Open Source Routing
KTH CSD Kick-Off Workshop

Robert Olsson Uppsala University

2008-09-02

Why Open Source?

- Reclaim research and development to universities etc
- To be a part in the development loop
- Open for wide collaboration
  - No national boundaries
  - No organizational boundaries
- Easy experimentation to prototype new ideas
  - Next-Generation Internet take-off
  - Other ideas we can't even think of right now

Why Open Source?

- Possibilities for superior quality
  Work can be reviewed by many people
- Very fast development can be achieved
- Process can be independent from business or politics
- Non-discriminate
- Economical possibilities
- Idea started in computer science

Relation to Open Source

- Your are getting other people's work for "free"
  Respect
- Open Source does not work without contributions
  Compare a relay race. Reuse and recycle work.
- Open Source has strong momentum
  Business models are developed etc

Open Source Networking Now

- Interesting suitable hardware
  - Technological breakthrough
    - Multi-Core CPU, other silicons
    - Fiber Optics
    - Fast buses PCI-Express
- There are interesting applications
- Open source OS has come a long way

Open Source Competitors

- MIT click modular router
- Berkeley, CA
  XORP
- Vyatta

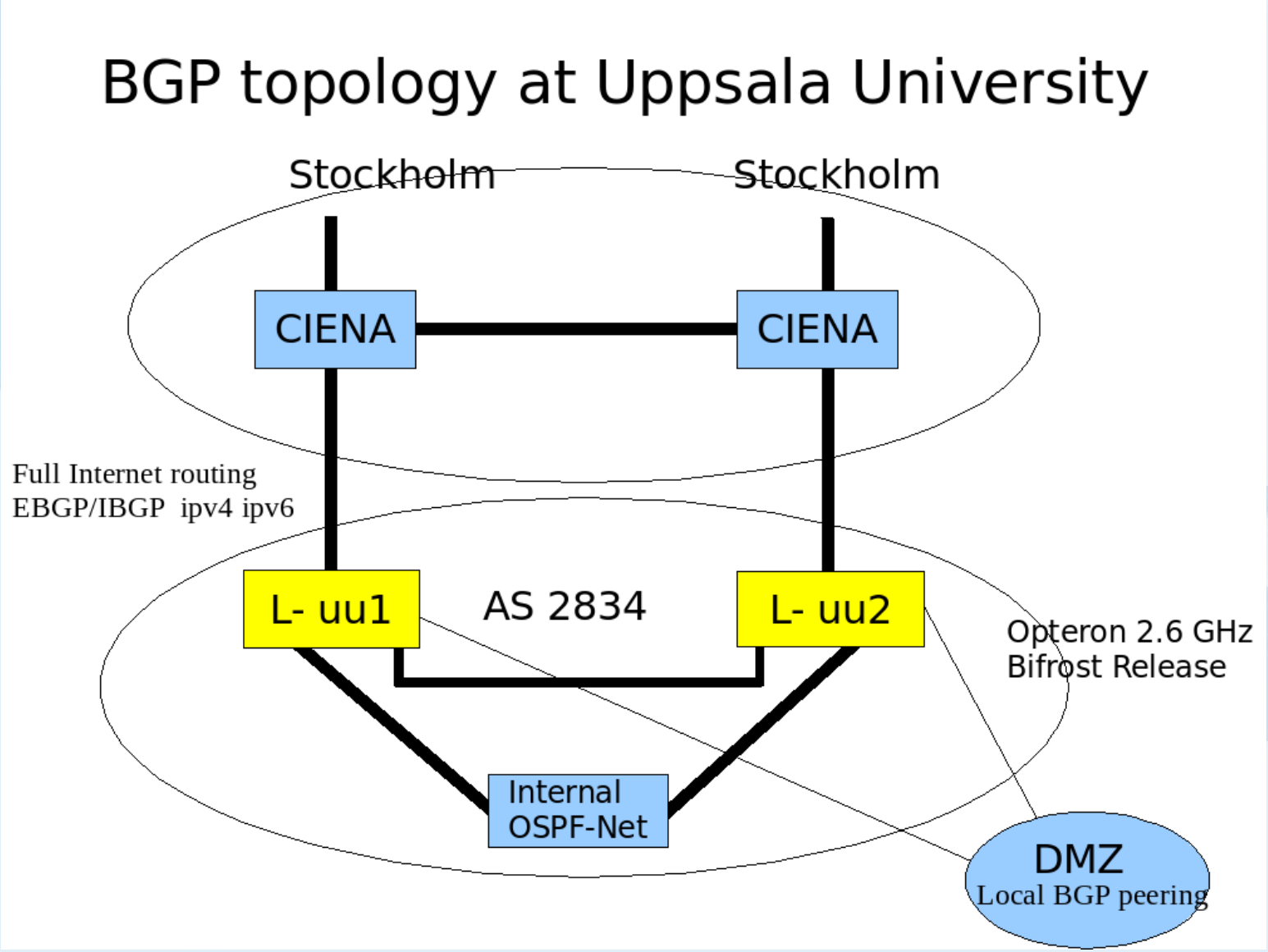
Over 10 years in production

UU facts

Dual ISP BGP connect GIGE
Local BGP peering GIGE
Ipv4/Ipv6
OSPFv2/OSPFv3
netfilter
Cisco 6500

10g planned.

Over 10 years in production

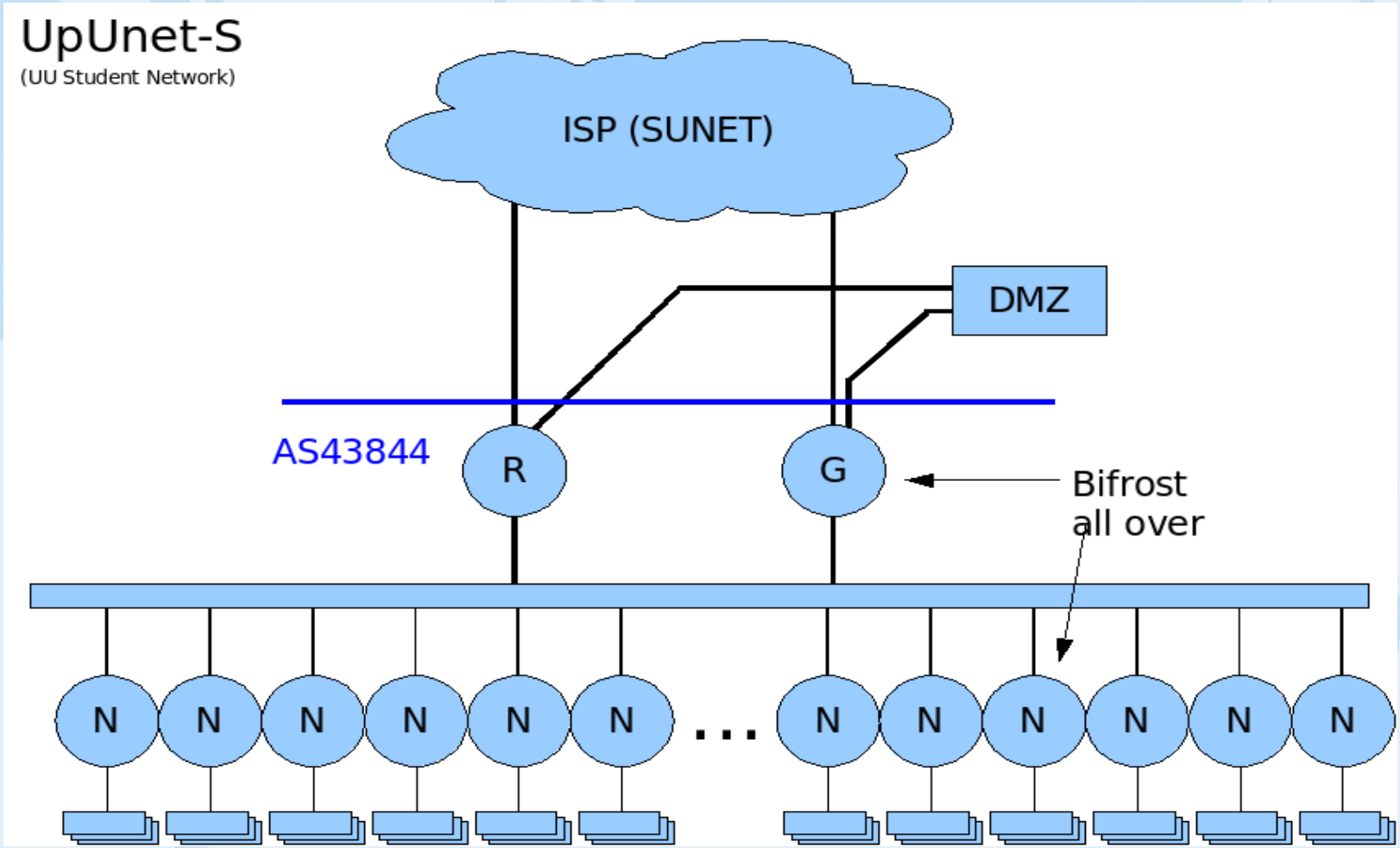


Over 10 years in production

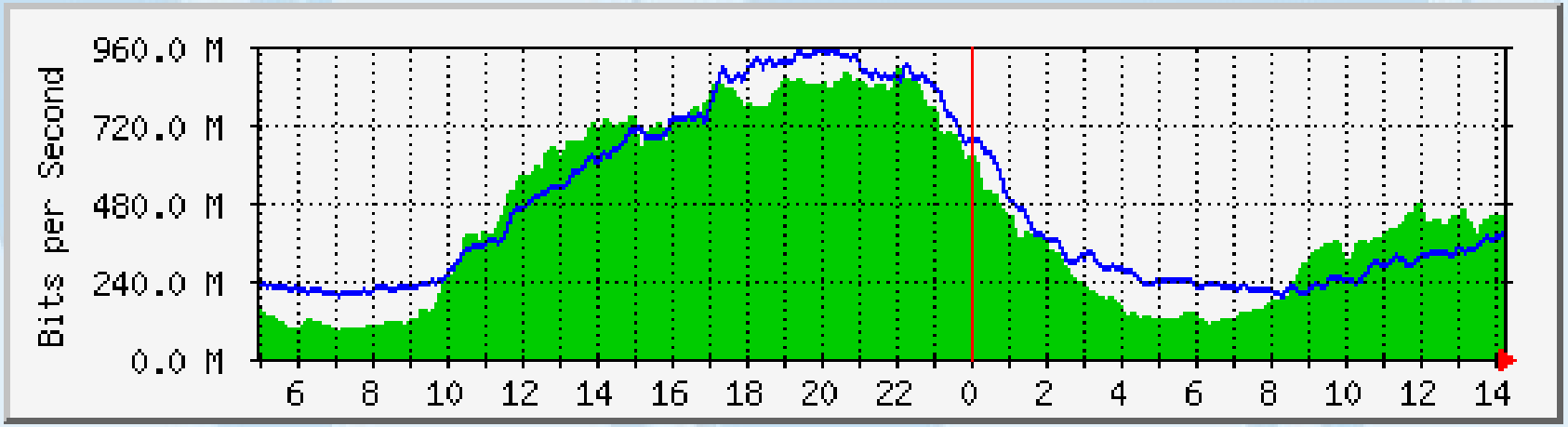
- Three major installations

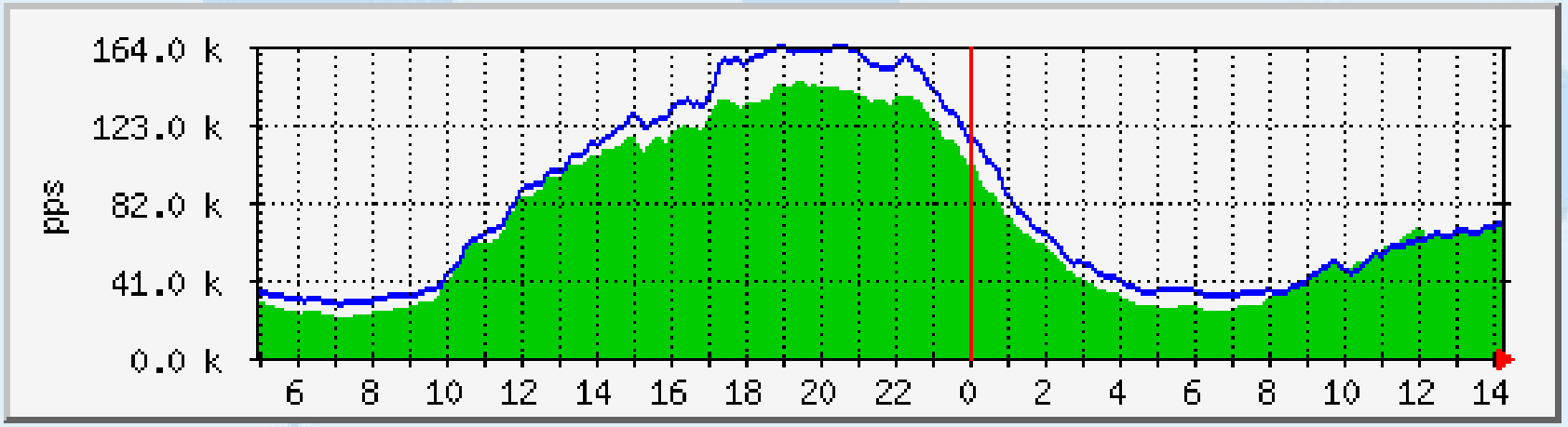
- UU core routers towards SUNET
- UU StudentNetwork 30.000 students
- ftp.sunet.se

Over 10 years in production



Over 10 years in production
Student Network Core Router





Over 10 years in production

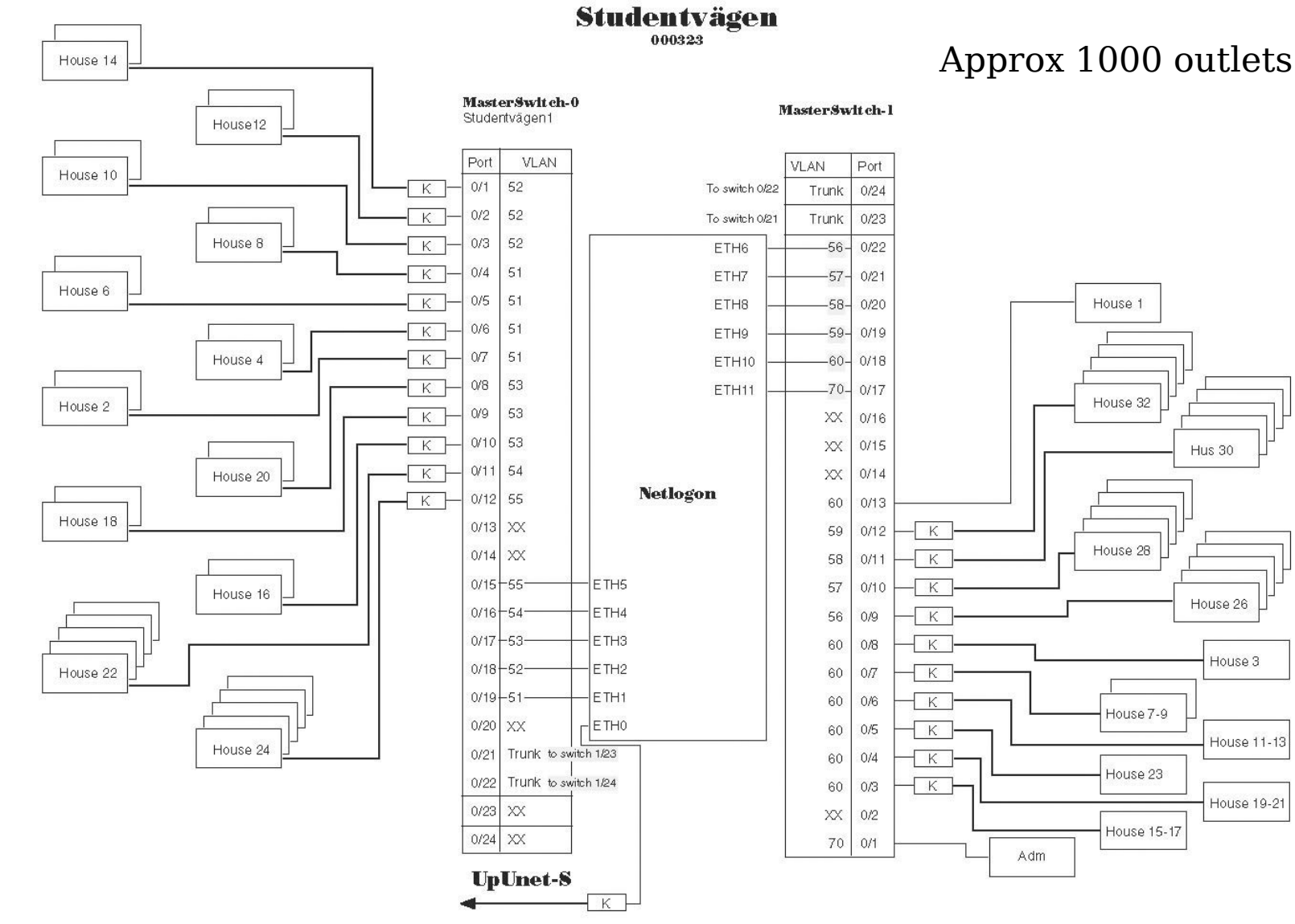
Student Network facts

Dual ISP BGP connect GIGE
Local BGP peering GIGE
Ipv4
OSPFv2
xxx netfilter rules
netlogin-service at premises

10g planned.

IP-login installation

at Uppsala University



Testing, Verification Development & Research

- Started out as simple testing.
- Curiosity, Open Source, Collaboration
- Relatively freedom , the idea to use in own infrastructure. No need for external funding.
- OS was intended for desktops.

Testing, Verification Development & Research

No need for test network. We could test in own infrastructure. (Or SLU)

Problem oriented vs Project oriented

We could work on complicated issues

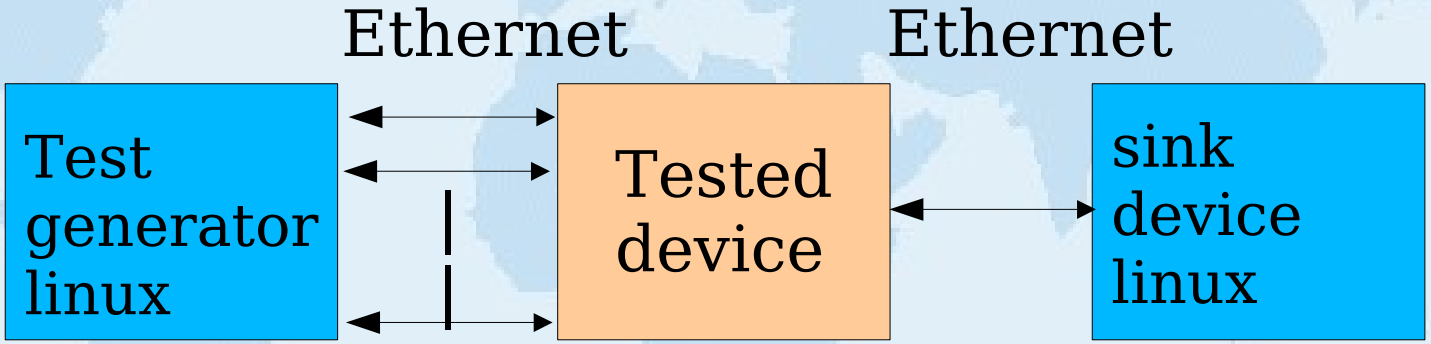
- NAPI 3years
- pktgen 2years
- fib_trie 1year
- TRASH 1year

Building Blocks

- Hardware:
  - PC
    - Motherboard/CPU/Memory
    - Network Interfaces
      - GIGE/10g WiFi etc
- Software
  - Operating System
    - Linux/BSD/Microsoft
    - Applications
      - Routing Daemons
        - Quagga/XORP
      - IP-login/netlogon
- Network
  - Cable, Fiber, Copper
  - Equipment, Switches

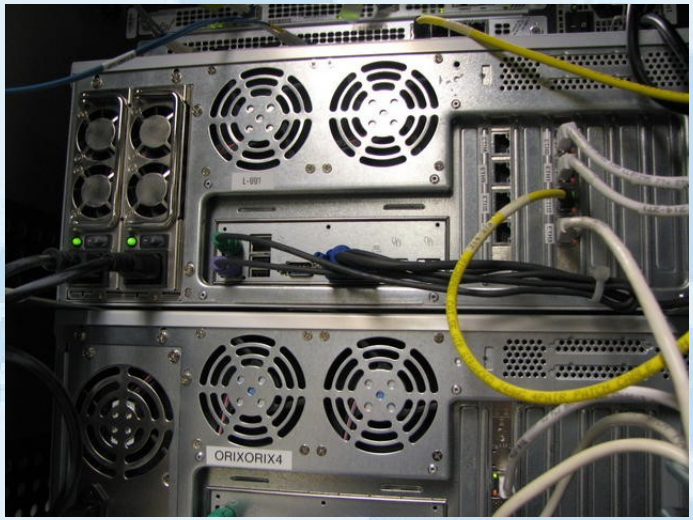
Flexible netlab at Uppsala University

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

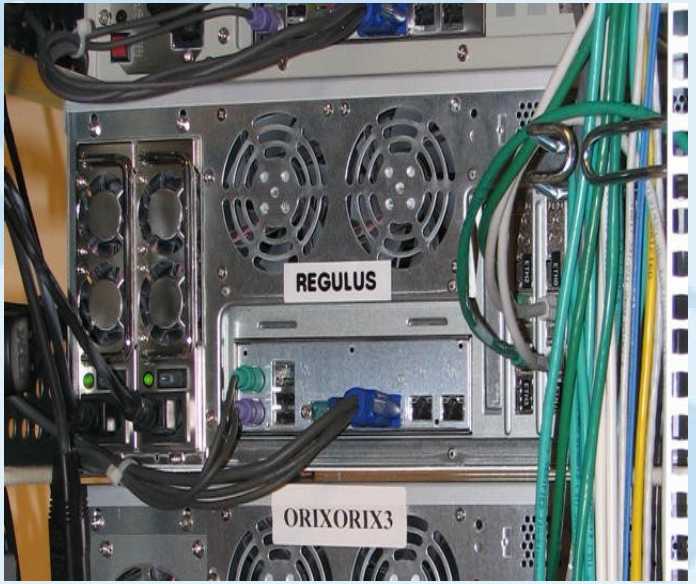


* Raw packet performance
* TCP
* Timing
* Variants

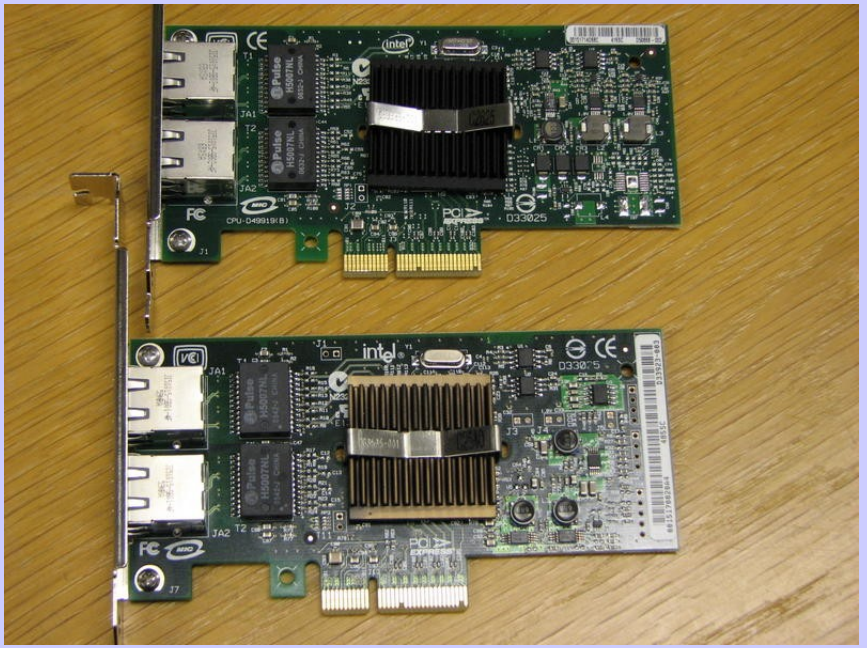
Lab

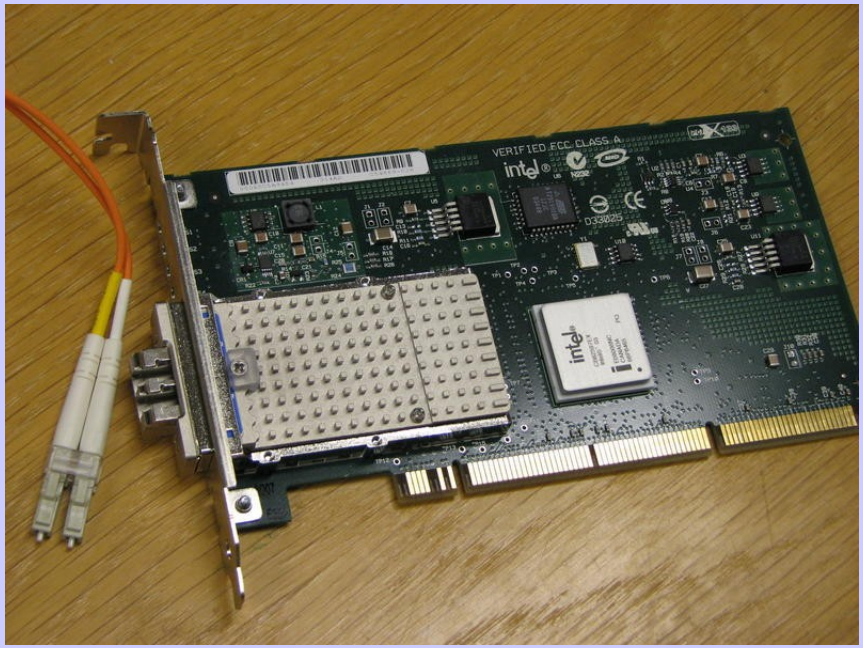

ORIXORIX4

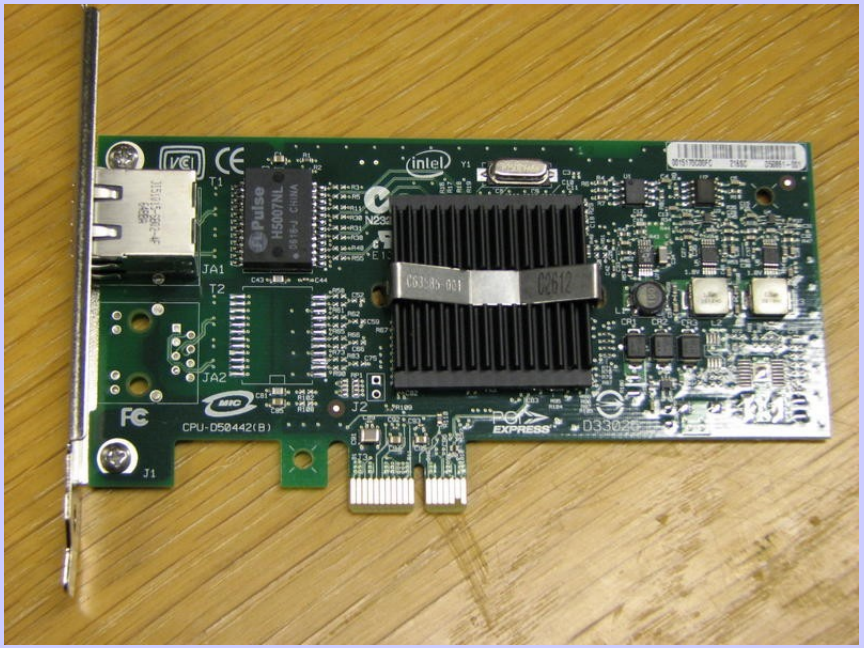

IBM
IBM
ARCHIVE-1
ARCHIVE-2


REGULUS
ORIXORIX3

IntelNIC's







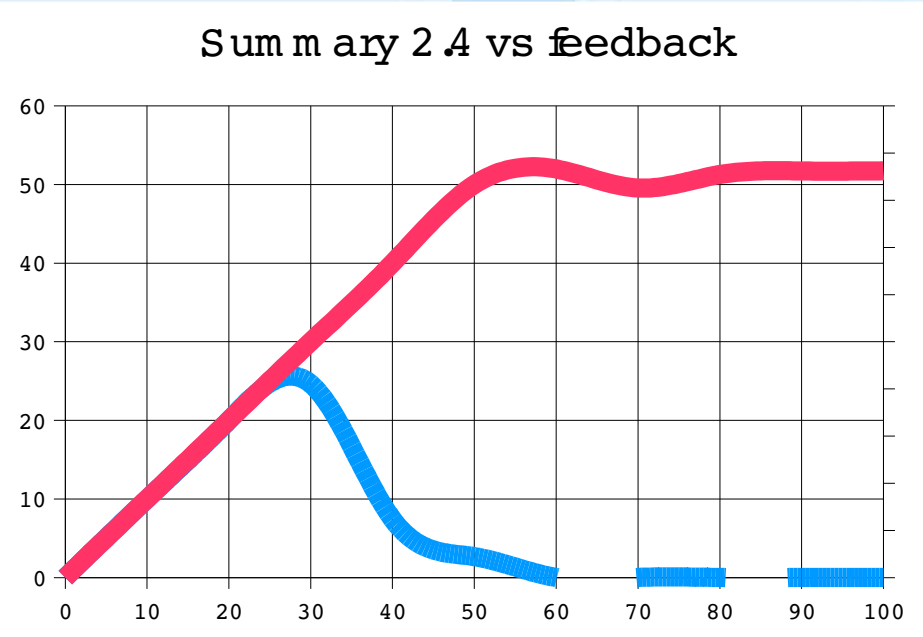
Notorwere blessed…

Bifrost concept

- Linux kernel collaboration
- 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)

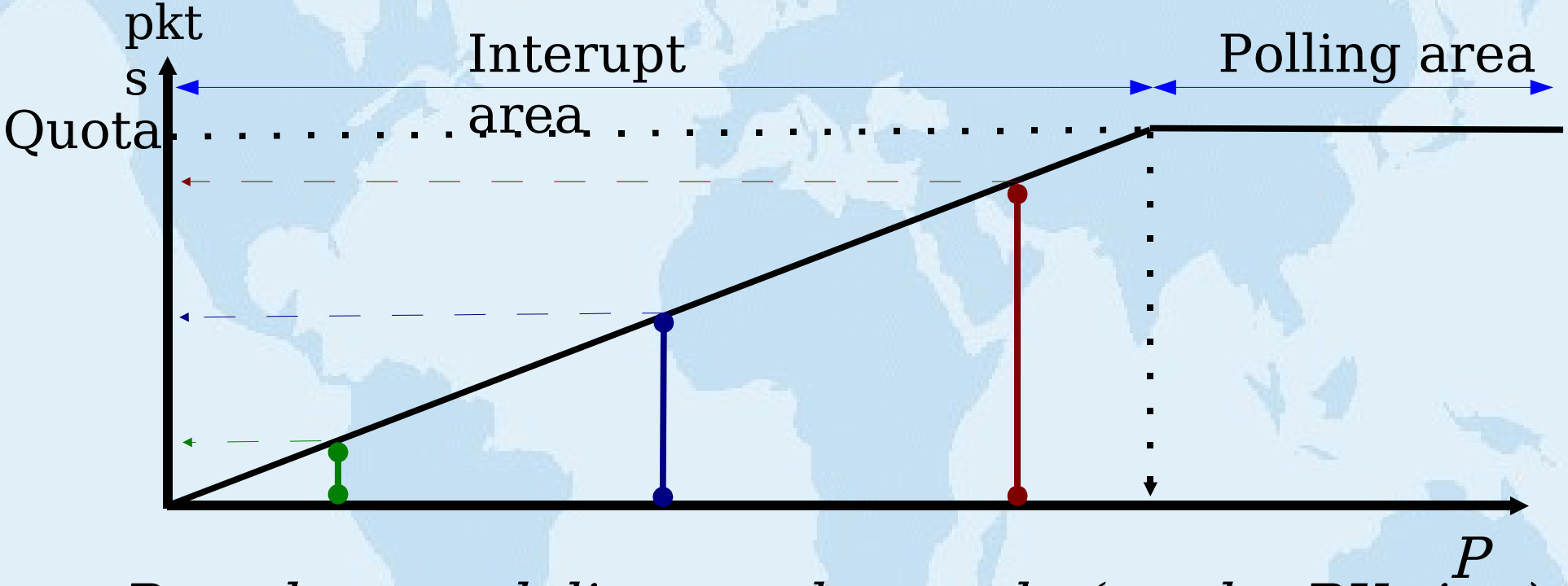
Overall Effect

- 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

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 ir
- Area under curve shows how many packets before next
- *Quota* enforces fair share

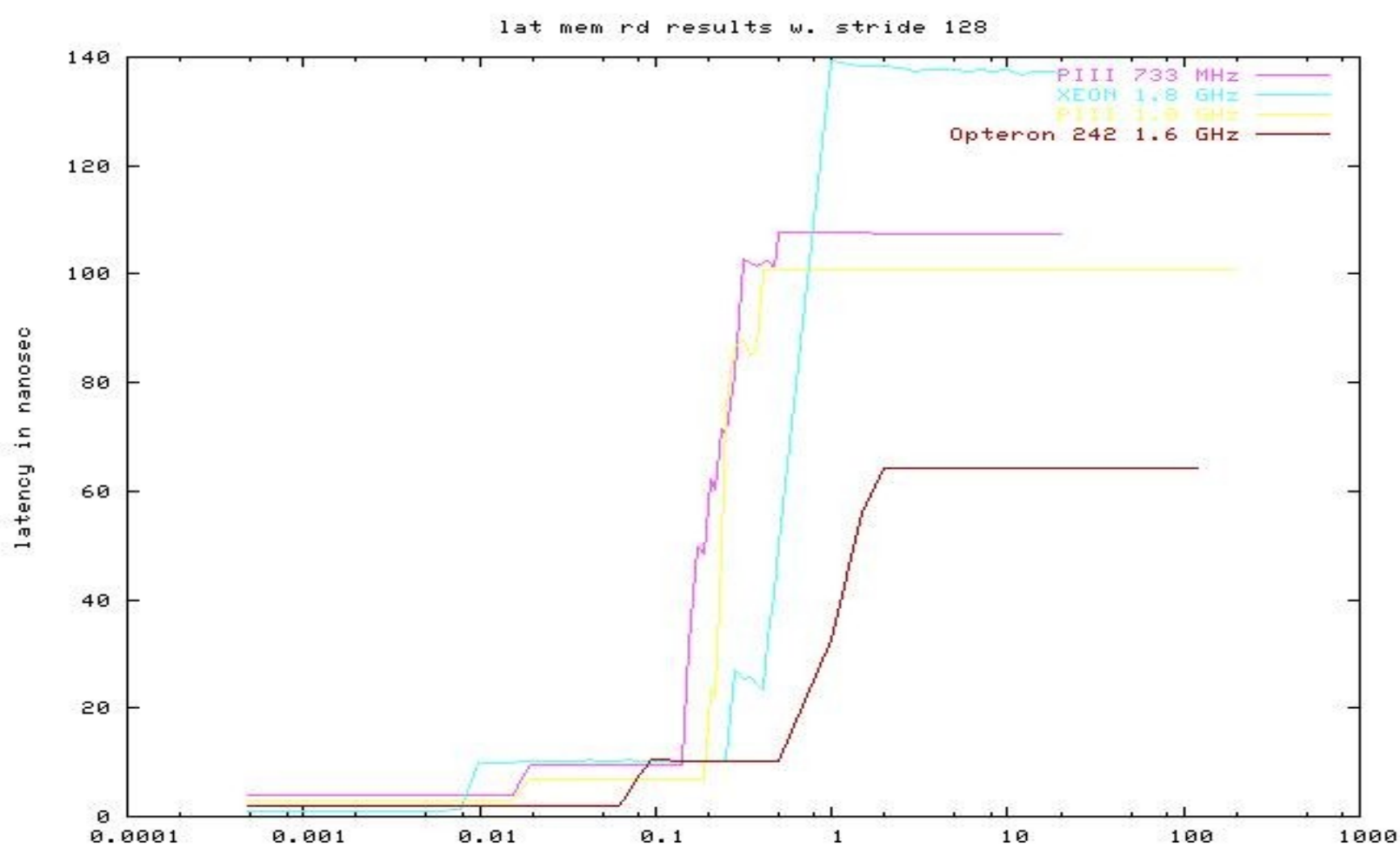
Kernel support

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

Cache effect/Performance



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

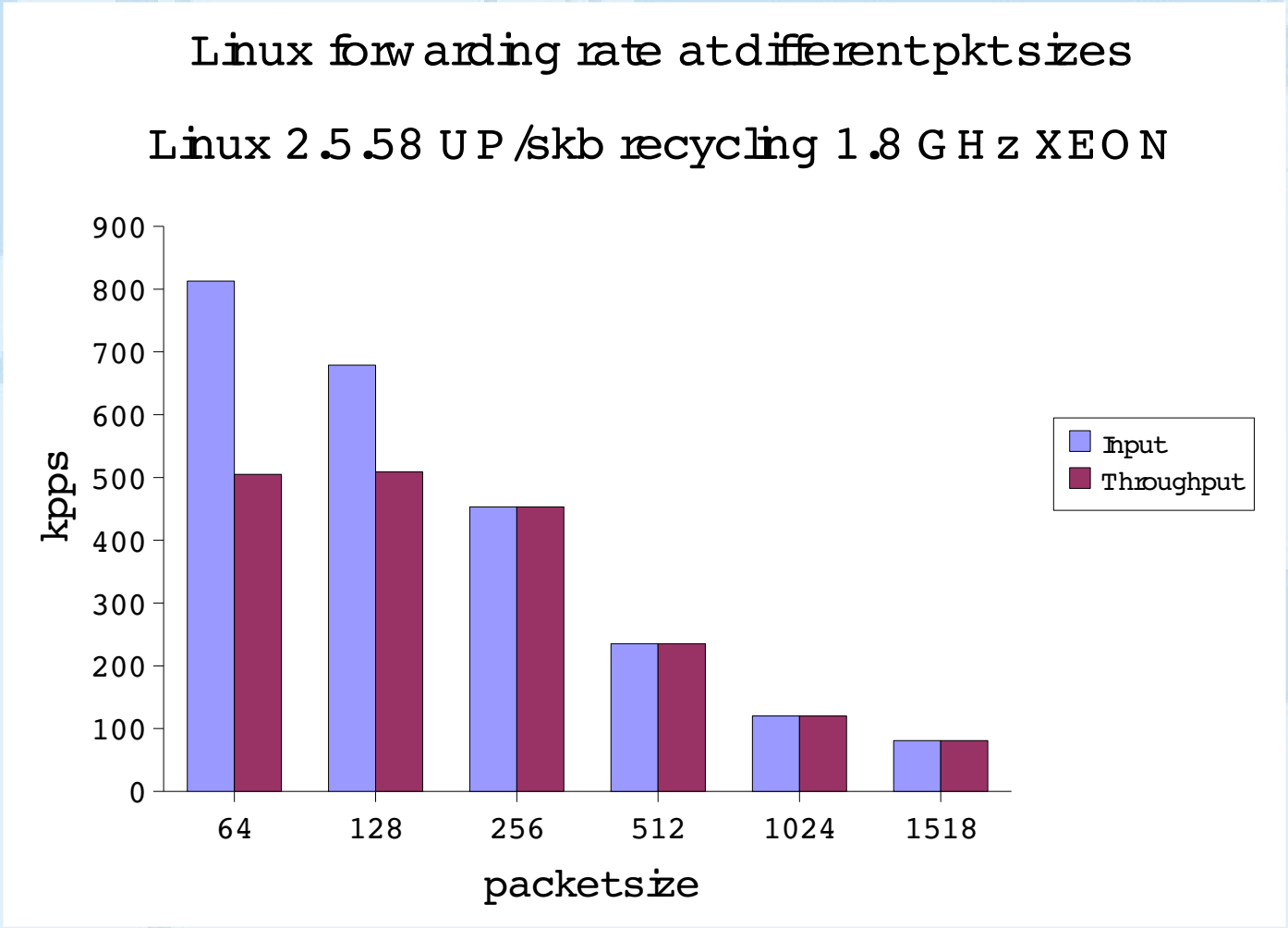
DMA copies of data into RAM

Does prefetch solve problems?

A new network symbol has been seen...

The Penguin Has Landed

Forwarding performance



Fills a GIGE pipe -- starting from256byte pkts

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

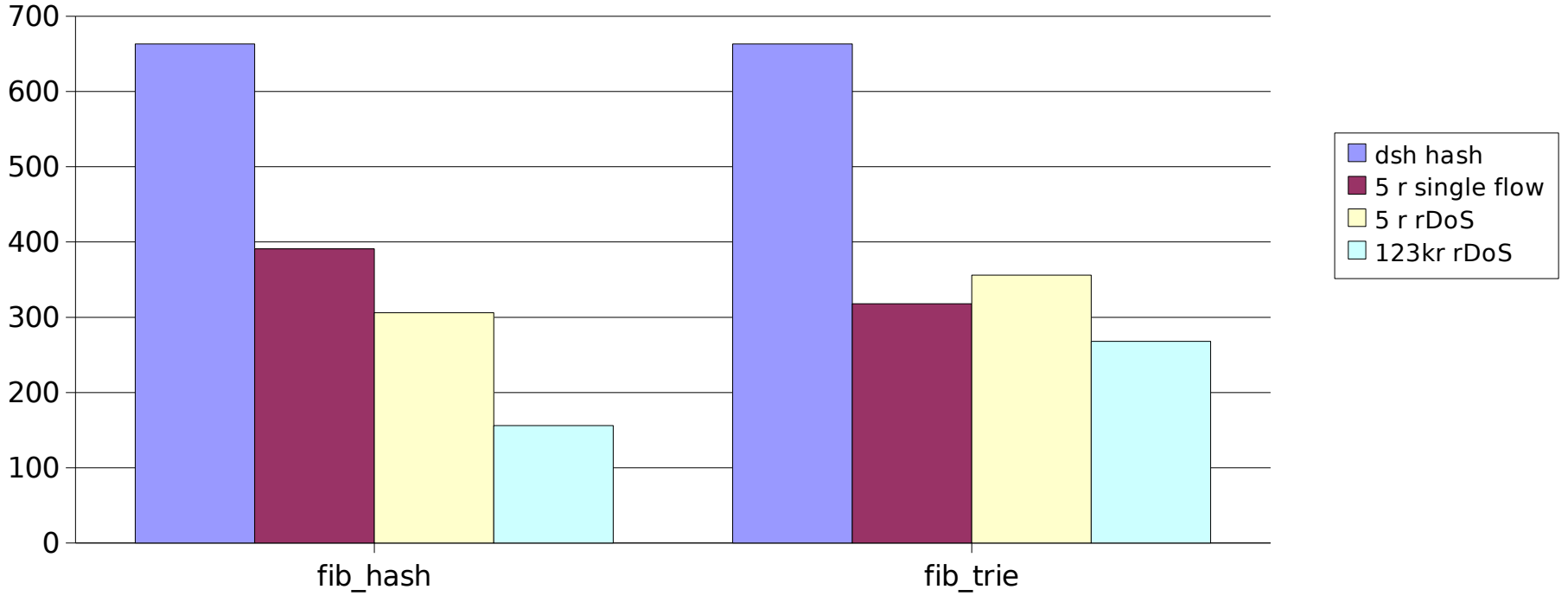
Discussed LC-trie with Alexey Kuznetsov.

LC-trie investigations. Got GPL from authors.

fib_trie performance comparison

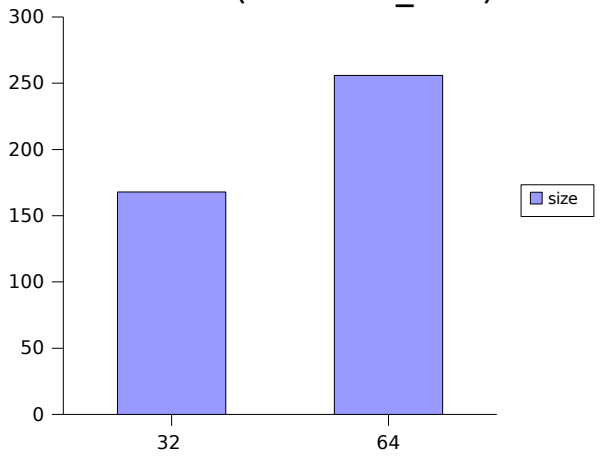
forwarding kpps

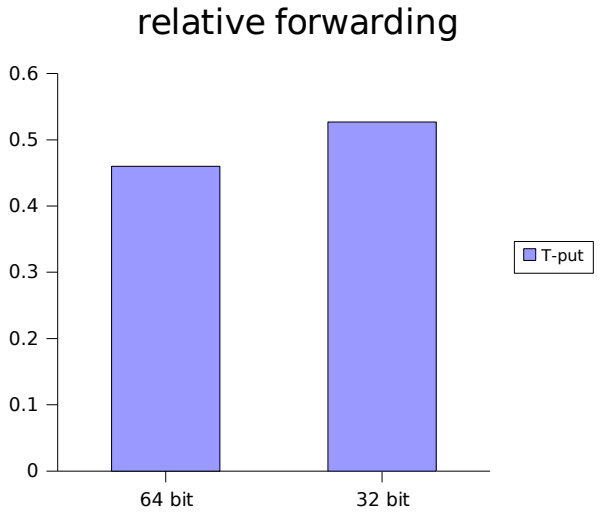
Linux 2.6.16 1 CPU used(SMP) Opteron 1.6 GHz e1000



Preroute pathes to disable route hash

32/64 bit || sizeof(sk_buff)



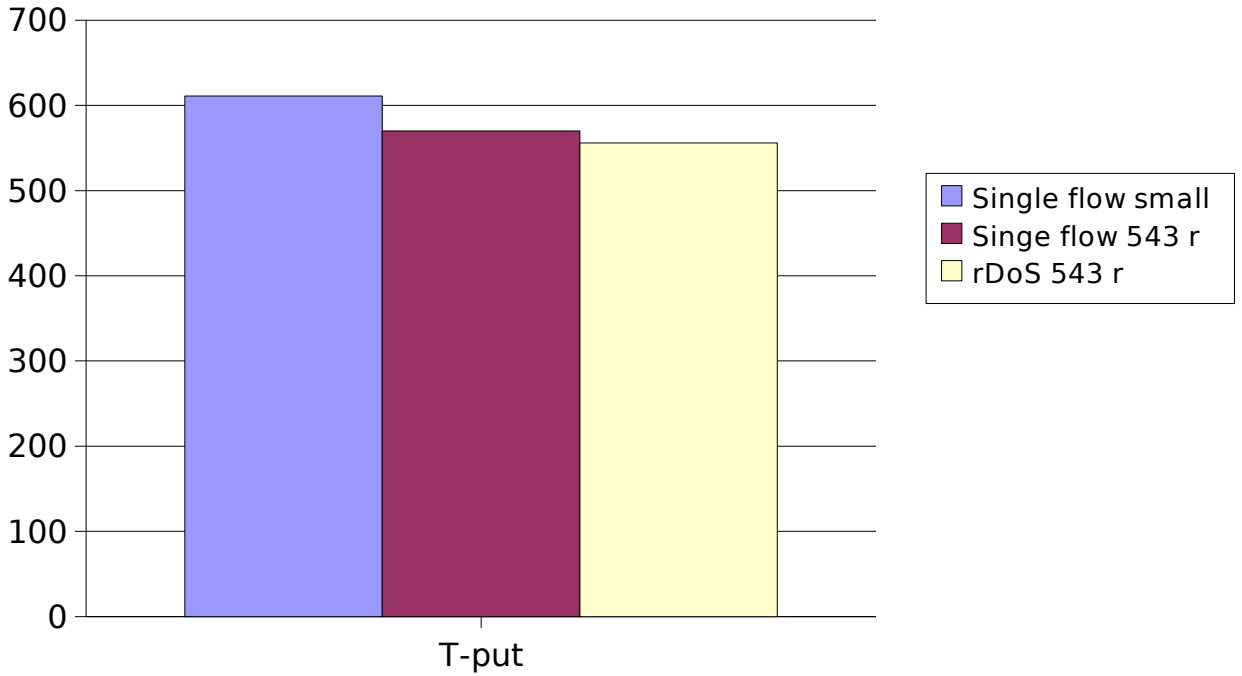


Gcc 3.4 x86_64 vs i686 on same HW

ipv6 performance

Forwarding kpps 76 byte pkt.

Linux 2.5.12 1 CPU(SMP) Opteron 1.6 GHz e1000



How rDoS work on sparse routing table?

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 fullkey bokup. src/dst/sport/dport/proto/if etc and later socket.

Foreven ip6 with little perform nace degradation

Could be a candidate for the grand unified bokup

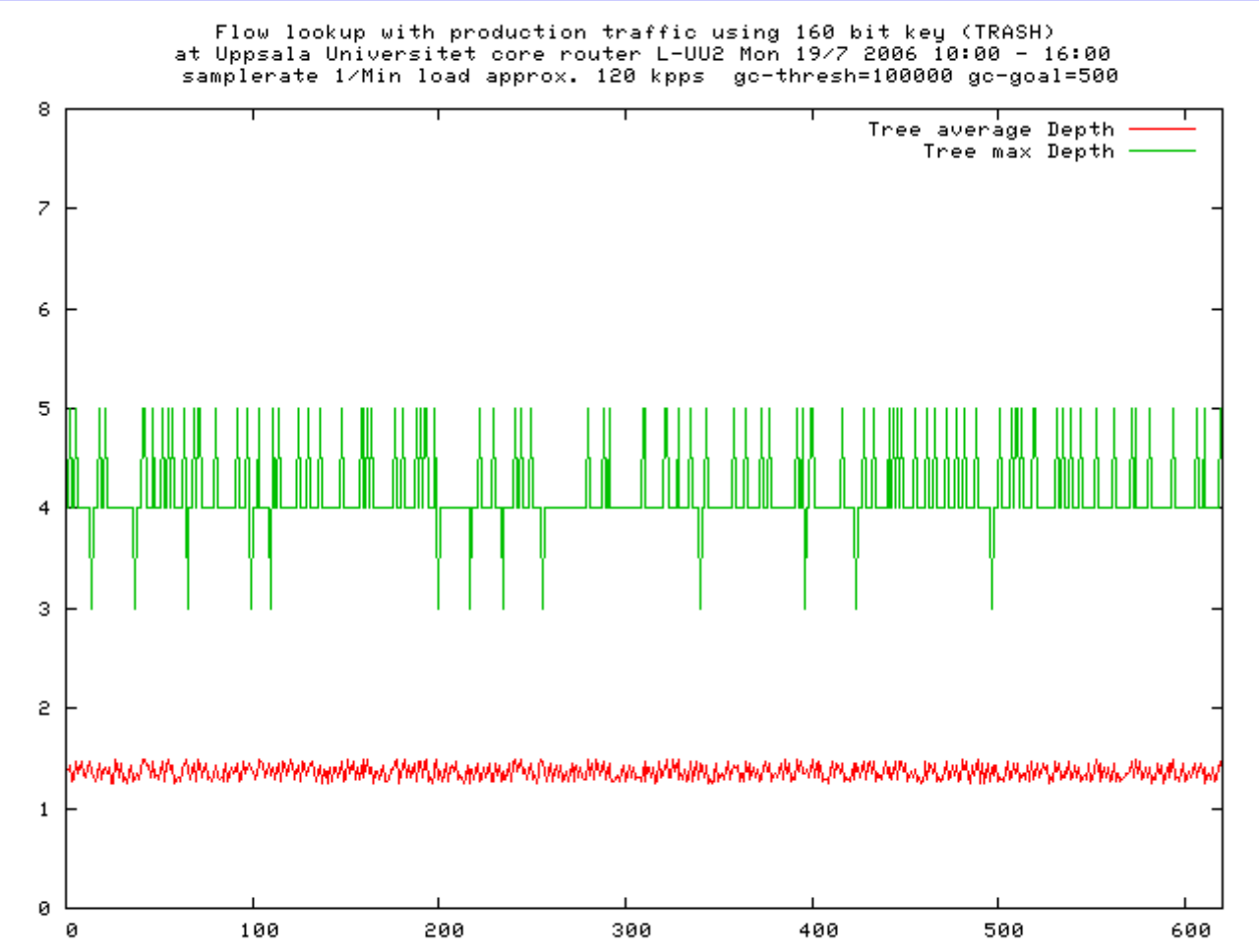
Fullflow bokup can understand connections.

Free flow bgging etc

New garbage collection (GC) possible. Active GC stated AGC in the paper. Listen to TCP SYN, FIN and RST Show to be perform ance winner.

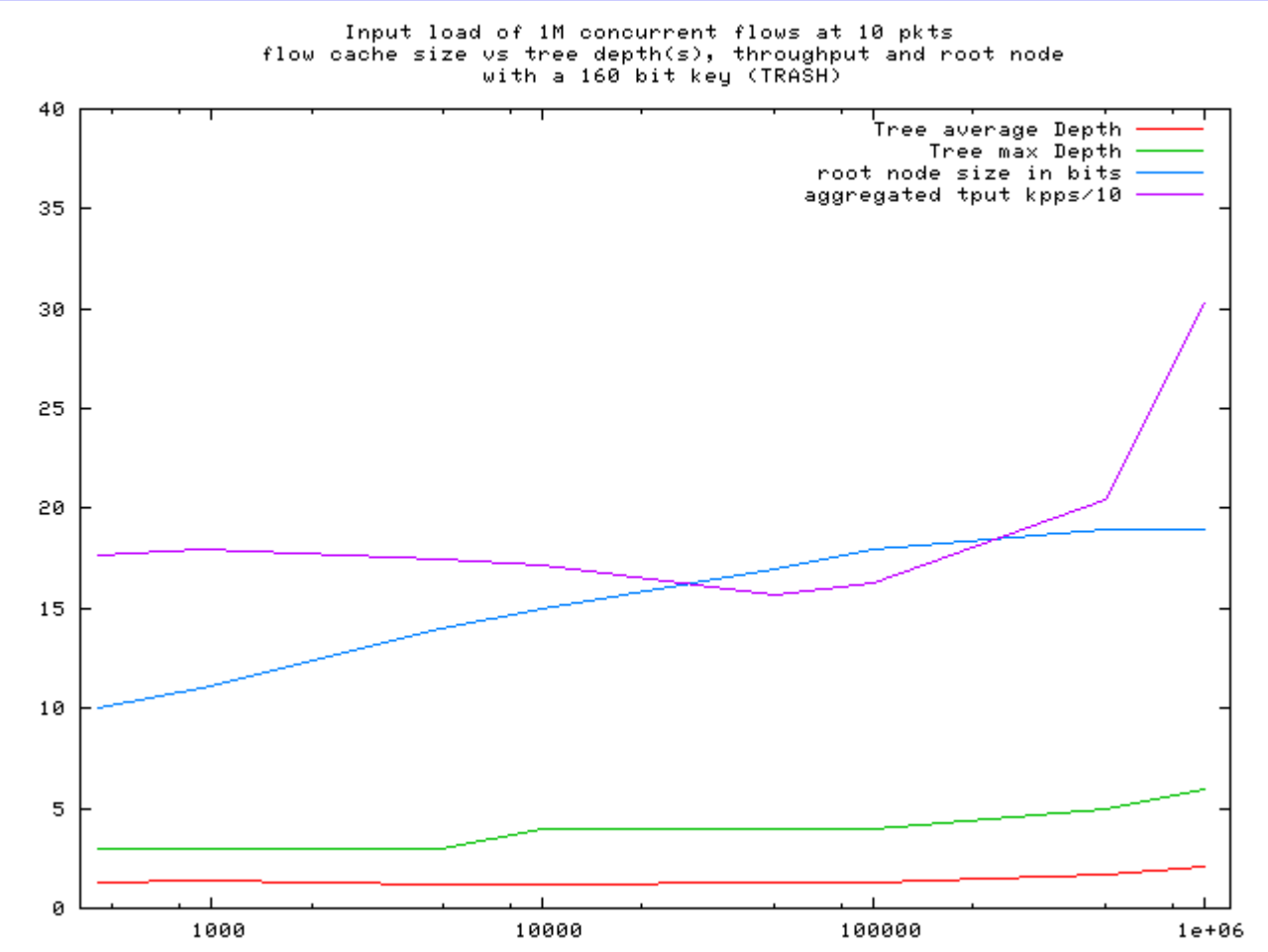
Trash datastucture

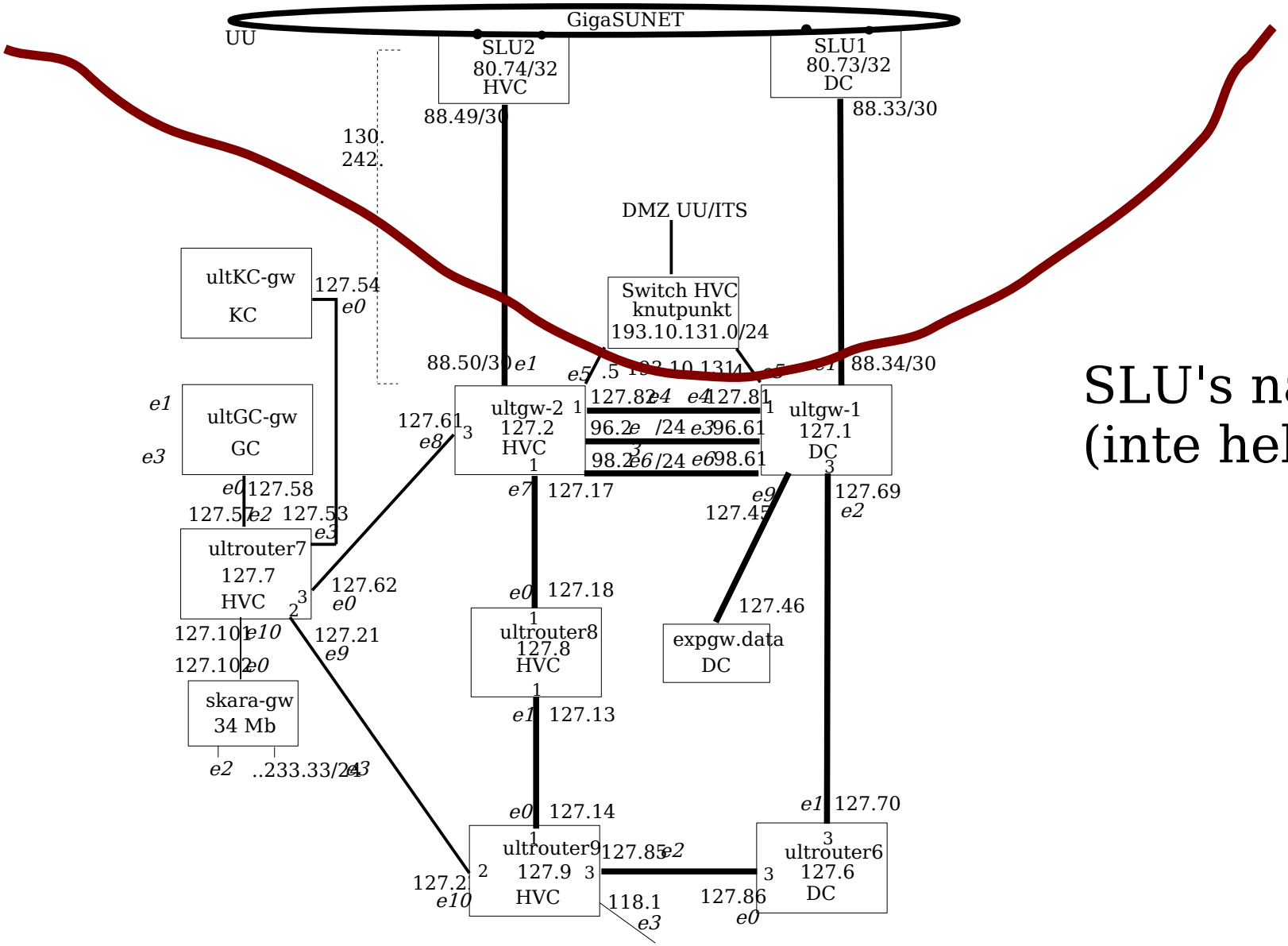
Uppsala Universitet core router



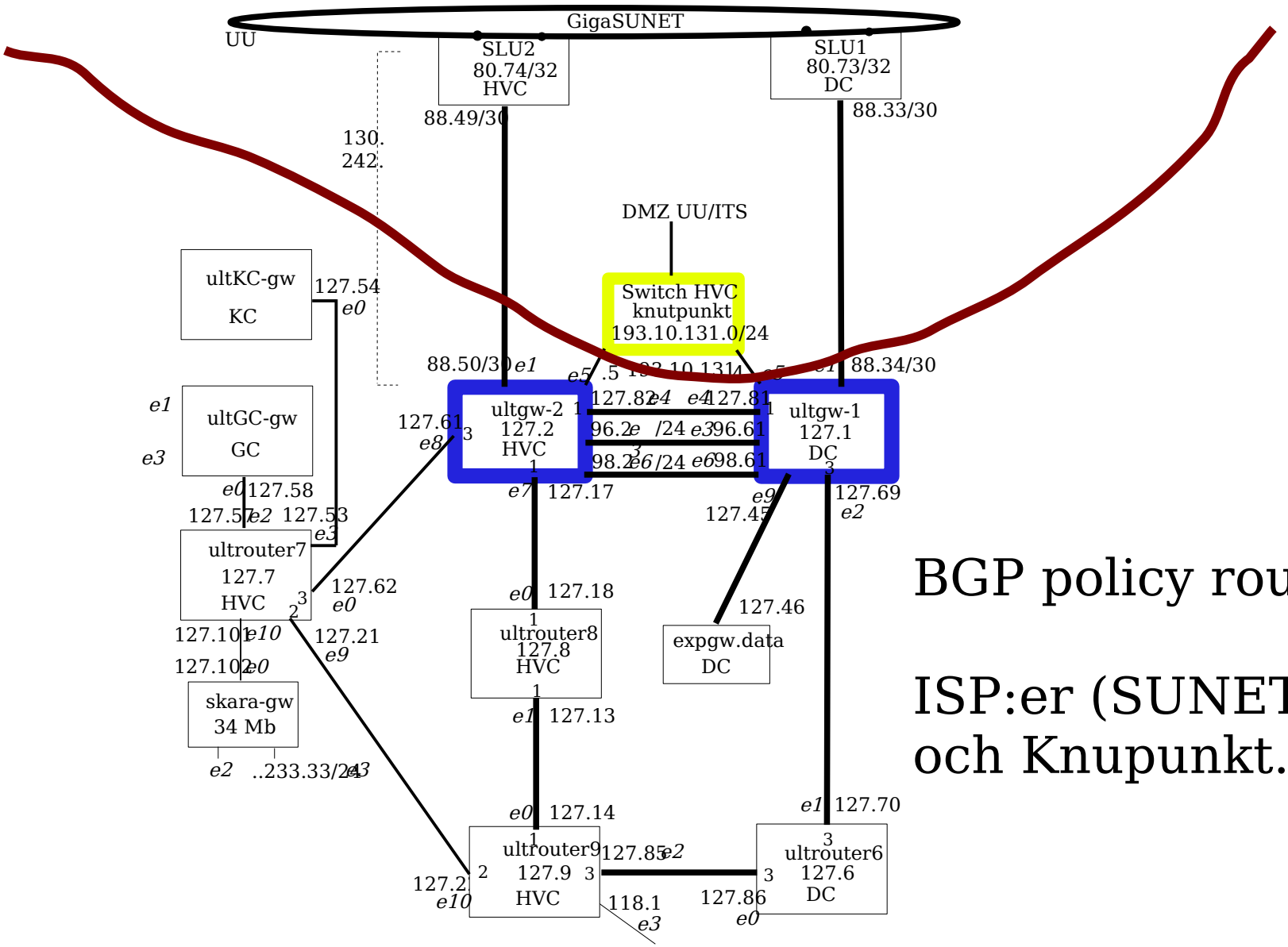
Trash datastucture

Very flat(fast) trees



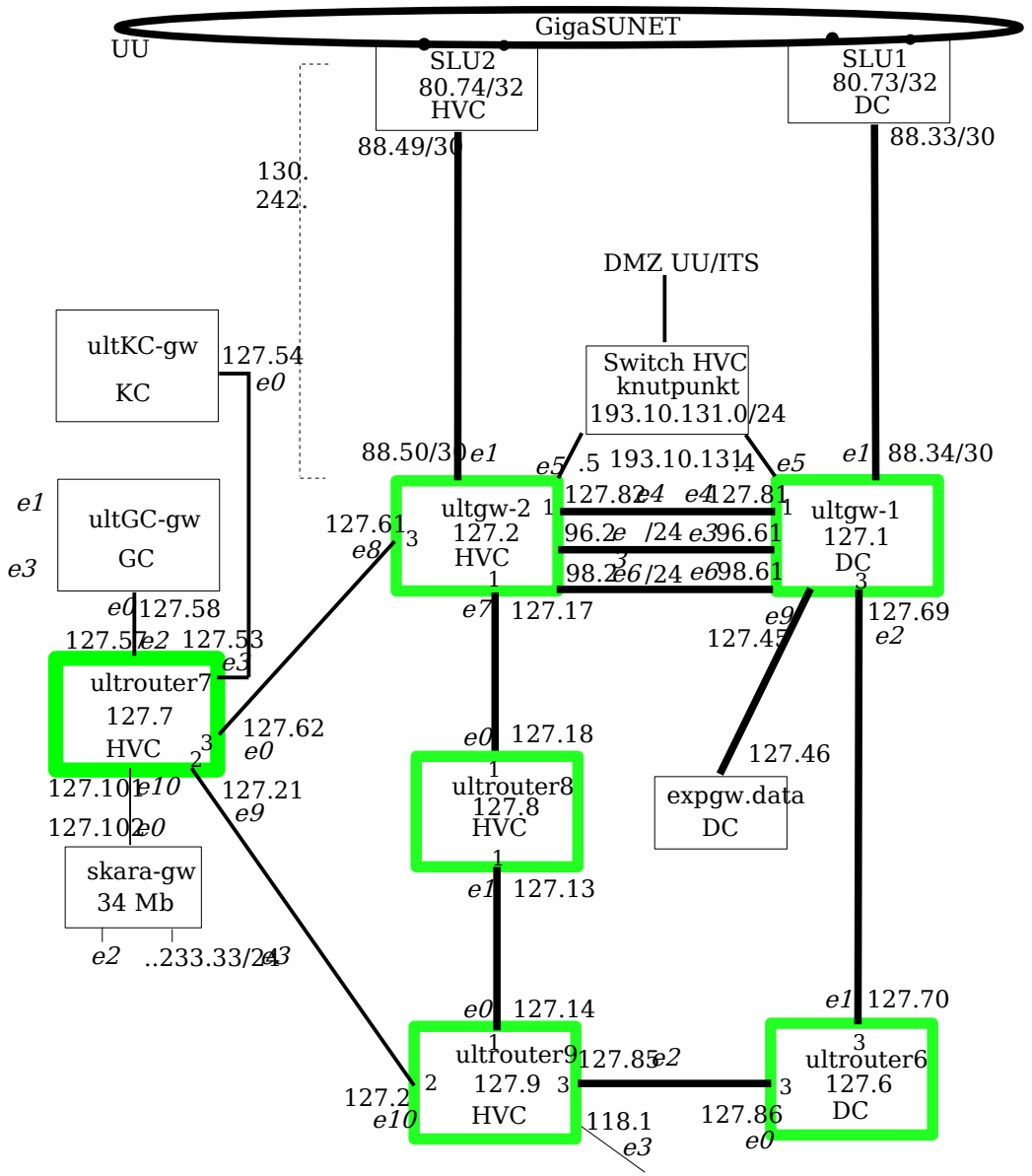


SLU's nät (inte hela)

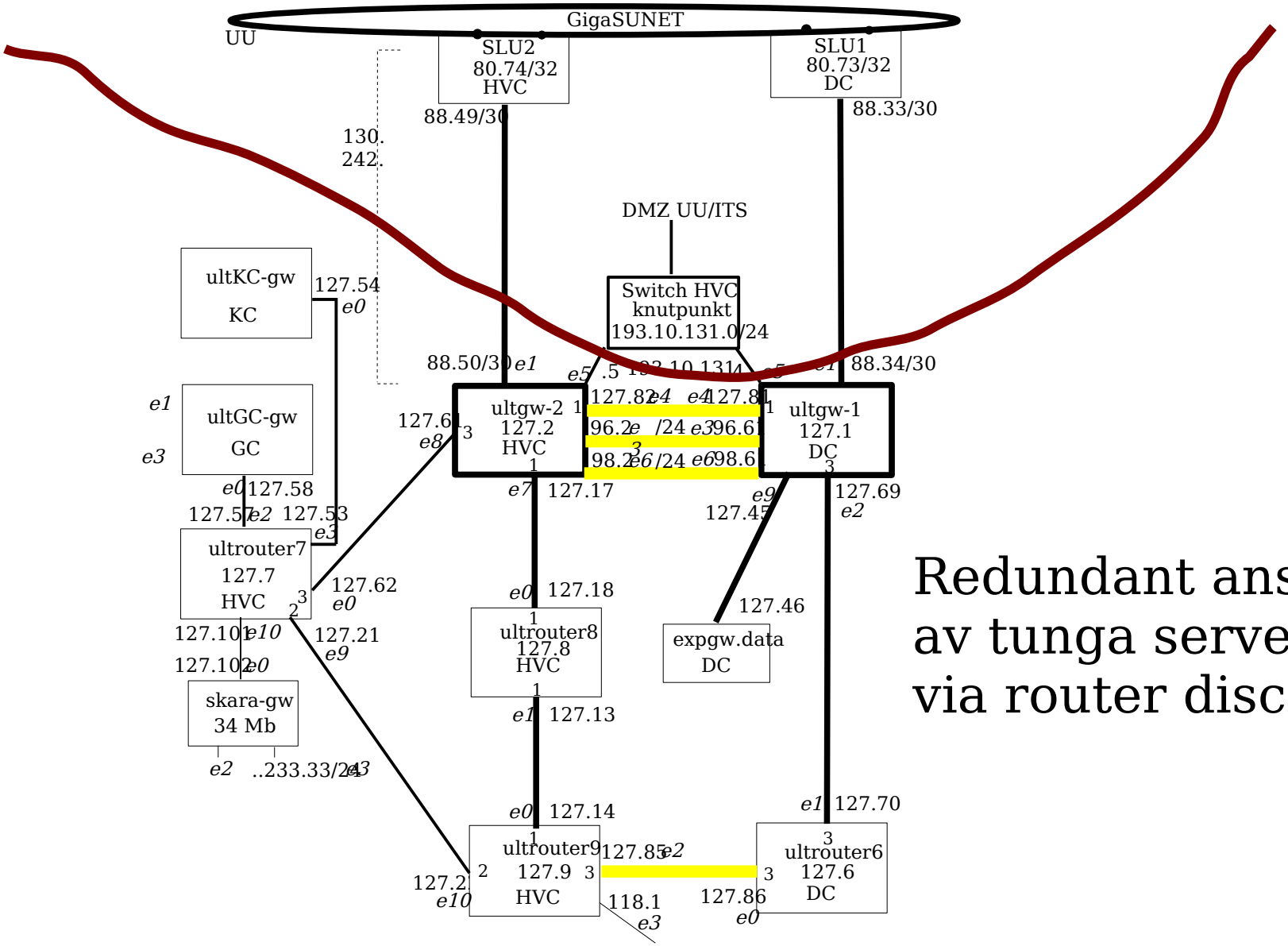


BGP policy routing

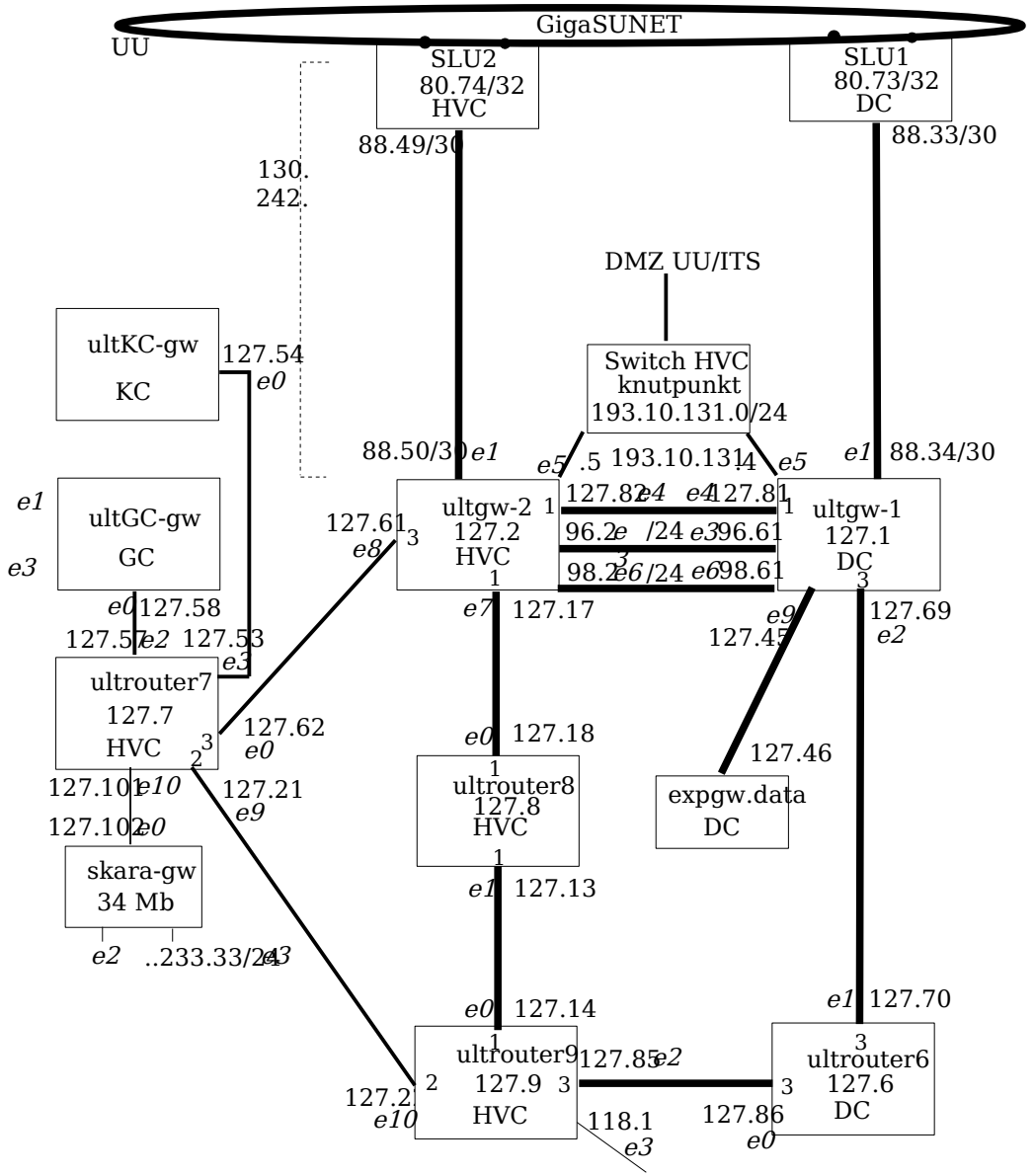
ISP:er (SUNET) och Knupunkt.



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
ultKC-gw
KC
127.54
e0
Switch HVC
knutpunkt
193.10.131.0/24
88.50/30
e1
e5
.5
193.10.131.4
e5
e1
88.34/30
e1
ultGC-gw
GC
e3
127.61
e8
ultgw-2
127.2
HVC
127.82
e4
e4
127.81
96.2
e
/24
e3
96.61
98.2
e6
/24
e6
98.61
ultgw-1
127.1
DC
e0
127.58
127.57
e2
127.53
e3
e7
127.17
e9
127.45
127.69
e2
ultrouter7
127.7
HVC
127.62
e0
e0
127.18
127.46
127.101
e10
127.21
e9
ultrouter8
127.8
HVC
expgw.data
DC
127.102
e0
skara-gw
34 Mb
e1
127.13
e2
..233.33/24
e3
e0
127.14
e1
127.70
ultrouter9
127.9
HVC
127.85
e2
ultrouter6
127.6
DC
127.2
e10
118.1
e3
127.86
e0

Almost all...

OpenSource
bifrost
zebra/quagga