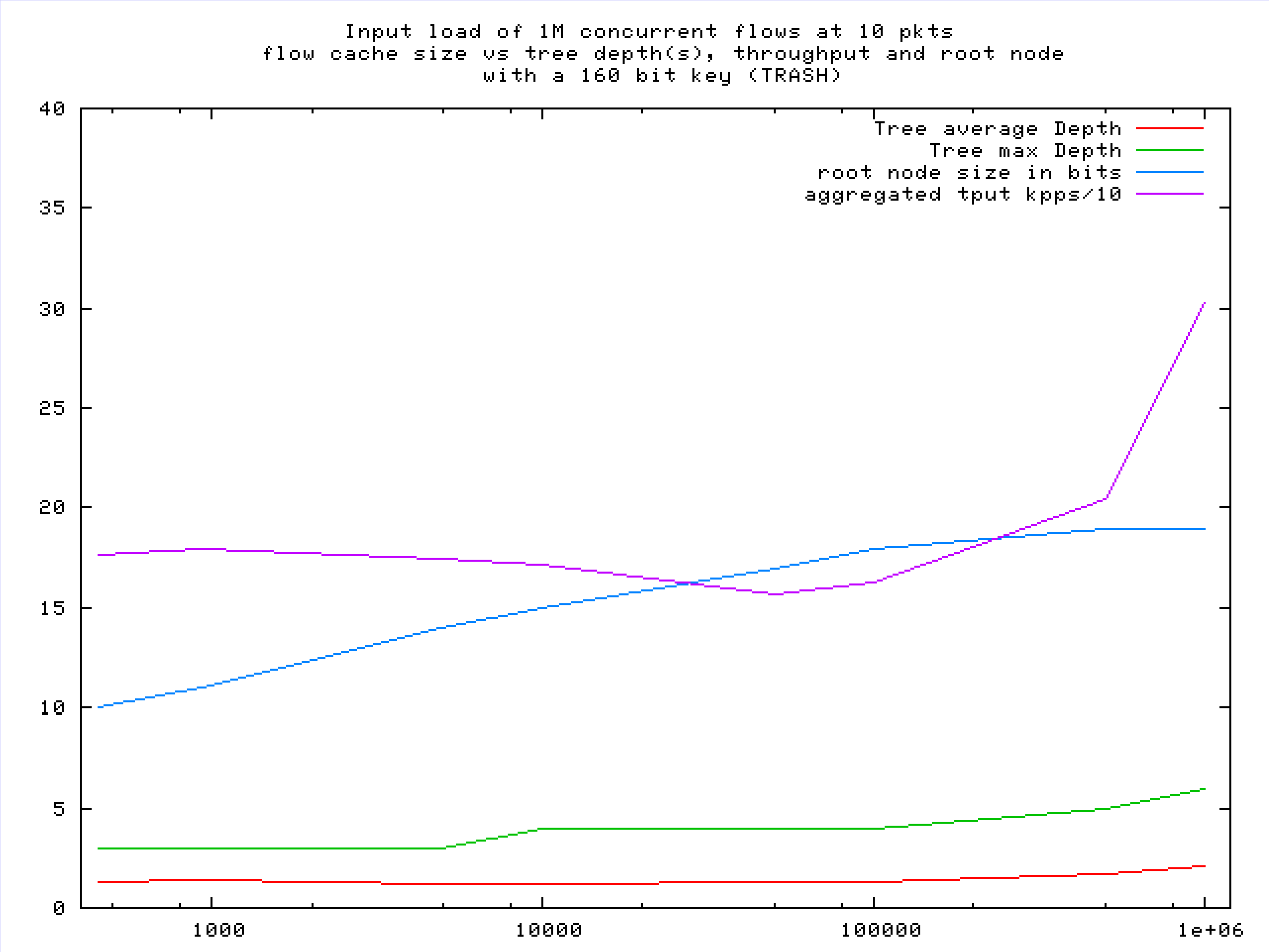
Trash datastucture

Uppsala Universitet core router



Trash datastucture

Very flat(fast) trees



Trash datastucture

Very flat(fast) trees

Paper written to avoid patents

Paper was accepted for IEEE 2007
routing & switching in New York

Linux implementation by author.

OpenWrt

Very nice platform for various applications. Enjoy! www.openwrt.org

Enormues amount of supported SW already included.

Apps, WLAN, router, gateway, servers

Many architectures. MIPS/x86 etc linksys ASUS etc etc,

Used by freifunk (adhoc), whiterussian.

OpenWrt

Nice ready crosscompile platform

SVN subversion build.

Menubased config

Linux kernel 2.4/2.6

Builds ready-to-use firmware

OpenWrt/logger project

Needed a cheap flexible hi-perf logger

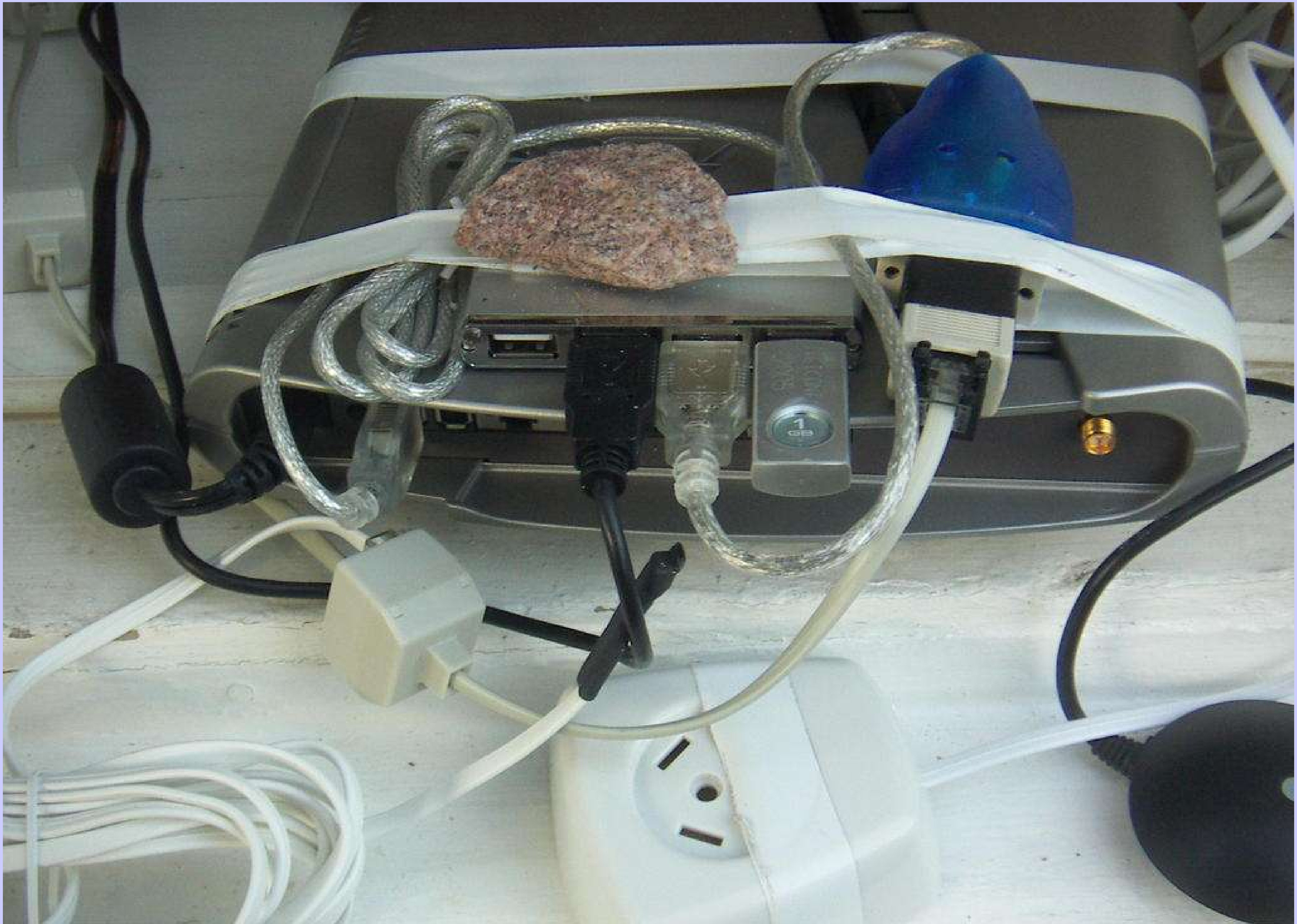
gps used with NTP for accurat time

Dallas 1-wire measurement bus.

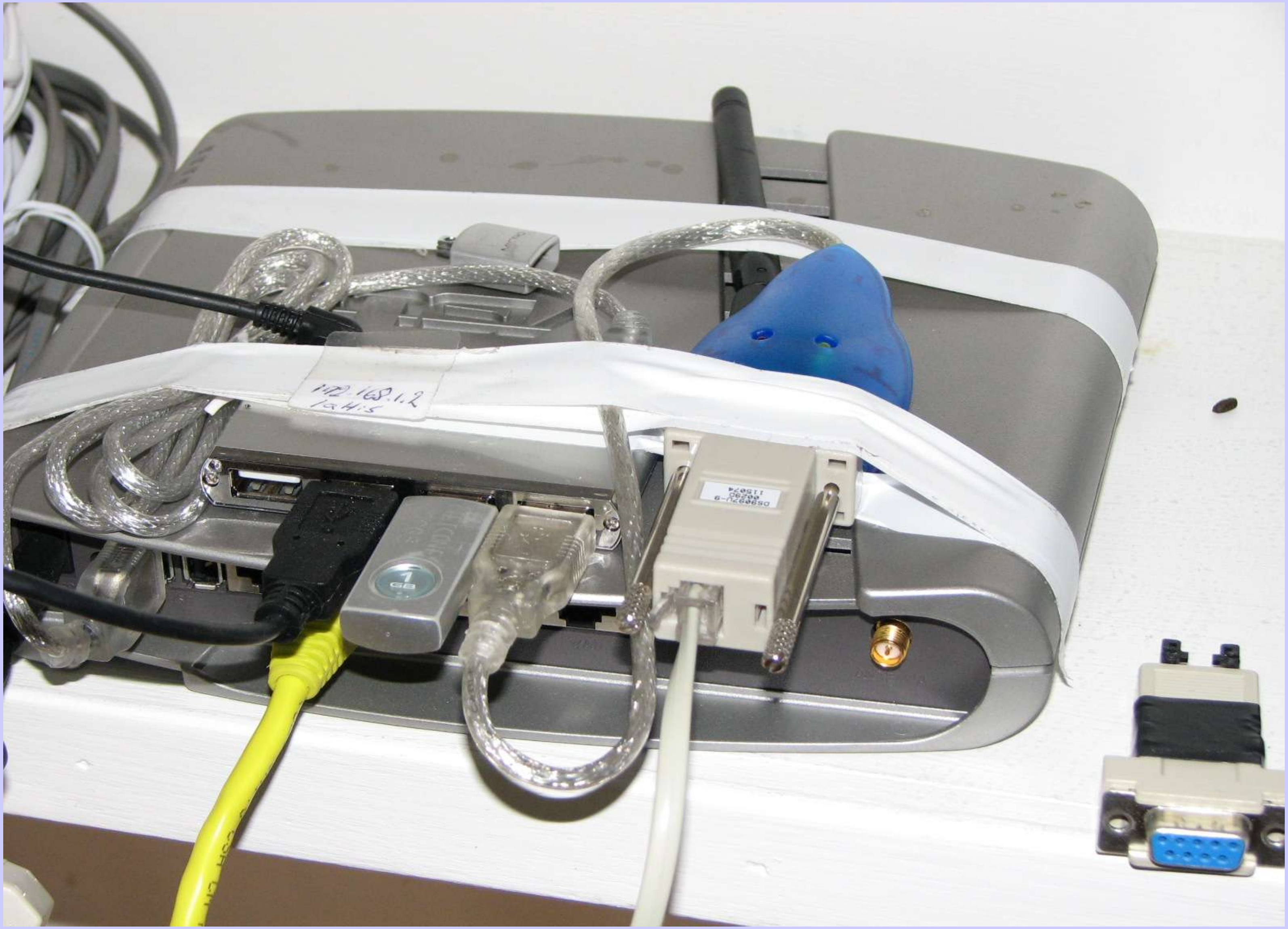
See www.digitemp.com

Need lots of storage

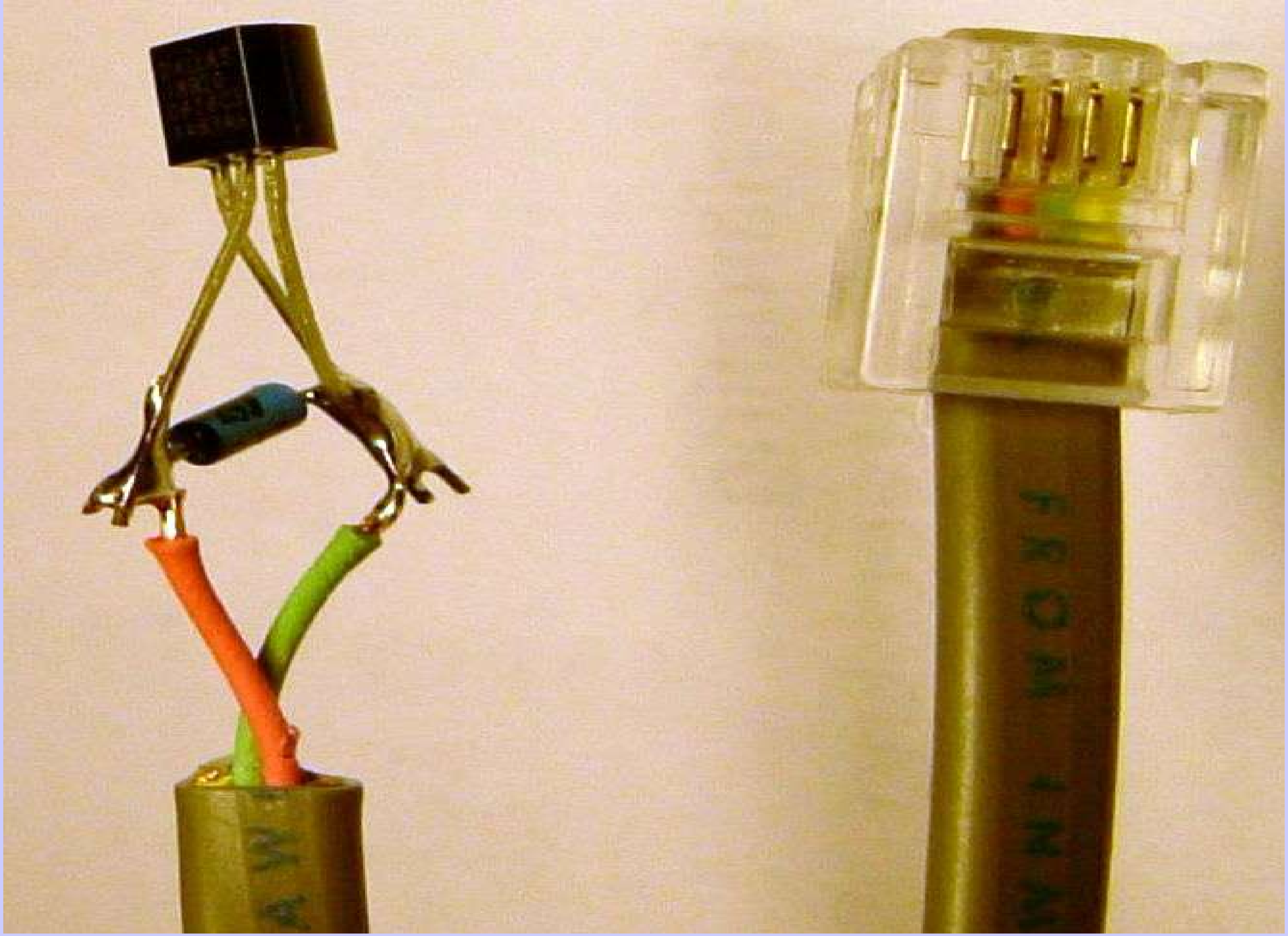
OpenWrt/logger project



OpenWrt/logger project

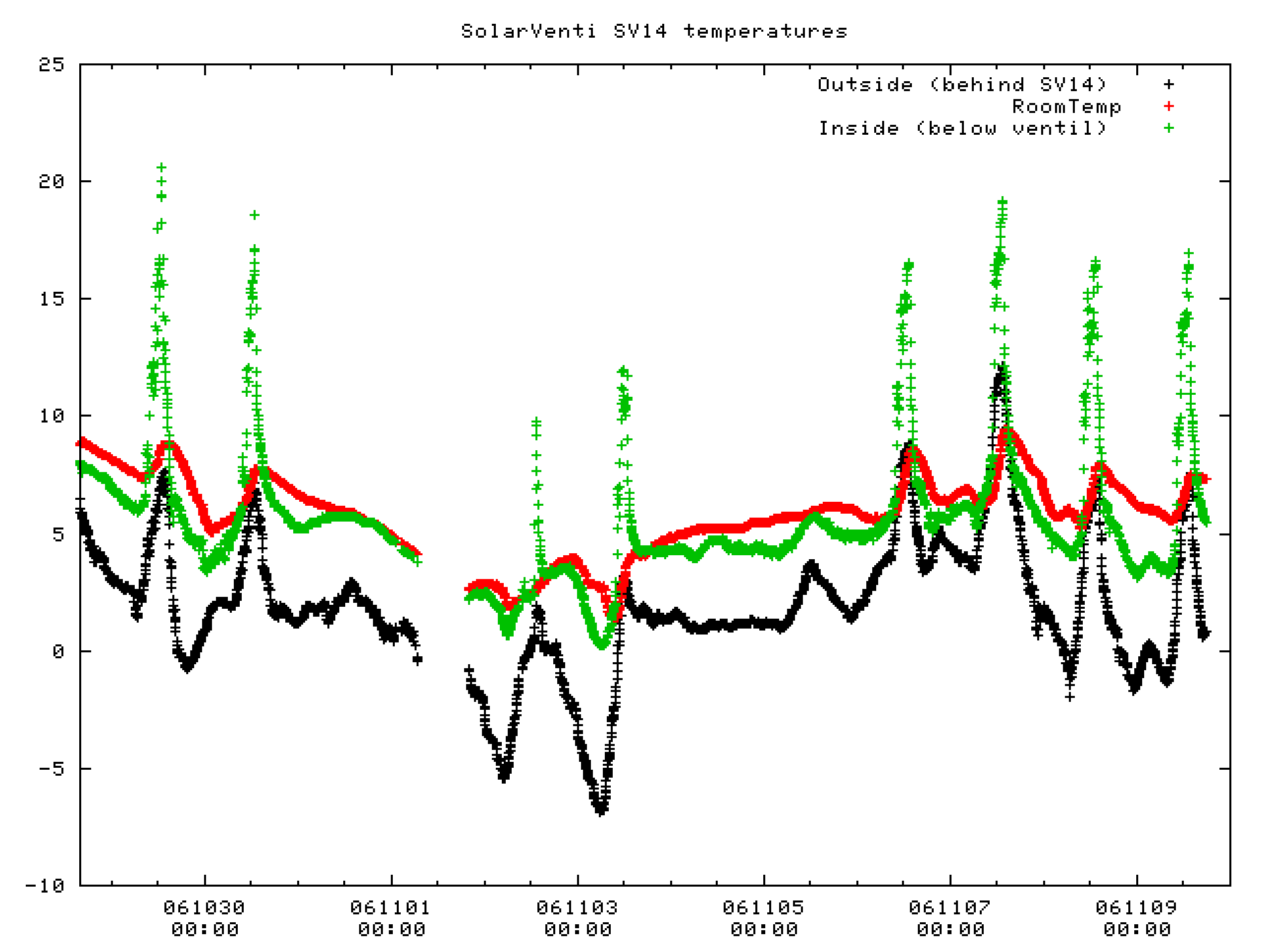


OpenWrt/logger project



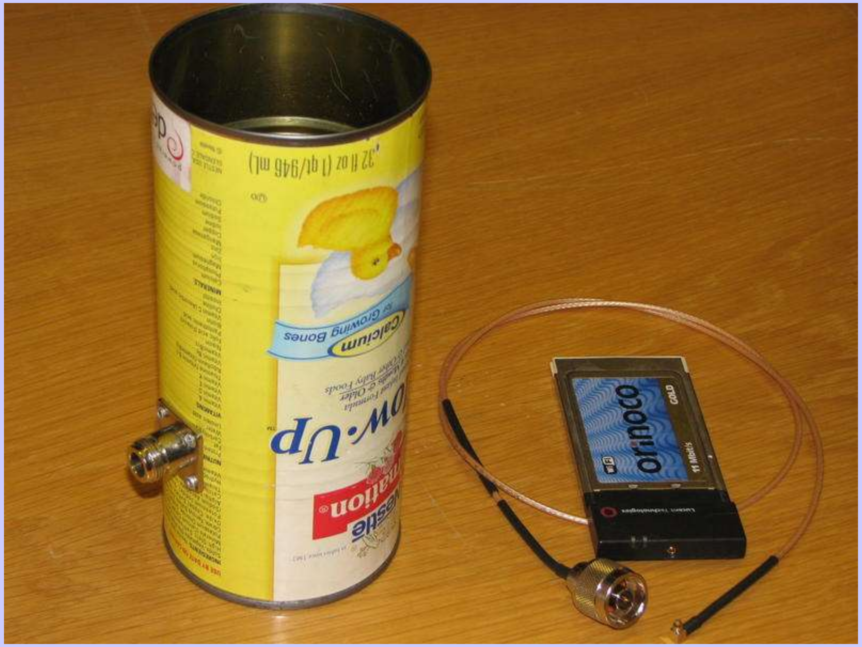
OpenWrt/logger project

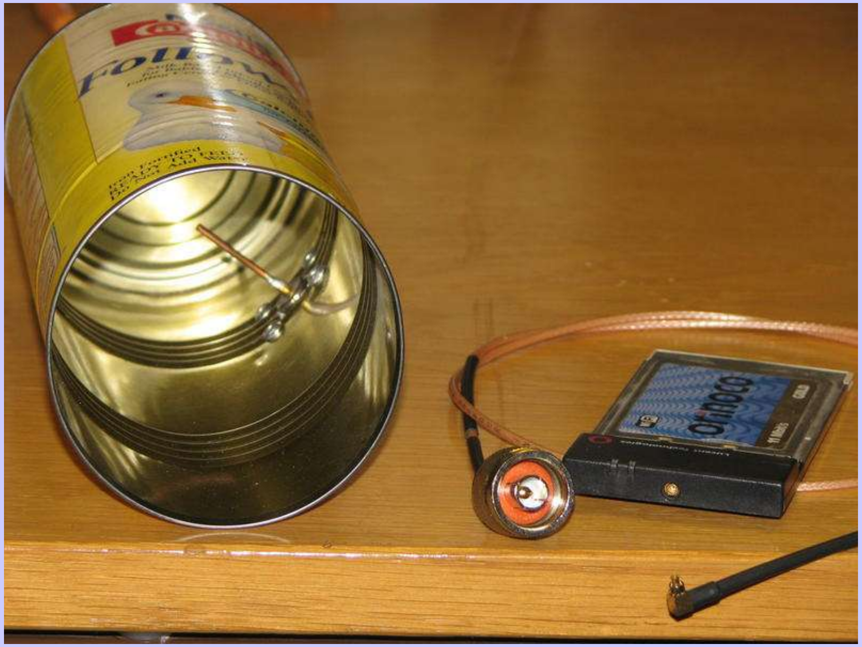
OpenWrt/logger project



Netgear med USB

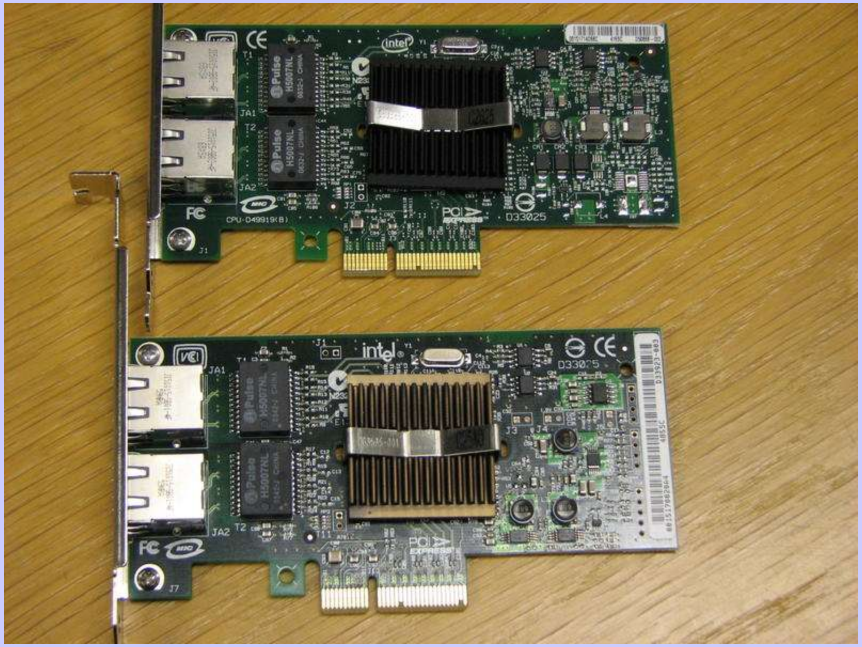
CANTENNA

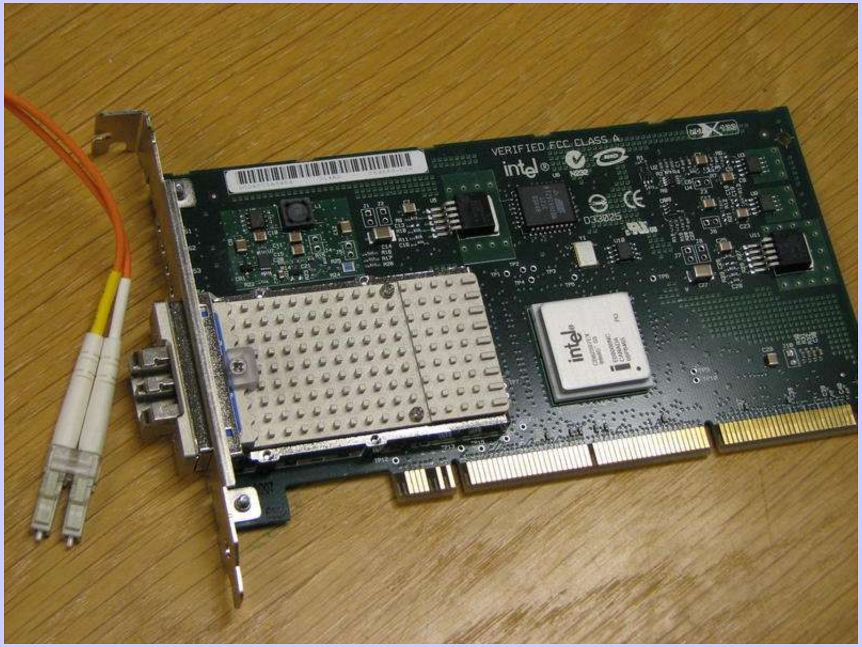


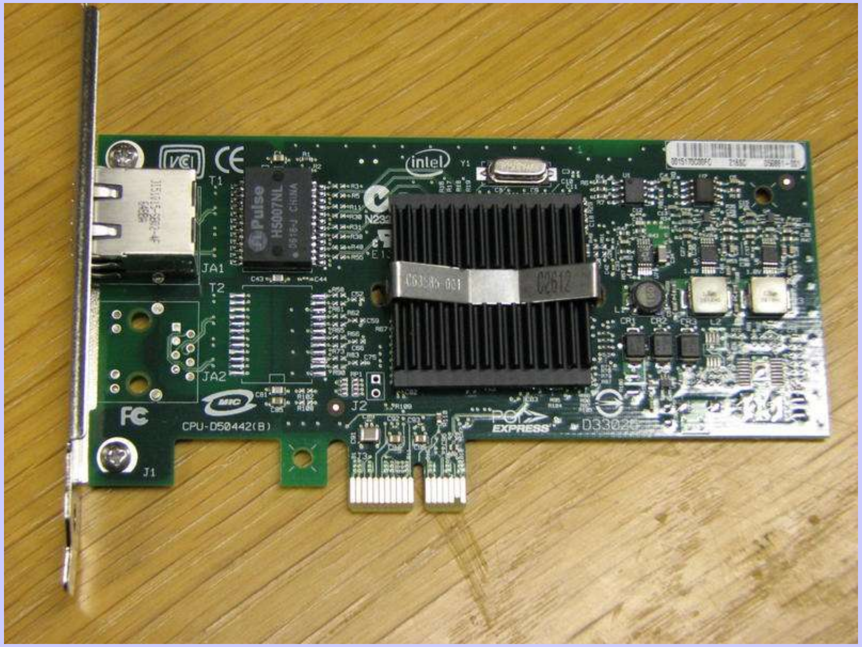


Labbet

Intel NIC's


CPU-D49919(B)
D33025
D33025


VERIFIED FCC CLASS A
D33025


CPU-D50442(B)
D33025

Nextgen NIC's

Multiple RX and TX queues

This together with MSI-X interrupts
And HW classifiers on NIC's

A breakthrough.....

Nextgen NIC's

Input path

1) HW classifier can direct traffic/network load to a selected RX ring.

2) RX ring has an assigned irq to a CPU-core via MSI-X

This way network load can get distributed along many CPU's

Nextgen NIC's

Input path cont.

Linux OS has work for a very long time with architectures and locking issues for multiCPU.

API and tool for controlling HW classifiers is needed most likely ethtool.

Very exciting....

Nextgen NIC's

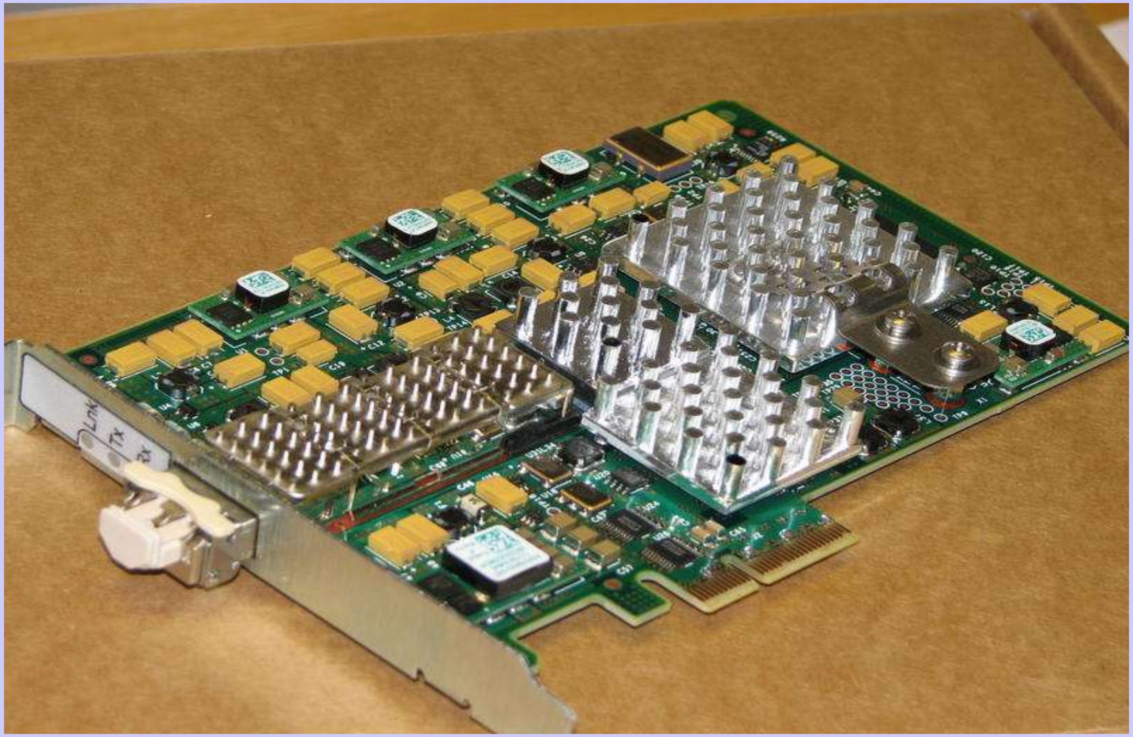
Output path

Device drivers can send to a selected TX-ring

1) to prioritize traffic
2) to avoid cache bouncing

As a prosal now a field queue_mapping is added to skb struct. So network stack can hint device driver about queue usage.

s2io



PCIe-x4

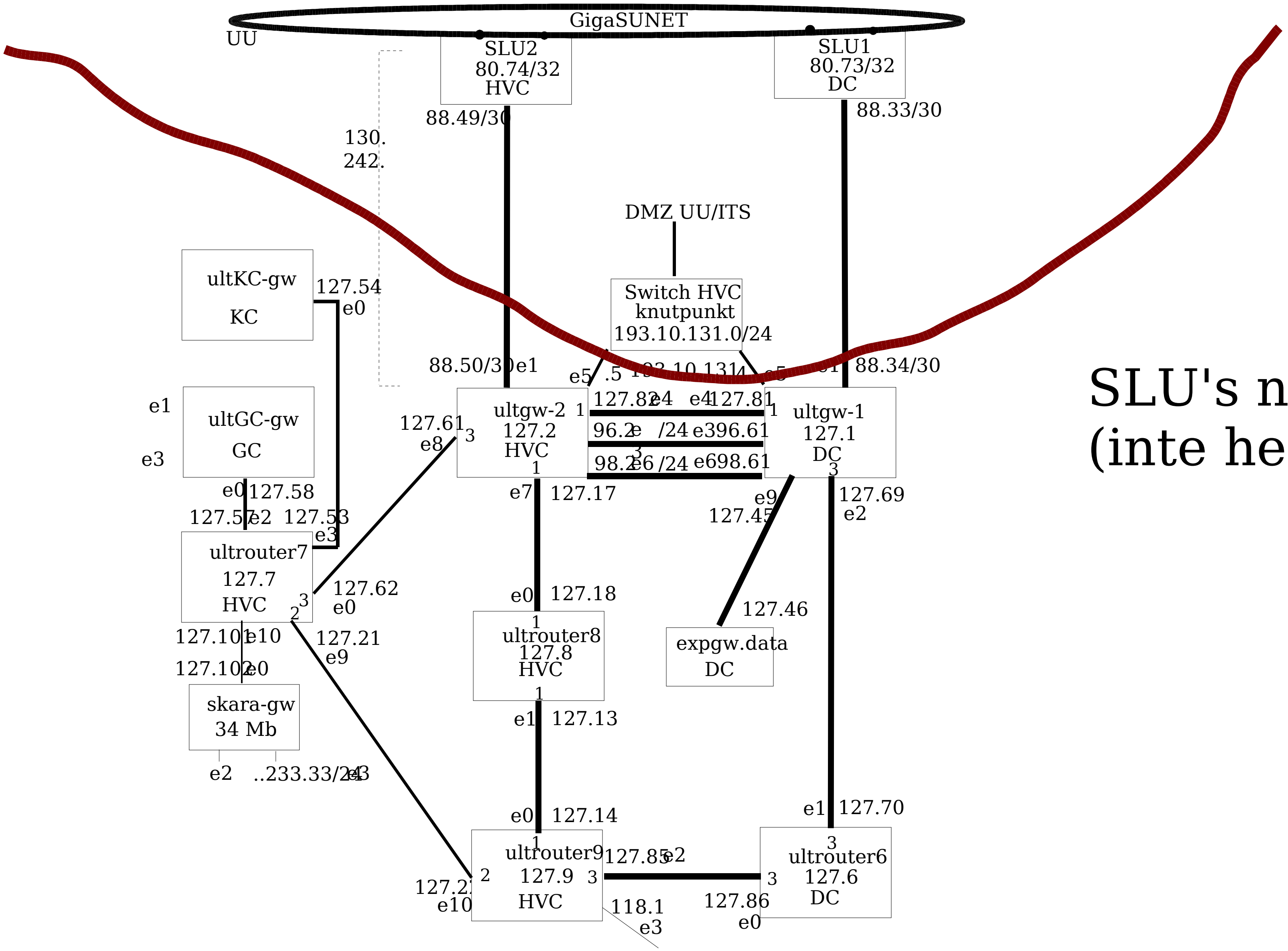
8 RX queues

8 TX queues

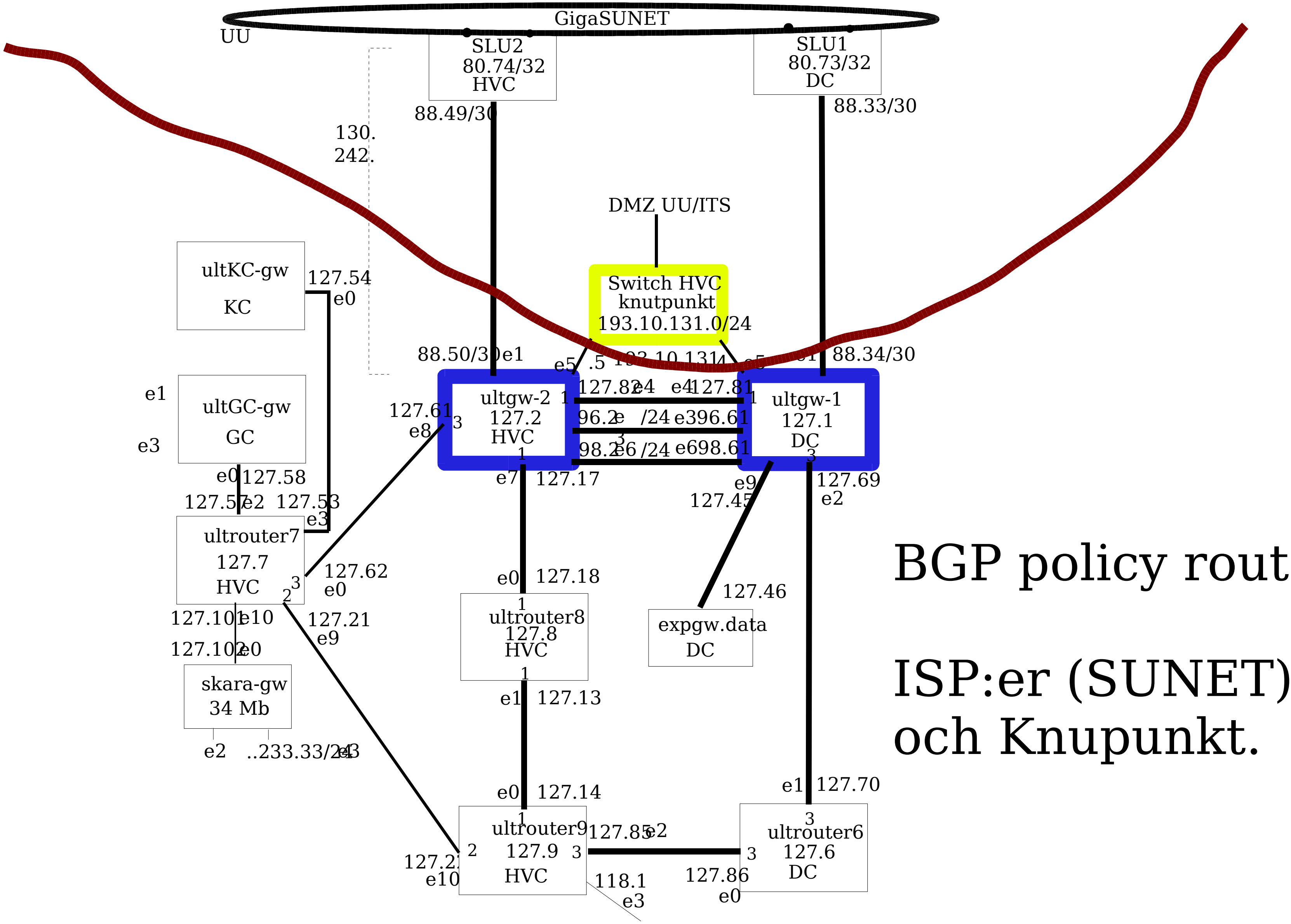
64 inq's

Various offload

Många voro kallade...

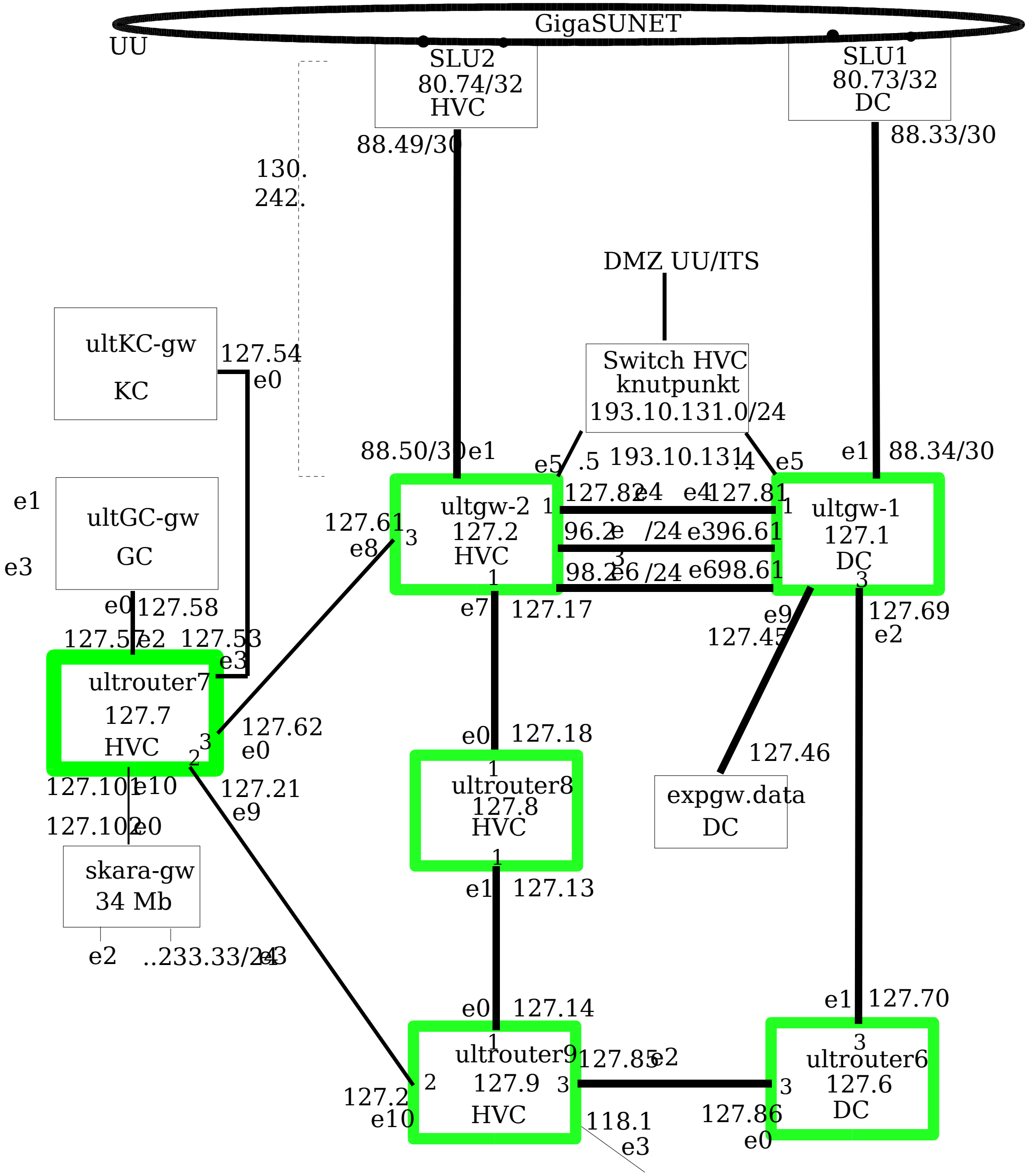


SLU's nät
(inte hela)

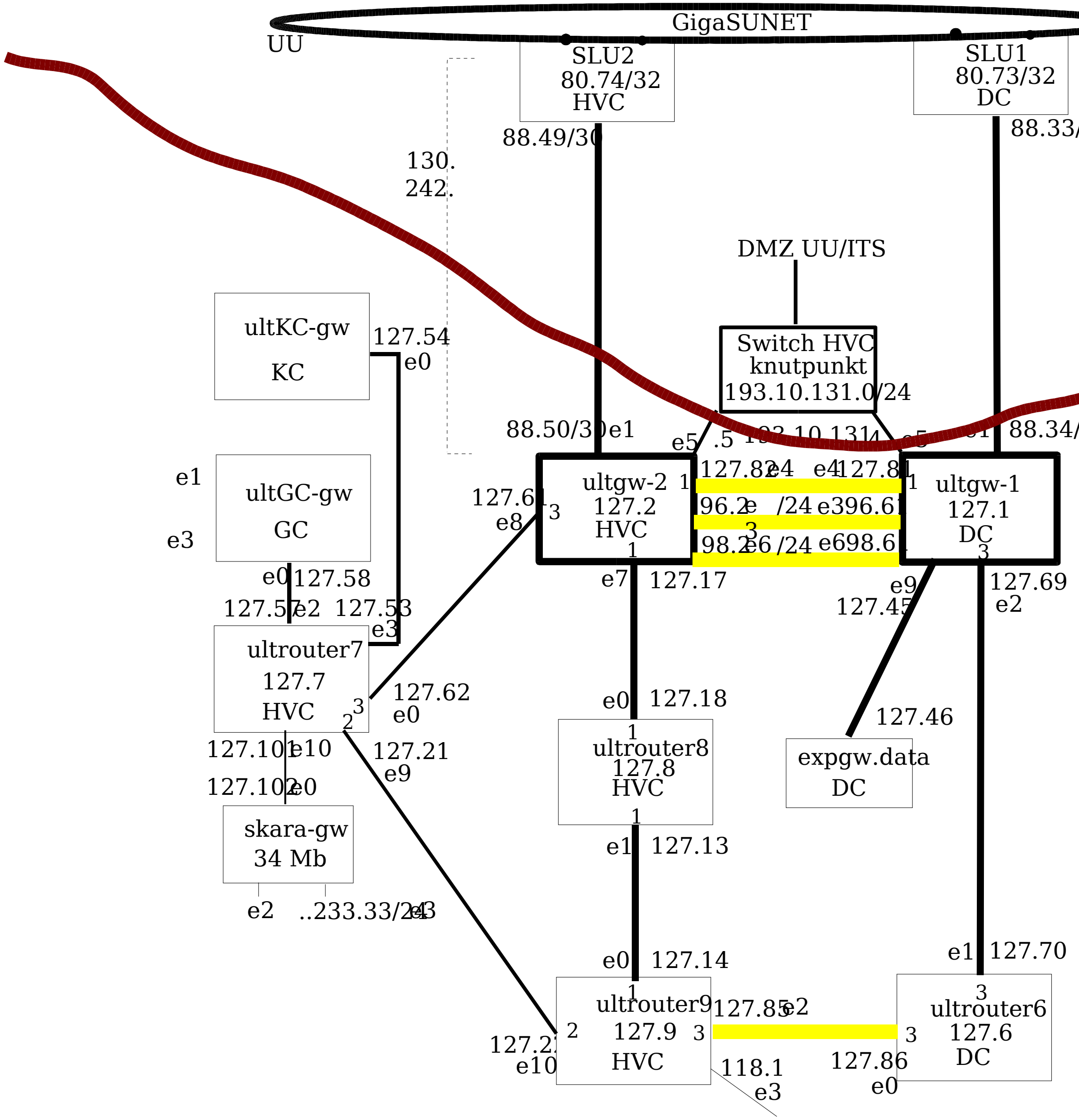


BGP policy routing

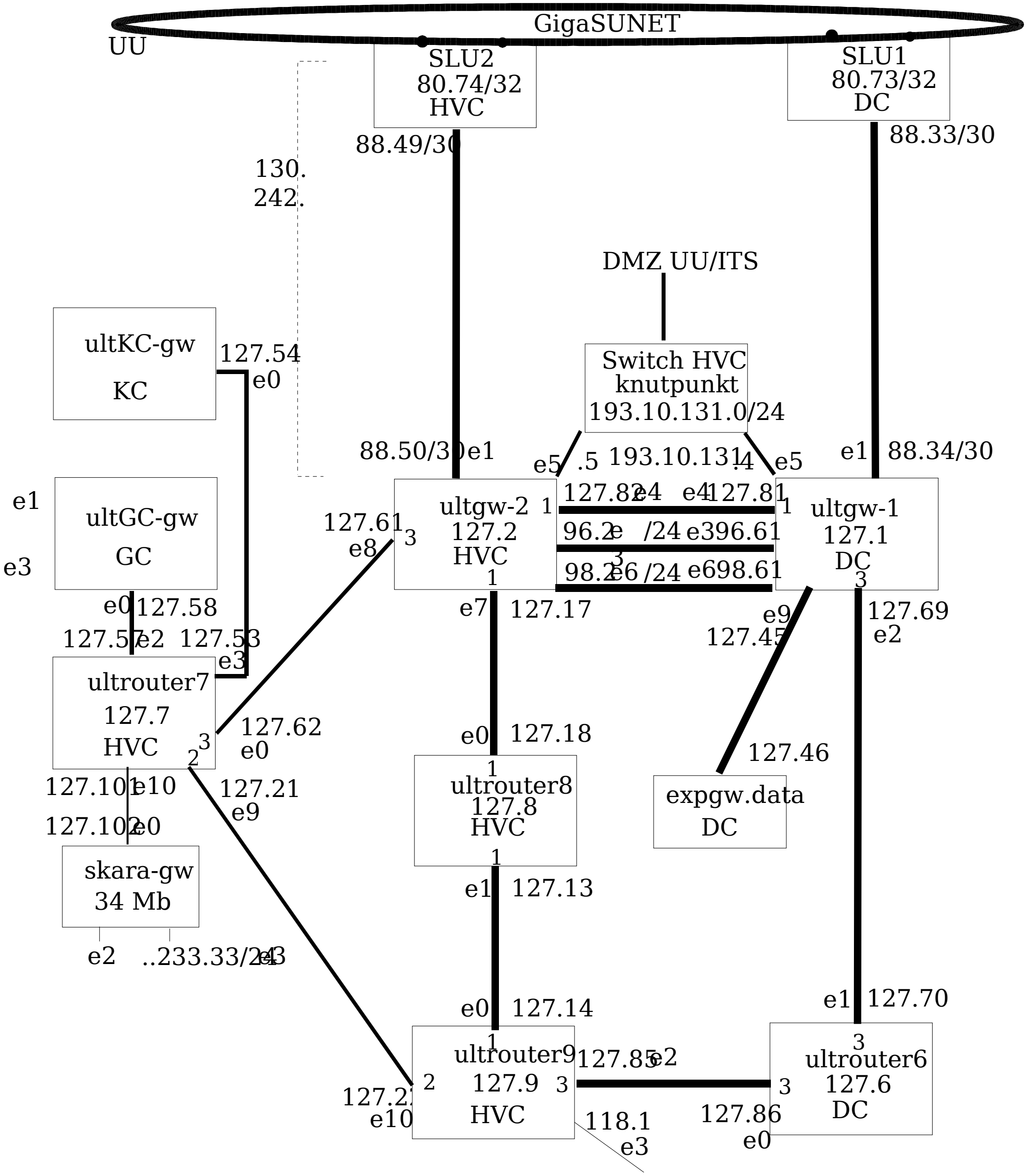
ISP:er (SUNET) och Knupunkt.



Redundant inre kärna



Redundant ansluting av tunga servernät via router discovery


GigaSUNET
UU
SLU2
80.74/32
HVC
SLU1
80.73/32
DC
88.49/30
88.33/30
130.
242.
DMZ UU/ITS
Switch HVC
knutpunkt
193.10.131.0/24
ultKC-gw
KC
127.54
e0
e1
ultGC-gw
GC
e3
88.50/30
e1
e5
.5
193.10.131
.4
e5
e1
88.34/30
ultgw-2
127.2
HVC
1
3
1
127.82
e4
e4
127.81
1
96.2
e
/24
e3
96.61
3
98.2
e6
/24
e6
98.61
ultgw-1
127.1
DC
3
127.61
e8
e0
127.58
127.57
e2
127.53
e3
e7
127.17
e9
127.45
127.69
e2
ultrouter7
127.7
HVC
3
2
127.62
e0
e0
127.18
1
ultrouter8
127.8
HVC
1
127.46
expgw.data
DC
127.101
e10
127.21
e9
127.102
e0
skara-gw
34 Mb
e1
127.13
e2
..233.33/24
e3
e0
127.14
e1
127.70
1
ultrouter9
127.9
HVC
2
3
127.85
e2
3
ultrouter6
127.6
DC
3
127.22
e10
118.1
e3
127.86
e0

Nästan uteslutande...

- OpenSource implementation
- bifrost
- zebra

A new network symbol has been seen...

The Penguin Has Landed

ifstat2

- output errors etc ncurses?

rtstat

- output, how, mask groups?

Oprofile statisk länkat paket

dok

logo

bifrost-usb-boot-HOWTO

bifrost-grub-HOTO

unified lookup/connection tracking

test & hipac-HOWTO

Hi-perf PCI-E TYAN 2915? Test

pktgen drop ingress qdisc

Bosh LH-jetronic
flexitune (www.flexitune.se)
flextec

Hacking car fuel system....

GPS breakthrough

Other Linux hacking

Ready for serious work?

Enhanced radix tree (LEF) better then LC-trie?

Userland code with full BGP table indicate
LEF is 3 times faster than LC-trie. Full sensation!

ftp://robur.slu.se:21/pub/Linux/tmp/radix_test.tgz

What is going on? Data structures are bigger and
we accessing more nodes?

Kernel test not yes done.. but userland results
are mysterious Any qualified investigators???