February 2003 FIRST Technical Colloquium
February 10-11, 2003
@
Uppsala, Sweden

bifrost a high performance router & firewall

Robert Olsson
Hans Wassen

Bifrost concept

- Small size Linux distribution targeted for Flashdisks 20 MB
- Optimized for networking/firewalling
- Tested with selected drivers and hardware
- Open platform for development and collaboration
- Results and experiences shared

Bifrost concept

- Linux kernel collaboration
  - FASTROUTE, HW_FLOCONTROL, 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)

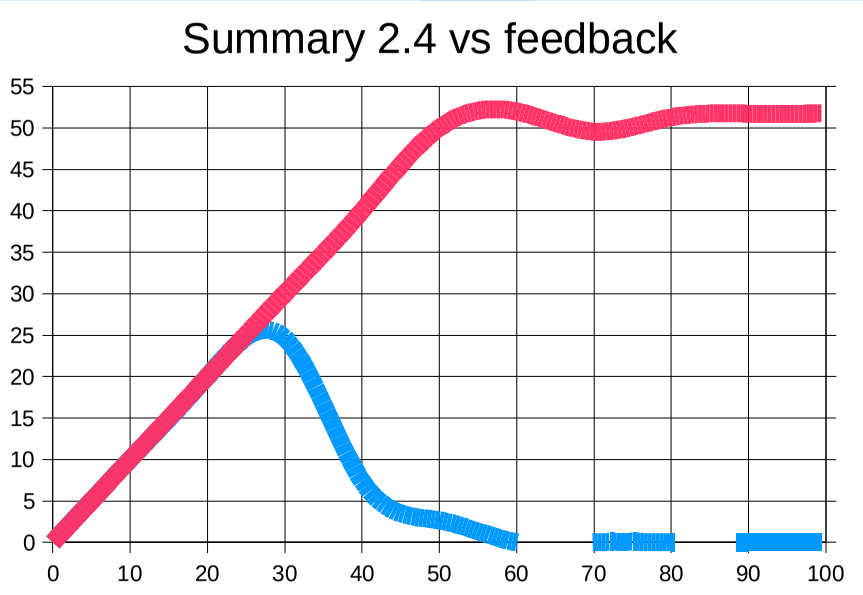
Collaboration/development
The New API

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

Overall Effect

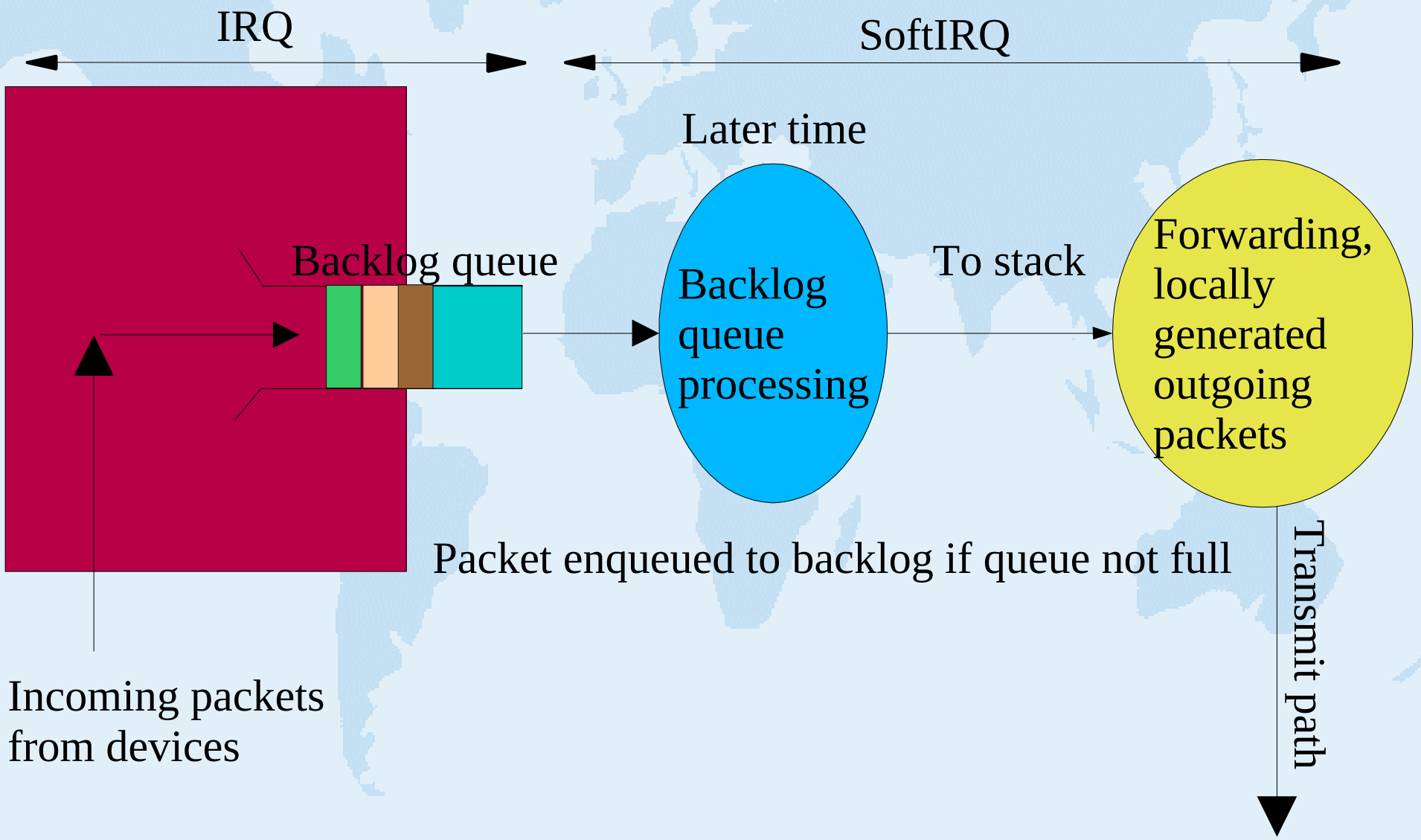
- Inelegant handling of heavy net loads
  - System collapse
- Scalability affected
  - System and number of NICS
    - A single hogger netdev can bring the system to its knees and deny service to others



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

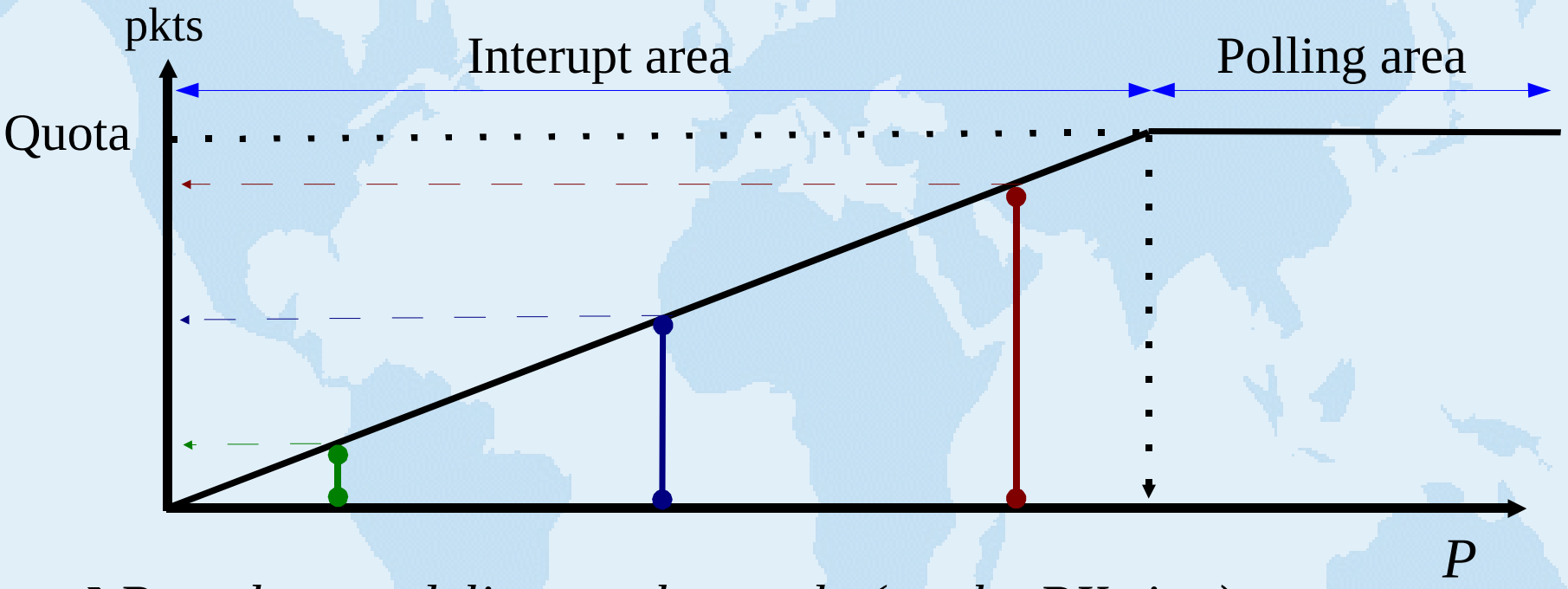
Looking inside the box



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

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 input rates
➔Area under curve shows how many packets before next interrupt
➔*Quota* enforces fair share

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.

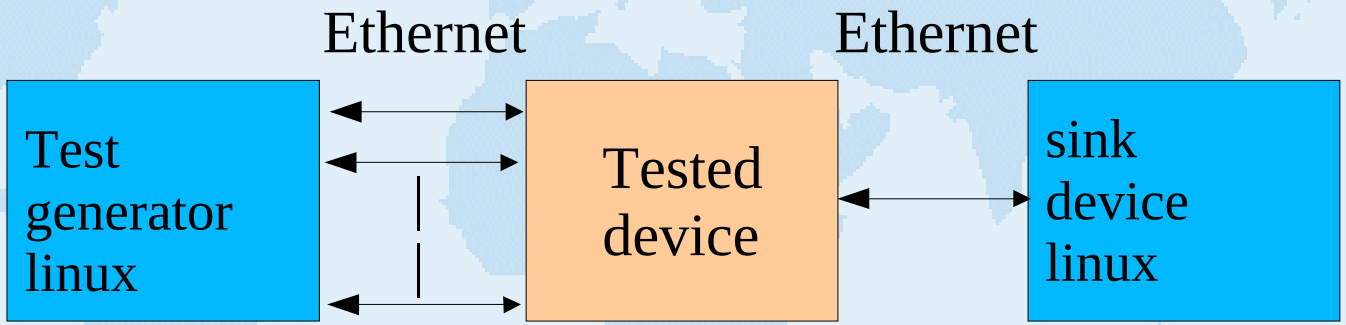
Old scheme added interrupt delay to handle
CPU from being killed by interrupts.

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?

Flexible netlab at Uppsala University

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



* Raw packet performance
* TCP
* Timing
* Variants

Hardware for high perf. Networking

Motherboard

CPU	Uni or multi-processor
Chipset	BX, ServerWorks, E750X
BUS/PCI-design	# PCI-BUS'es @ 133MHz
Interrupt design	PIC, IO-APIC etc

Standby Power (Wake on Lan) can be a problem with many NIC's

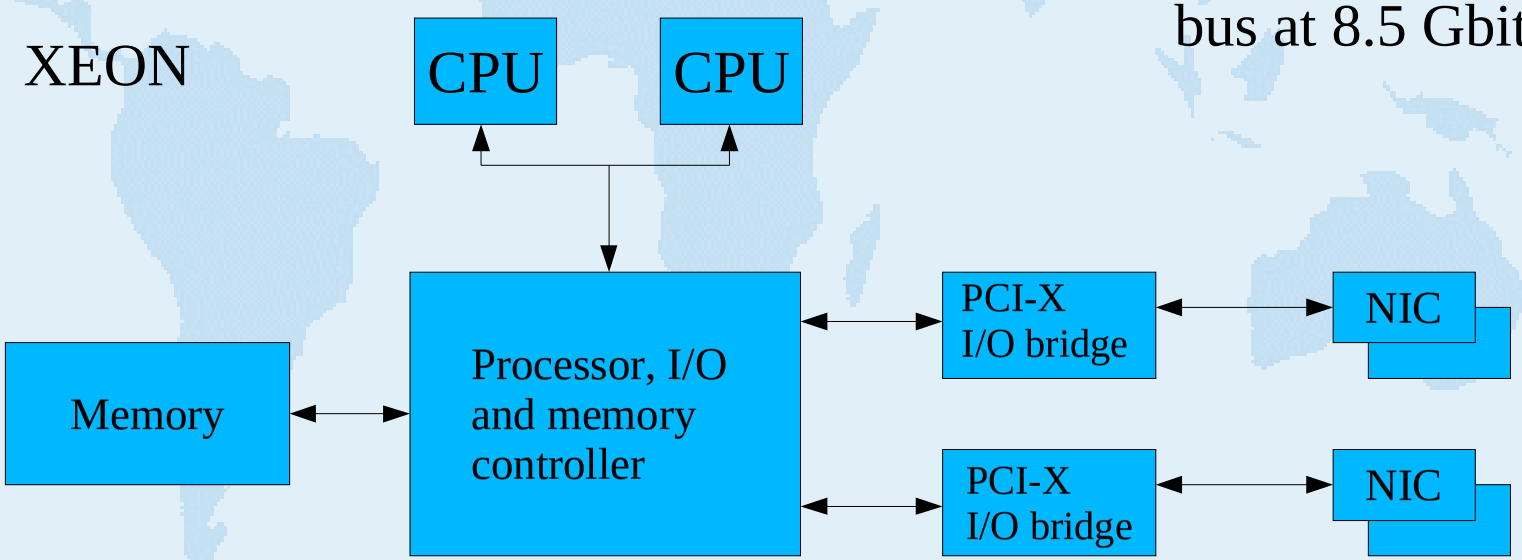
Hardware for high perf. Networking

ServerWorks,
Intel E750X
chipset
many PCI-X hubs/bridges

And dual XEON

Many vendors
use Compact
PCI already

PCI-X is here
bus at 8.5 Gbit/s



Hardware for high perf. Networking

Currently Intel has advantage. Broadcom can be a dark horse. All has NAPI drivers.

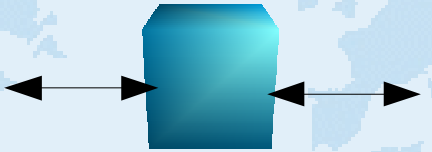
GIGE chipsets available for PCI

e1000	Intel -- e1000
BCM5700	Broadcom – tg3
dl-2k	D-Link -- dl2k

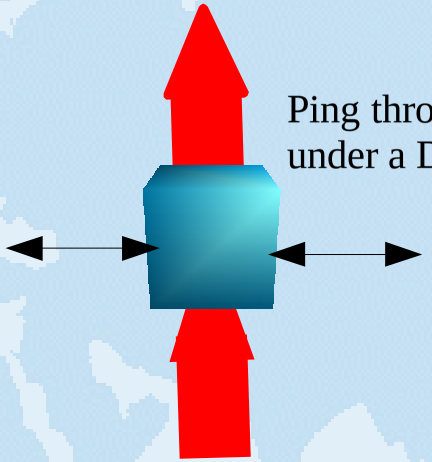
Some board manufactors switch chipset often. Chip documentation a problem.

Some GIGE experiments/NAPI

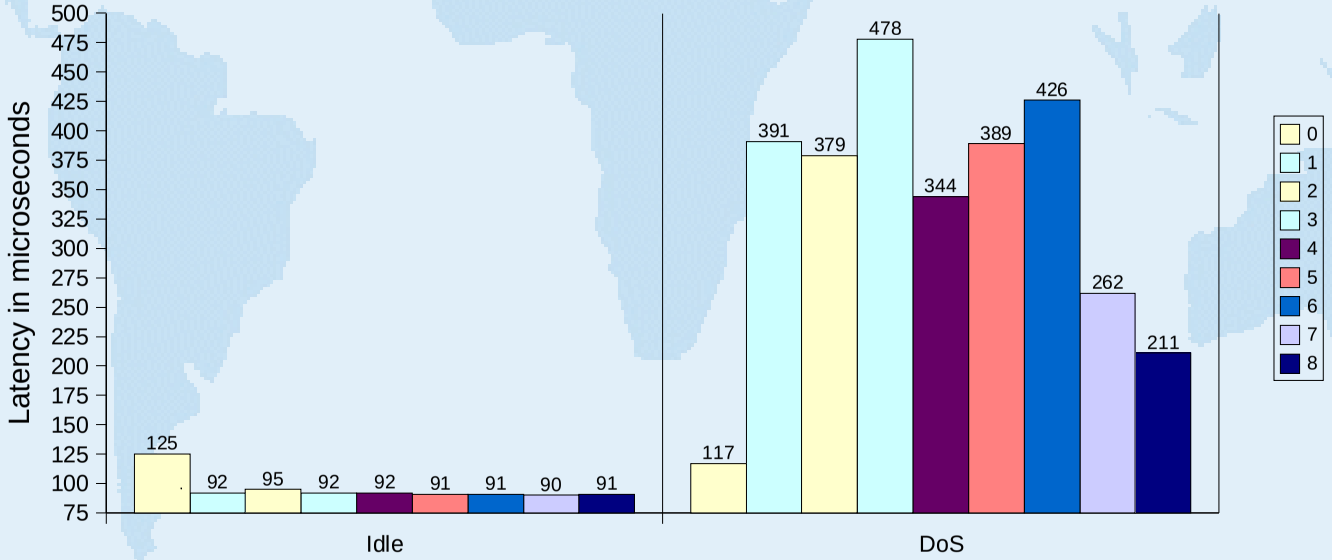
Ping through a idle router



Ping through a router
under a DoS attack 890 kpps



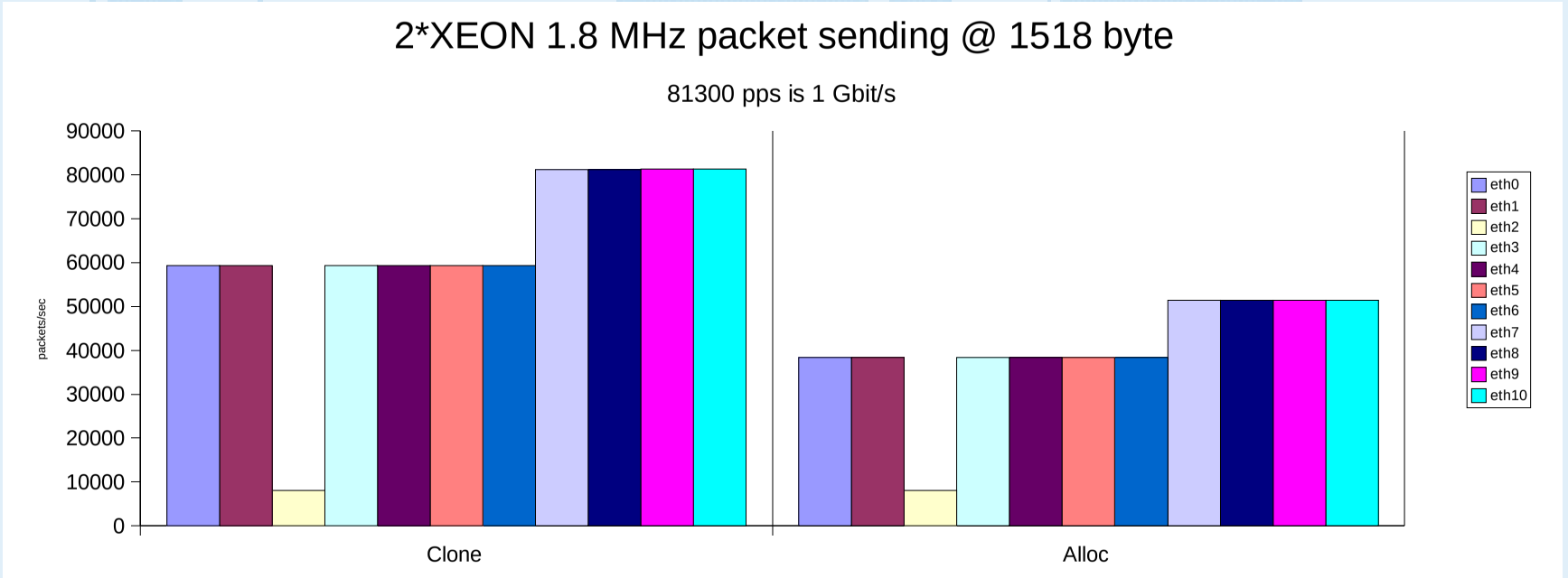
Ping latency/fairness under xtreme load/UP



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

Some GIGE experiments

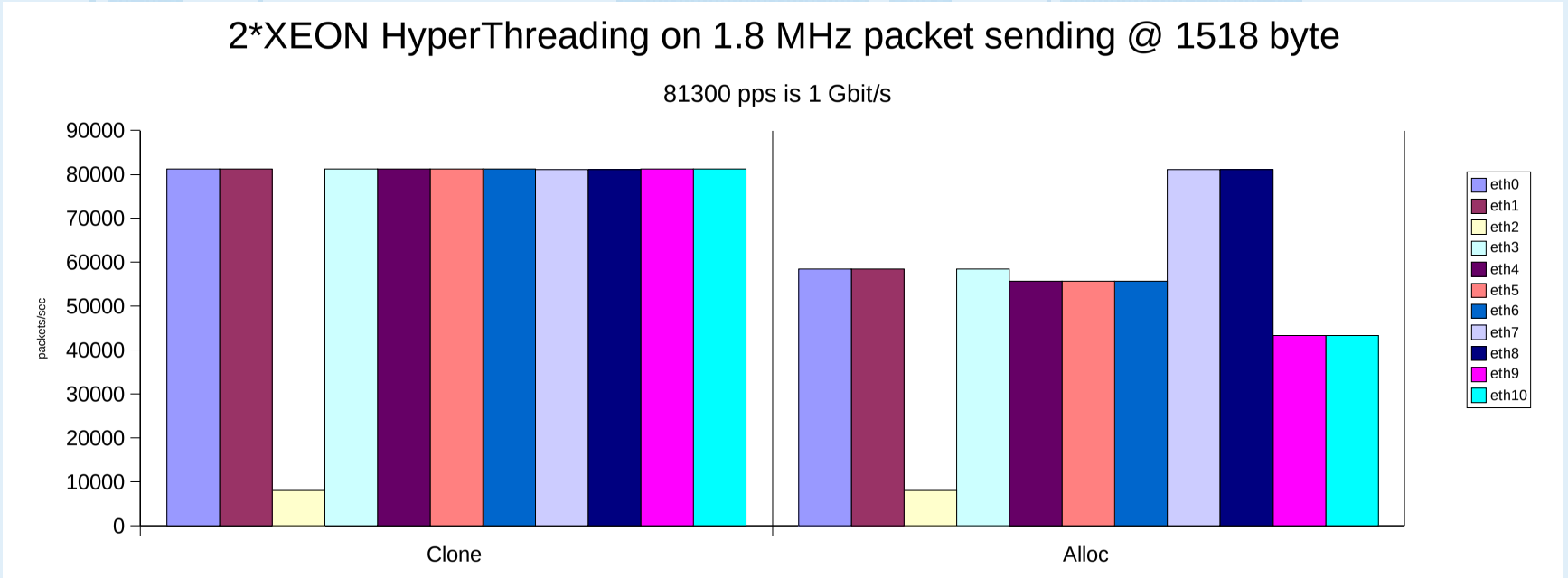
Pktgen sending test w. 11 GIGE
Clone = 8.5 Gbit/s
Alloc = 5.4 Gbit/s



SeverWorks X5DL8-GG Intel e1000

Some GIGE experiments

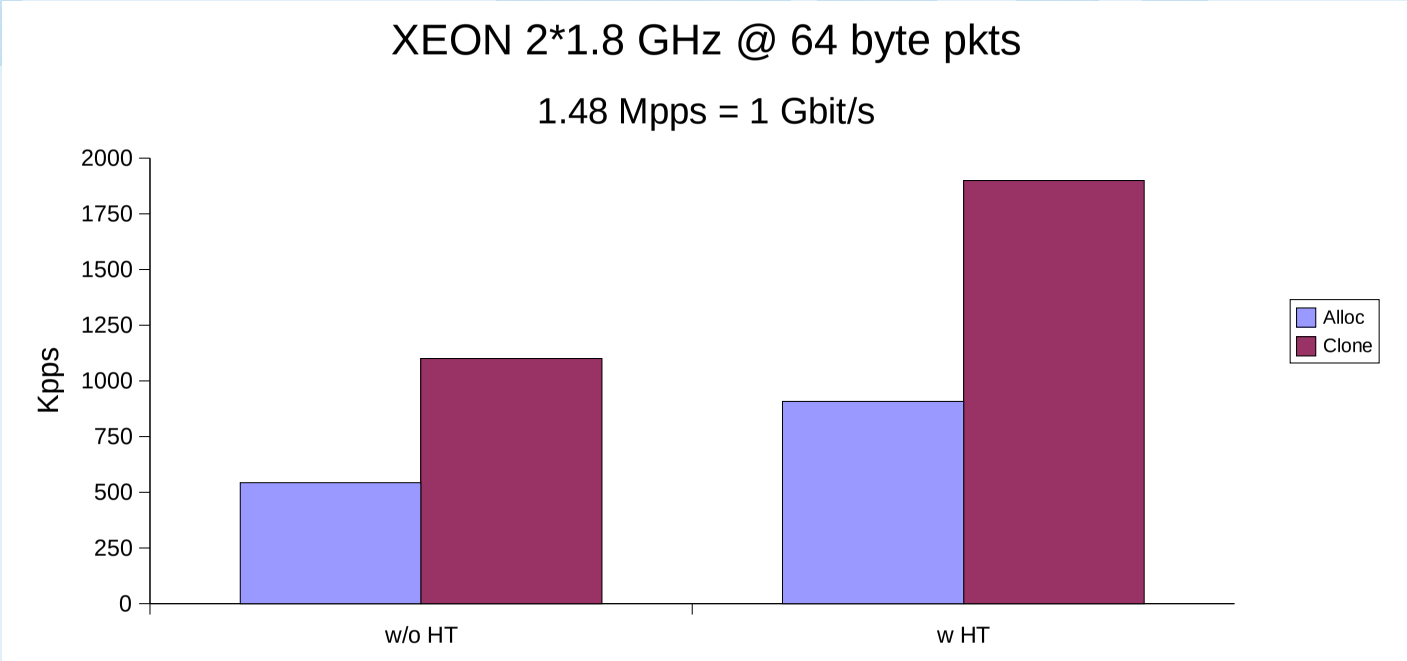
Pktgen sending test w. 11 GIGE
Clone = 10.0 Gbit/s
Alloc = 7.4 Gbit/s



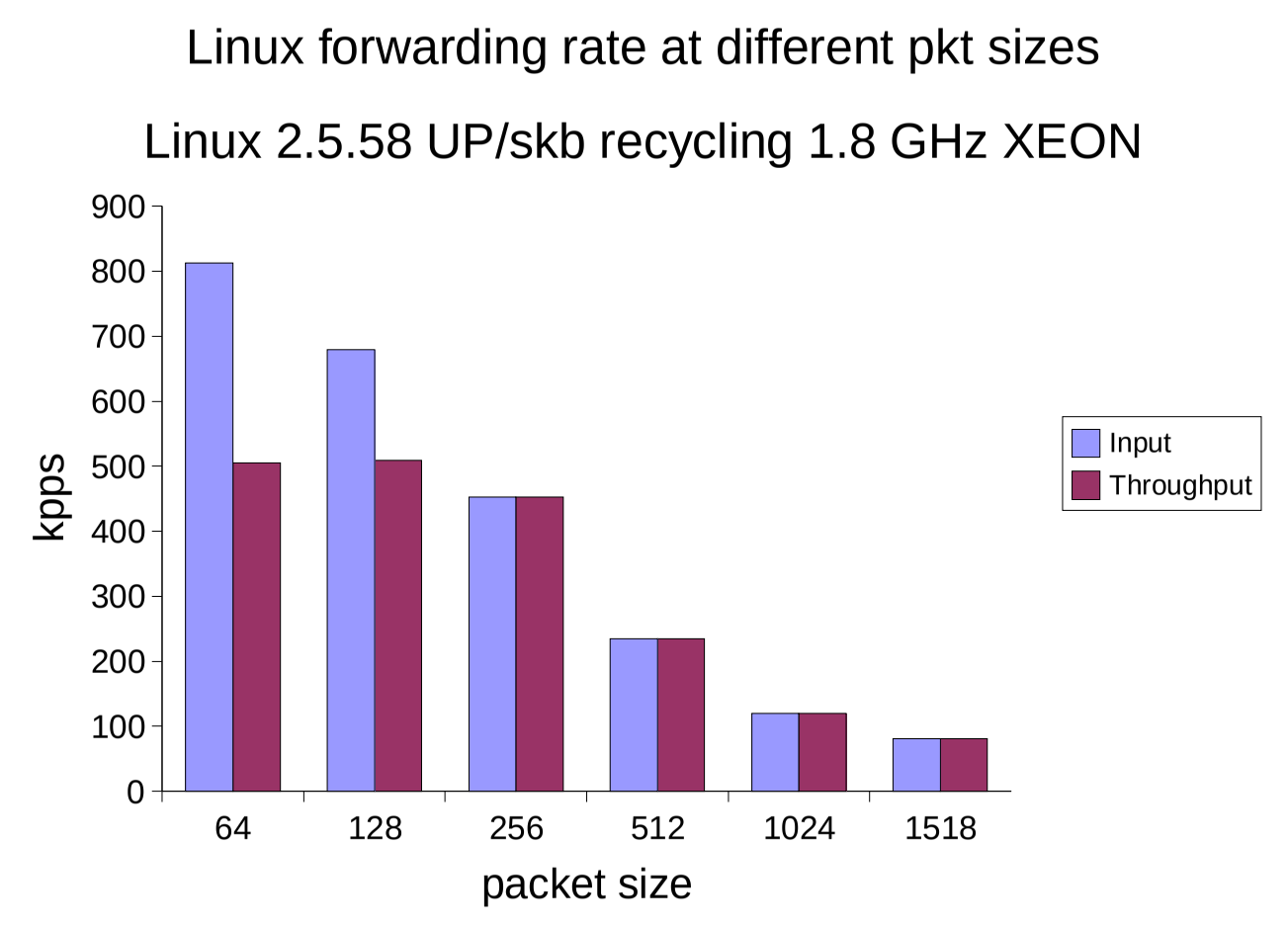
SeverWorks X5DL8-GG Intel e1000

Some GIGE experiments

Aggregated sending performance from pktgen w. 11 GIGE.

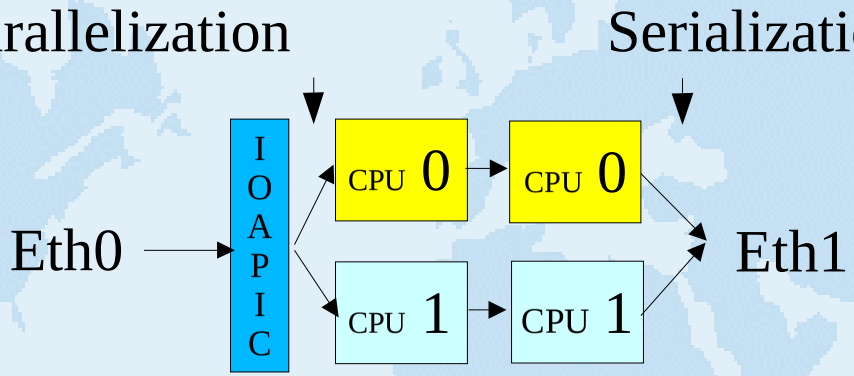


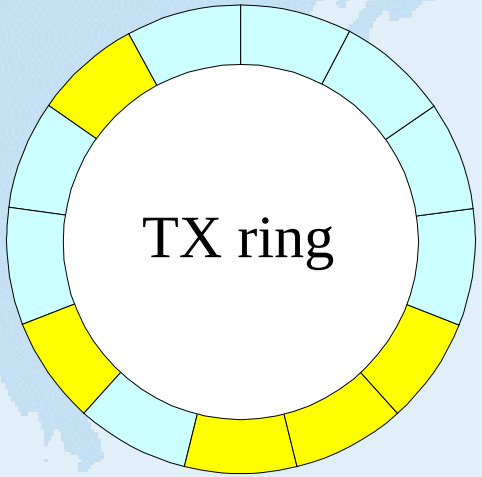
Forwarding performance



Fills a GIGE pipe -- starting from256byte pkts

R&D





For user apps new scheduler does affinty

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

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

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

R&D

Very high transaction packet memory system
for GIGE and upcoming 10GE

Profiling indicates slab is not fully per-CPU

SMP-2-CPU
300 kpps

```
Counter 0 counted GLOBAL_POWER_EVENTS events

vma      samples  %-age       symbol name
c0138e96 37970    8.23162     cache_alloc_refill
c0229490 37247    8.07488     alloc_skb
c0235e90 32491    7.04381     qdisc_restart
c0235b54 27891    6.04657     eth_type_trans
```

SMP-1-CPU
302 kpps

```
c02296d2 25675    8.67698     skb_release_data
c0235b54 24438    8.25893     eth_type_trans
c0235e90 24047    8.12679     qdisc_restart
c0229490 18188    6.14671     alloc_skb
c0110a1c 15741    5.31974     do_gettimeofday
```

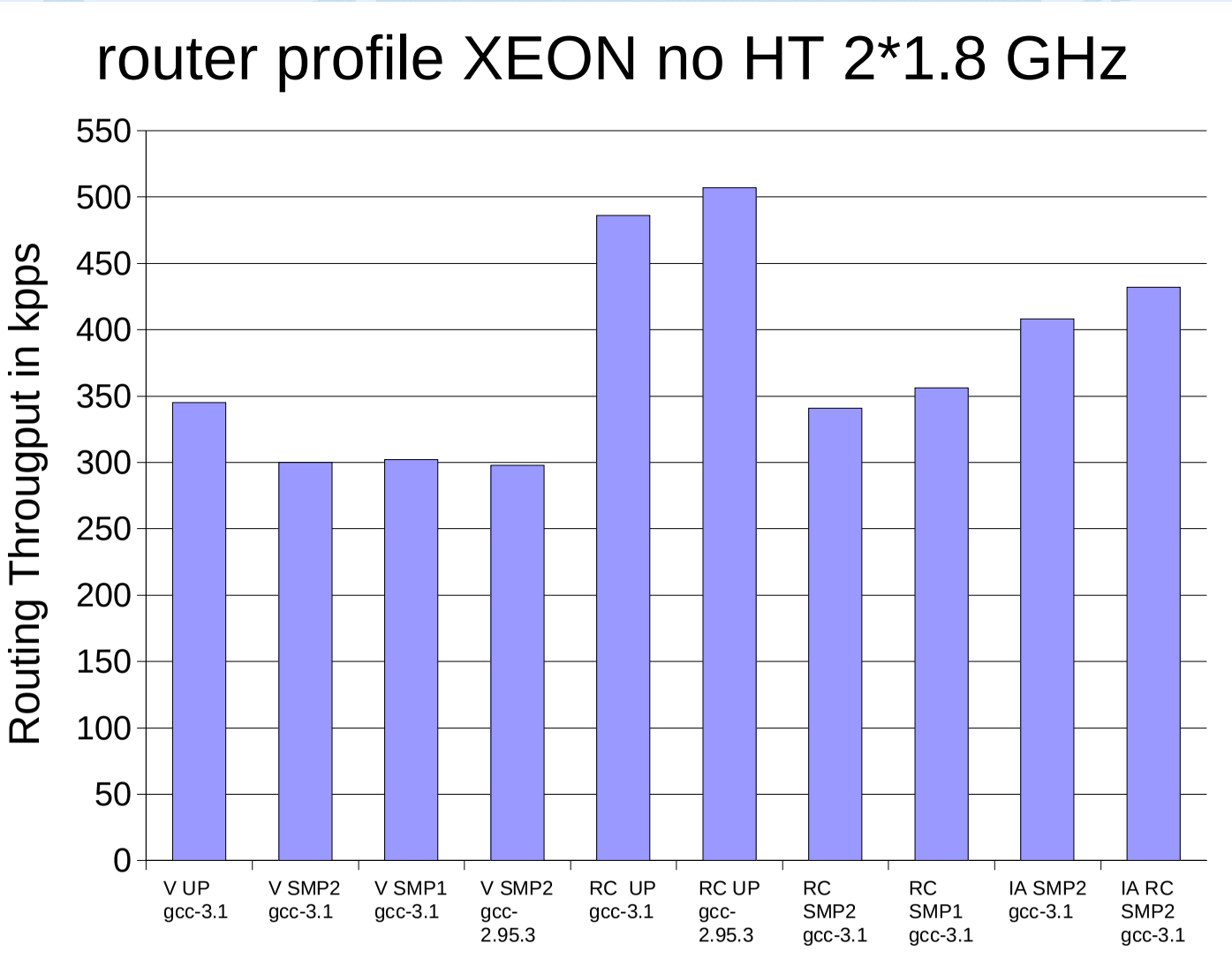
Note setting input affinity helps.
But we like to work on the general problem

R&D

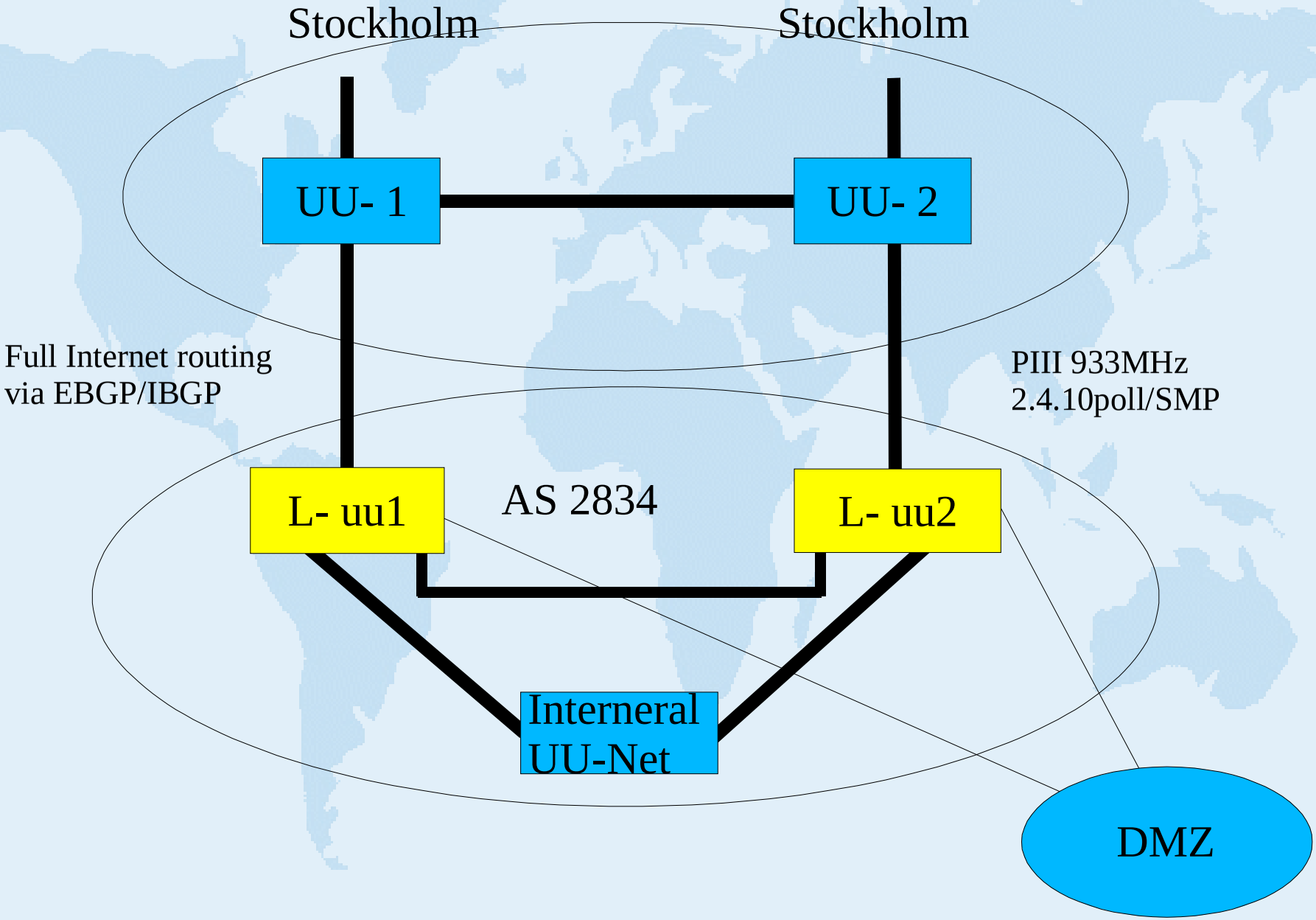
V=vanilla
UP=uniprossor
SMP1= SMP 1 CPU
SMP2= SMP 2 CPU
RC= skb recycling
IA=input affinity

Profile with p4/xeon
performance counters

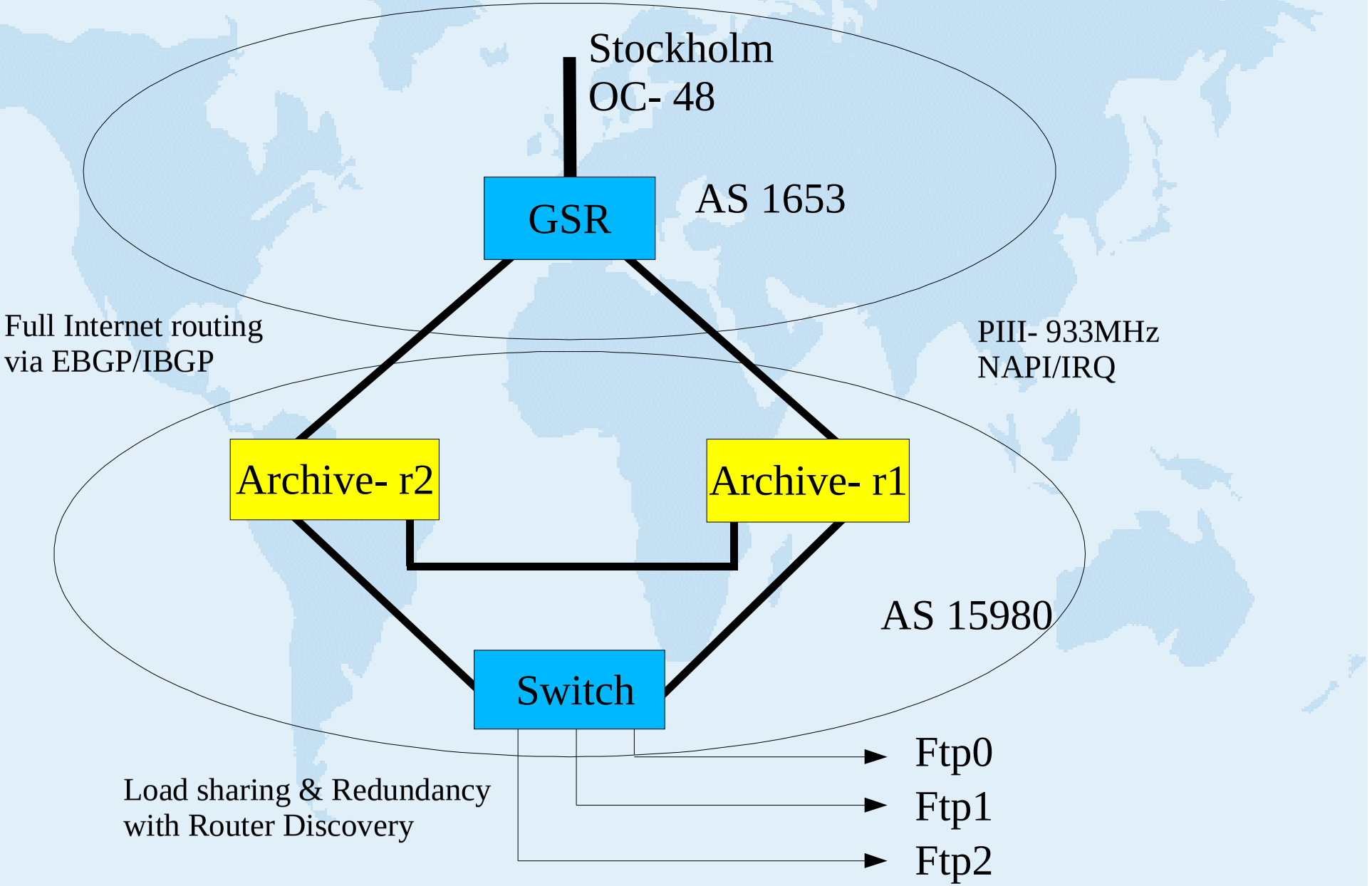
GLOBAL_POWER_EVENTS
MISPRED_BRANCH_RETIRED
BSQ_CACHE_REFERENCE
MACHINE_CLEAR
ITLB_REFERENCE



NAPI/SMP production in use: uu.se



Real World use:*ftp.sunet.se*



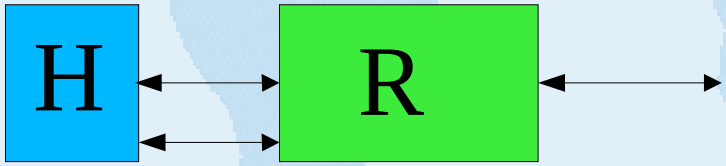
IP-login -- a Linux router app.

user authenticated routing

user@host IP- login router

User's can only reach the IP- login router. This hosts a web server.

User web requests are directed to webserver and asked for username, password ev. Authetication server. Today TACACS



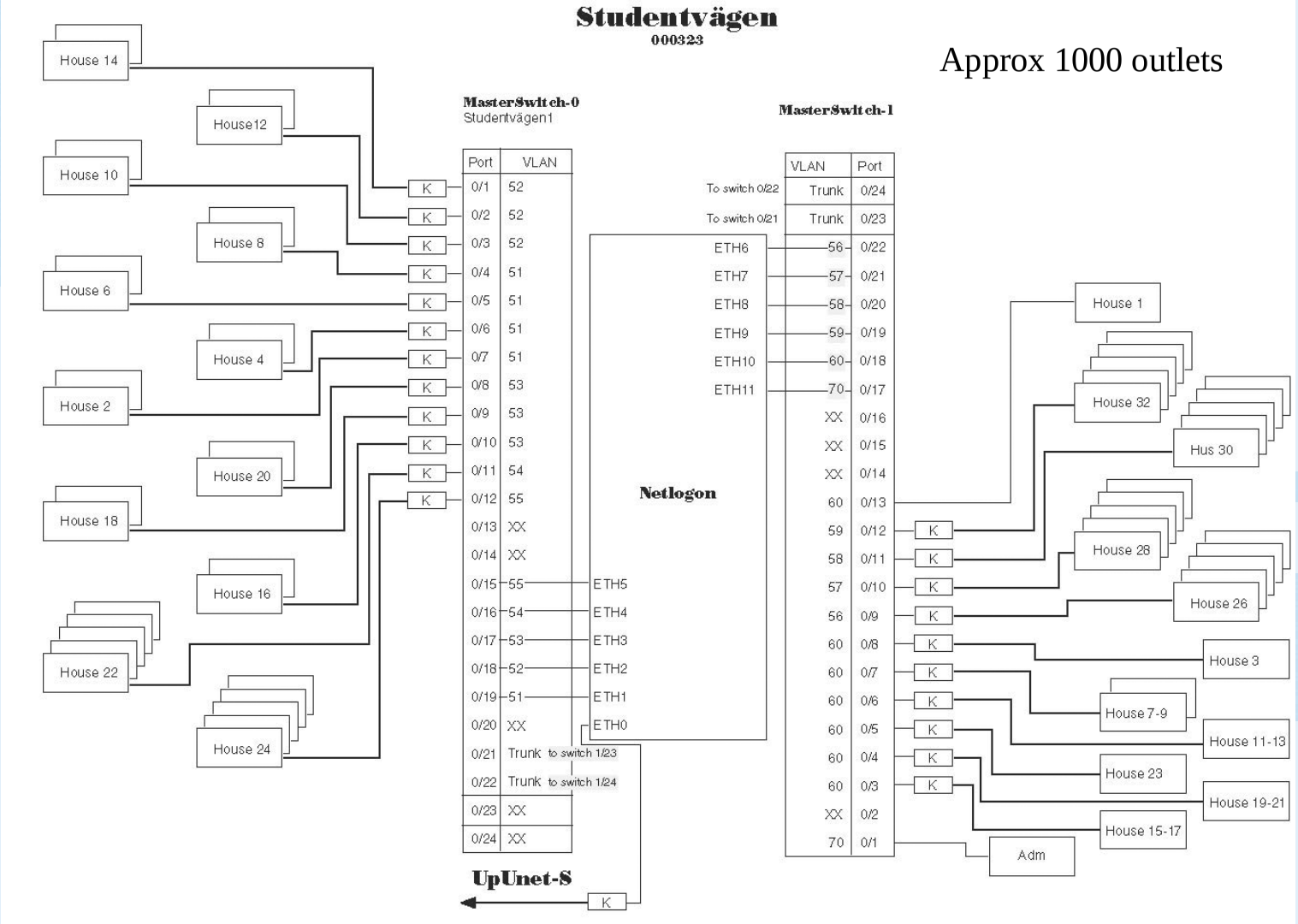
If user/passwd is accepted.
1) Forwarding is enabled for host
2) Monitoring arping is started

Based on stolen code from:
Pawel Krawczyk -- tacacs client
Alexey Kuznetsov -- arping

Loss of arping disables forwarding.

IP-login installation

at Uppsala University



A new network symbol has been seen...

The Penguin Has Landed

References and Other Stuff

- http://bifrost.slu.se
  - Claim they can do 435 Kpps on PIII 700
- http://www.pdos.lcs.mit.edu/click/
- http://www.cyberus.ca/~hadi/usenix-paper.tgz
  - Some other work
- http://robur.slu.se/Linux/net-development/