Netconf 2005

Robert Olsson

Experiments & Experiences
with FIB lookup and route cache

What we hear/got

dst cache overflow reports
- RCU related
- mistuned, misunderstod etc.

fib_lookup complaints
- what to expect
- BSD comparisons. Radix-tree
- ToS/semantic questionable
- fib_hash considered bad

Getting forward :)

“Infrastructure” for test & development

stats to understand what happens
tools and setups to study

Preroute patches w.. Jamal 2004
pktgen DoS, scripts w. routing table

steady Linux API work to prepare to
plugin new algos. Most from DaveM.

So much research
Still so little usable for Linux

FIB overview

FIB vs. dst hash performance

fib_hash
fib_hlist
fib_hash2
fib_trie
classifier lookup?
unified lookup?

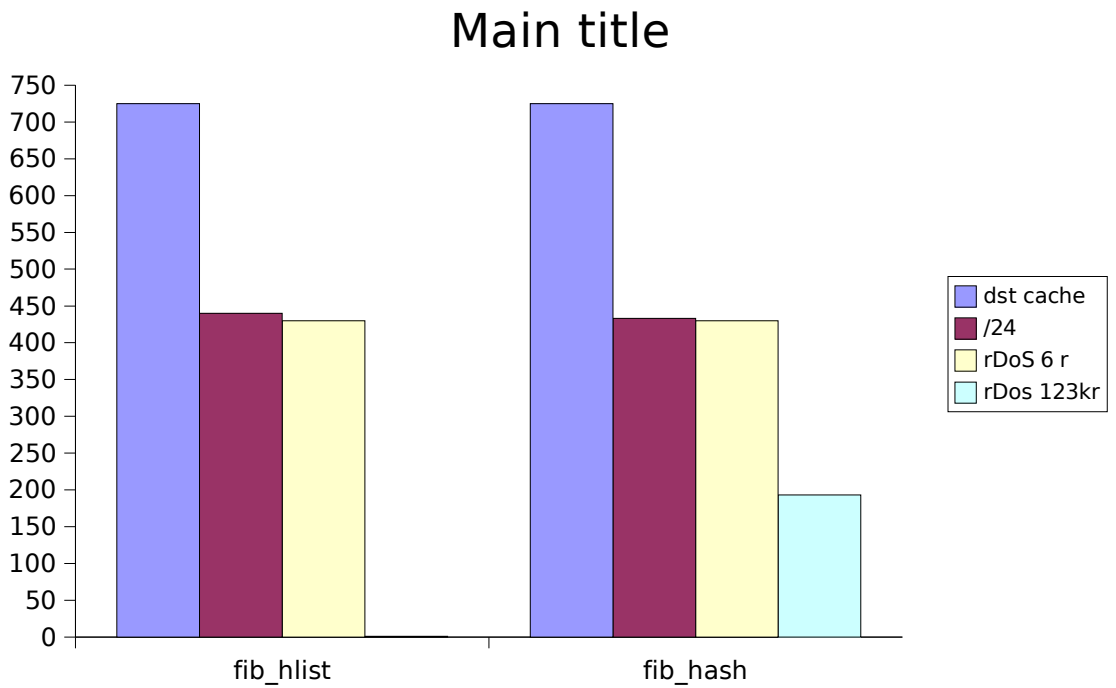
fib_hash (current)

Fast - Yes
General purpose
Very integrated

fib_hlist

- Tutorial
  - KISS
  - hlist with semantic_match
  - Very fast with small tables
  - For embedded system etc?

fib_hlist performance



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

fib_hash2

Vargese inspired, use what got

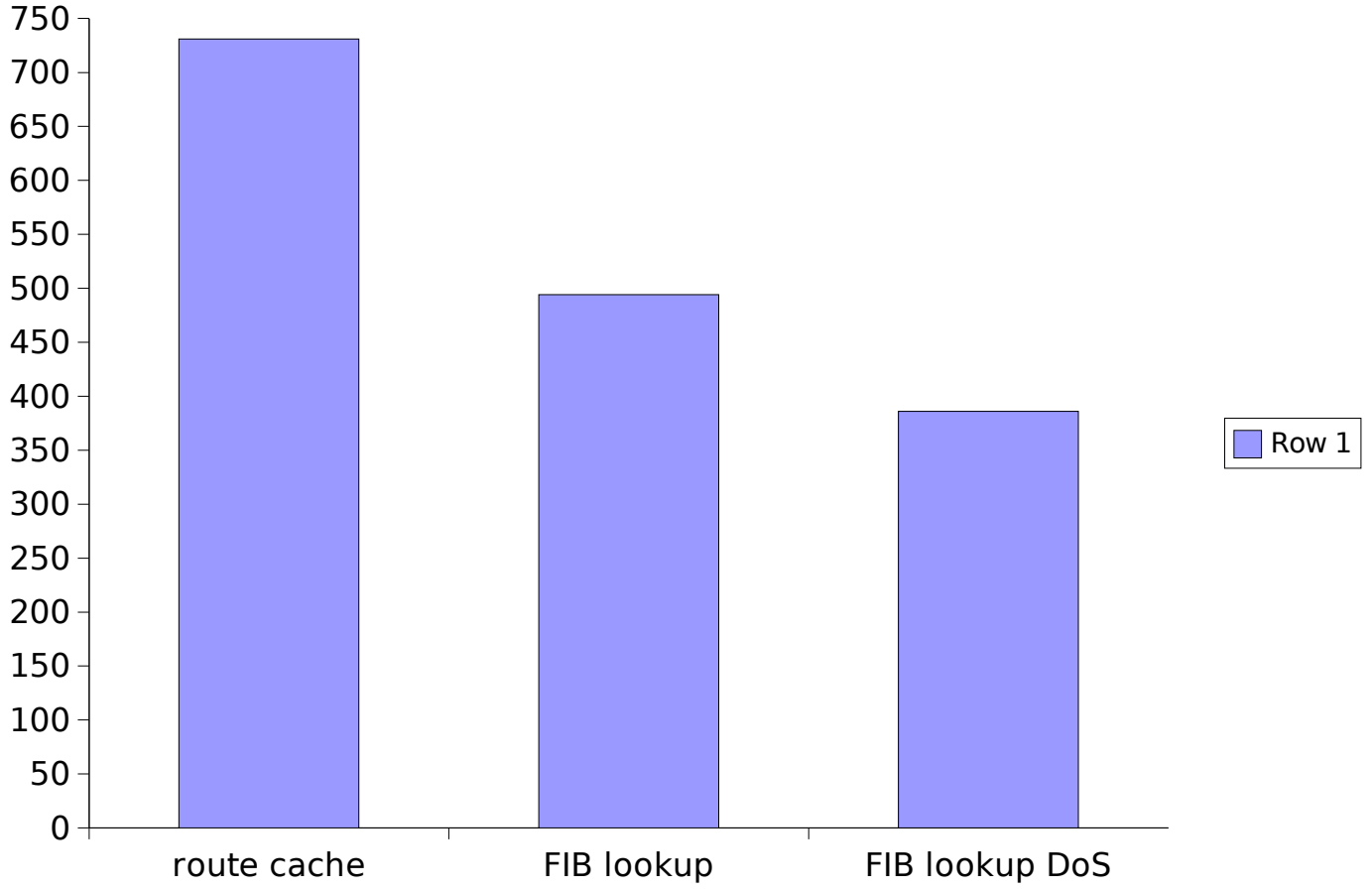
2^24 hash lookup w. sorted hlist
Makes /24 entries of plens 1-23
/0 special case. Huge...
TABLE_LOCAL with a few entries

Idea was to test performance
with the fastest algo we could think of.

Not for embedded system etc? :-)

Reduced it became fib_hlist

fib_hash2/route cache compare



fib_trie

First trie. In theory variable key length, 32, 128 bits etc

Algo for dynamic trie written in Java. Memory leak and stack handling were problems.

Also prefix matching based on fib_sematic match

Cisco CEF has fixed 256 childs 8-8-8-8 or 16-8-8 (GSR)
LC-trie is child size is dynamic 2-12 bits seen

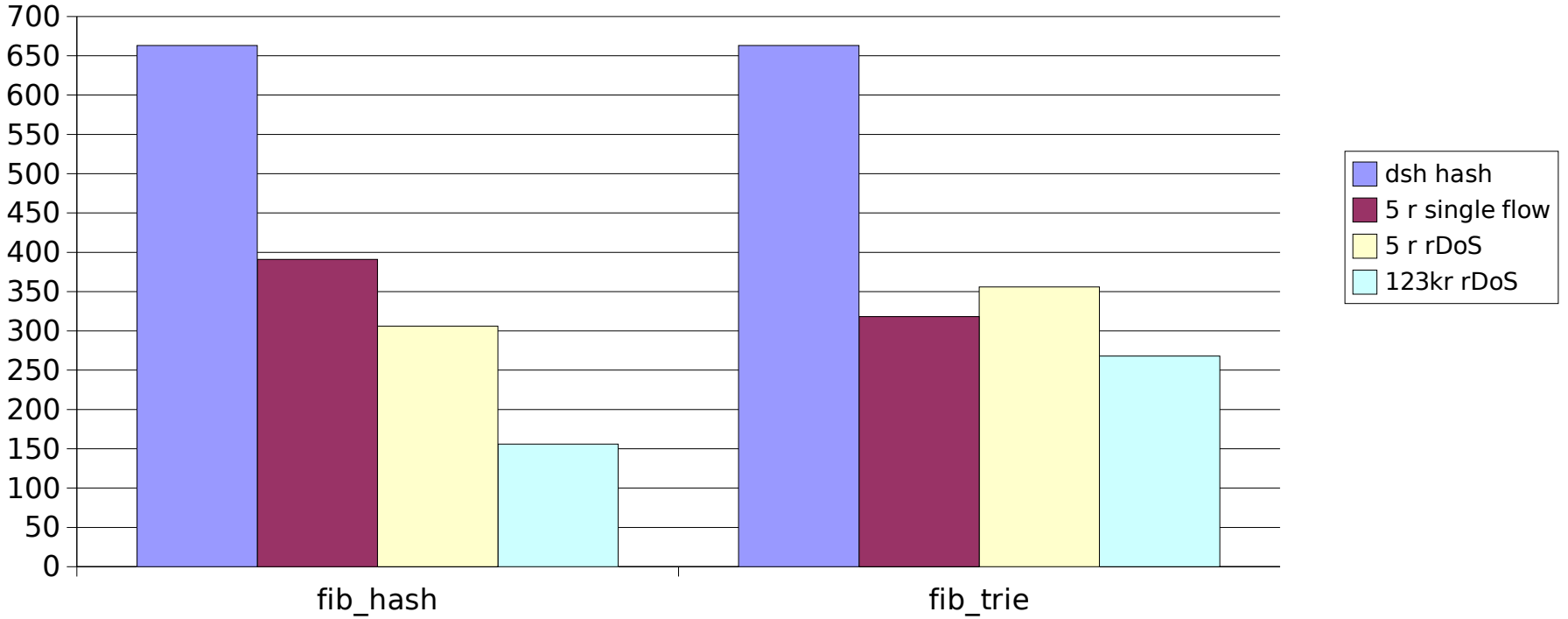
Need to be verified. New netlink call to do fib_lookup

Can be improved...

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

LOCAL/MAIN tables

fib_lookup() in ip_fib.h

Always looks up LOCAL table before MAIN

Extra lookup costs performance when not to localhost.

We discussed this with Alexey...

LOCAL/MAIN tables

Aver depth: 4.48
Max depth: 6
Leaves: 25
Internal nodes: 18

Aver depth: 3.22
Max depth: 7
Leaves: 158936
Internal nodes: 39440

Route hash/GC

Strategies for GC run. Better work!!

Timer based vs on demand
/proc/sys/net/ipv4/route/gc_interval
/proc/sys/net/ipv4/route/gc_min_interval_ms

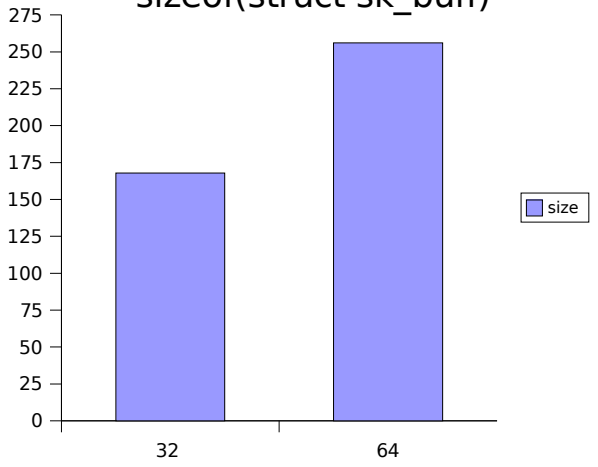
GC without GC run. Very robust...

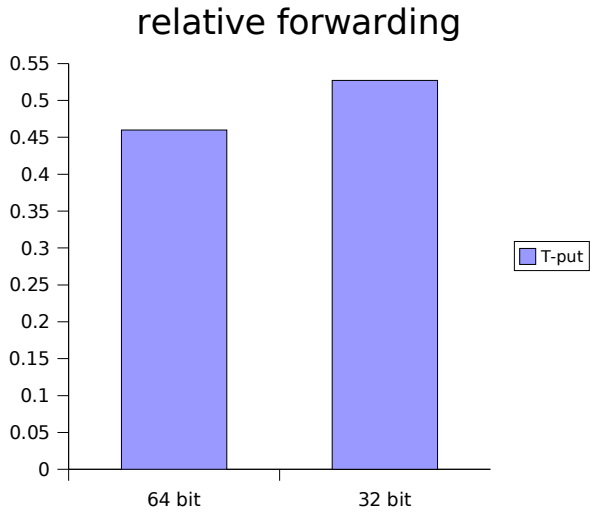
rt_intern_hash() cand rt_free() chain length to long.

ip_rt_gc_elasticity can be dynamic.... ????

total flush for fib insert/delete....

32/64 bit || sizeof(sk_buff)





Gcc 3.4 x86_64 vs i686 on same HW

Per device hash

Per device input route hash

isolate dev's
less locking
same performance

output used shared hash

given up for the moment

Preroute patches

Started hacking with Jamal a year ago

Do full fib_lookup() for every packet

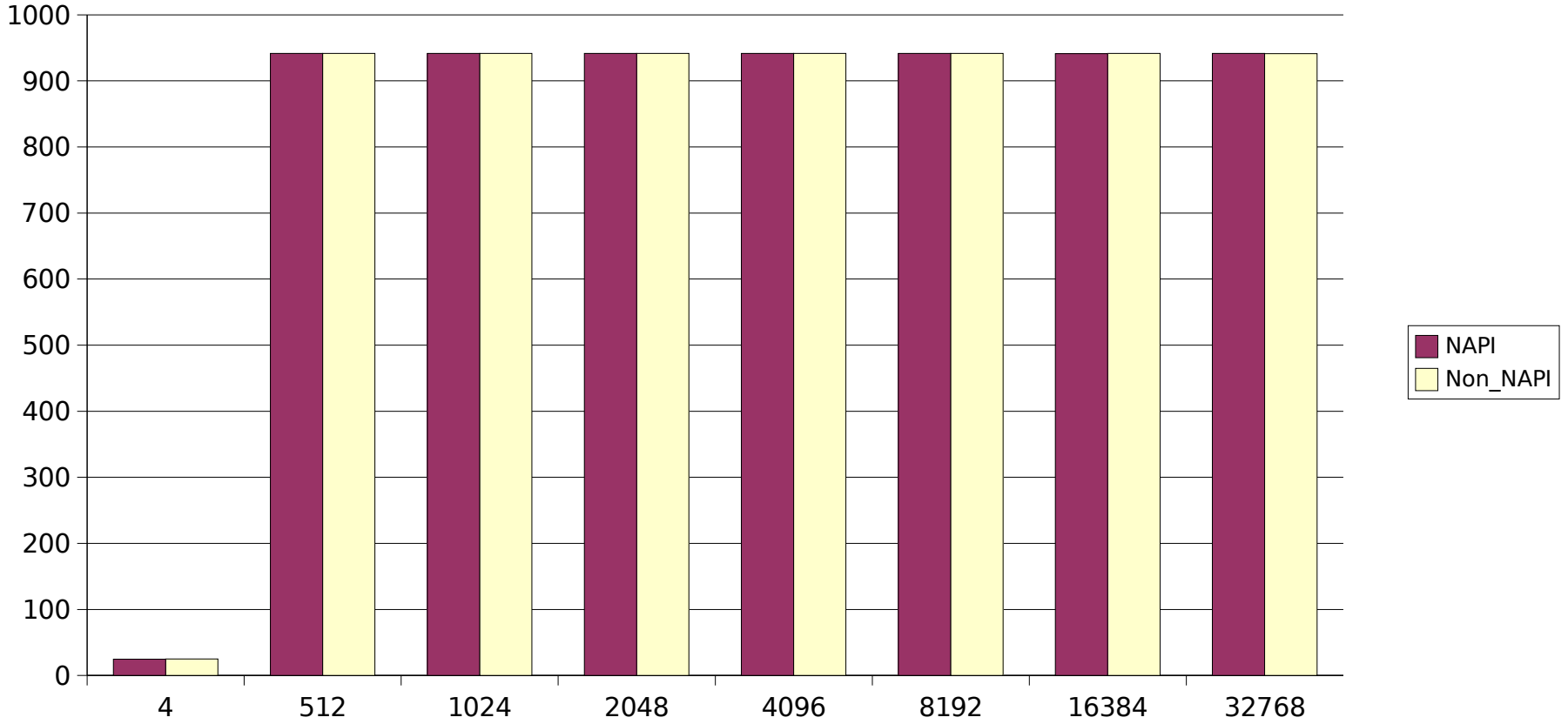
Lot's of interest from Paul's and people doing “hi-risk” hosting.

Very useful for FIB testing.

Works only with gatewayed hosts.

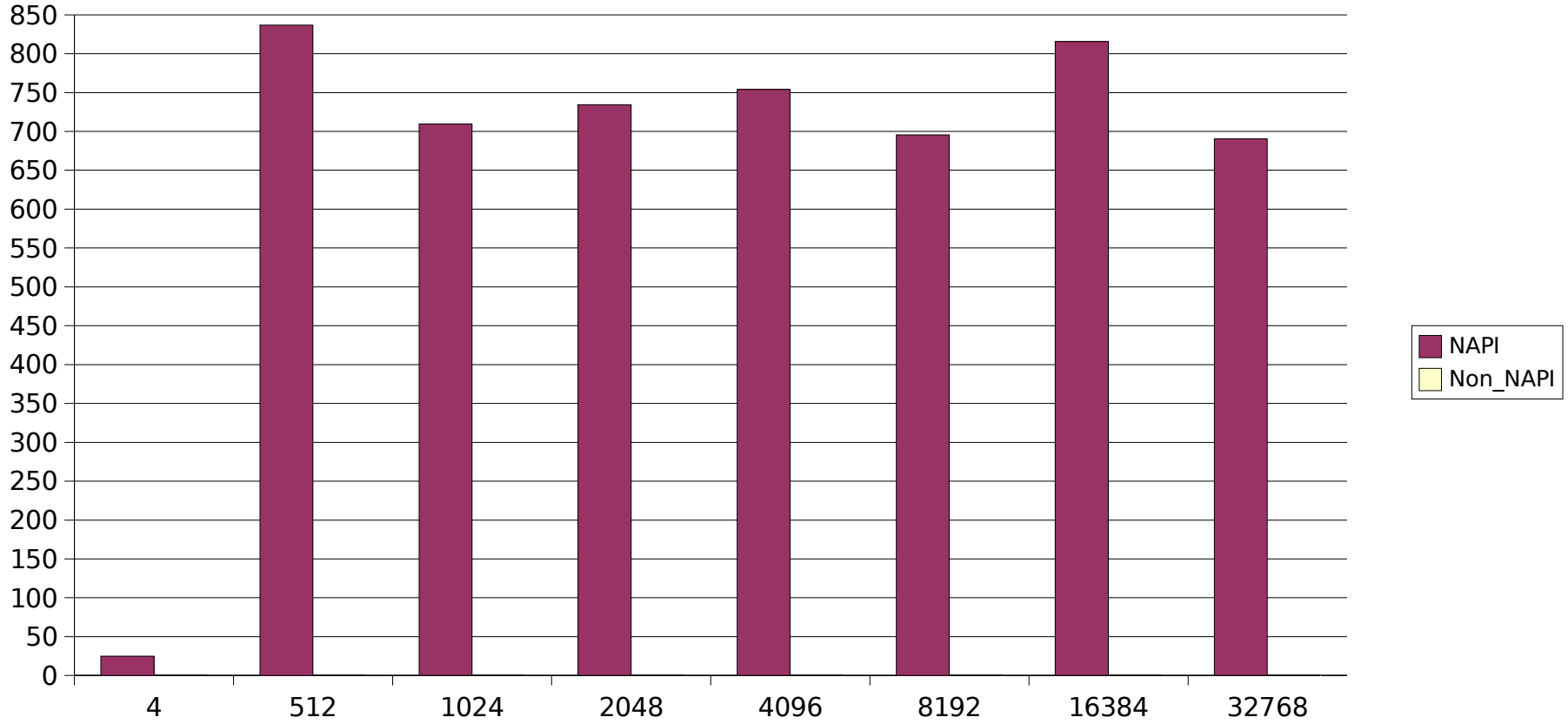
Skb recycling/reuse

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

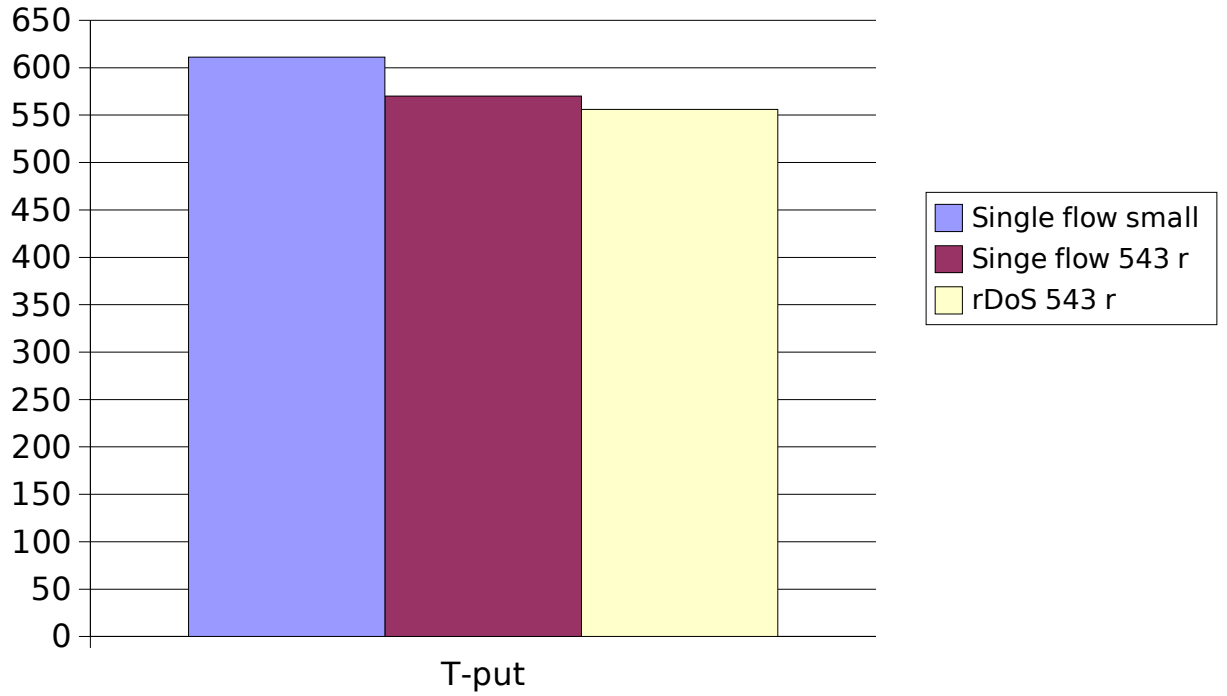


2.6.11.7 SMP kernel using one CPU driver e1000 NAPI - no-NAPI. Opteron 1.6 GHz e1000 w 82546GB.

ipv6 performance

Forwarding kpps 76 byte pkt.

Linux 2.5.12 1 CPU(SMP) Opteron 1.6 GHz e1000

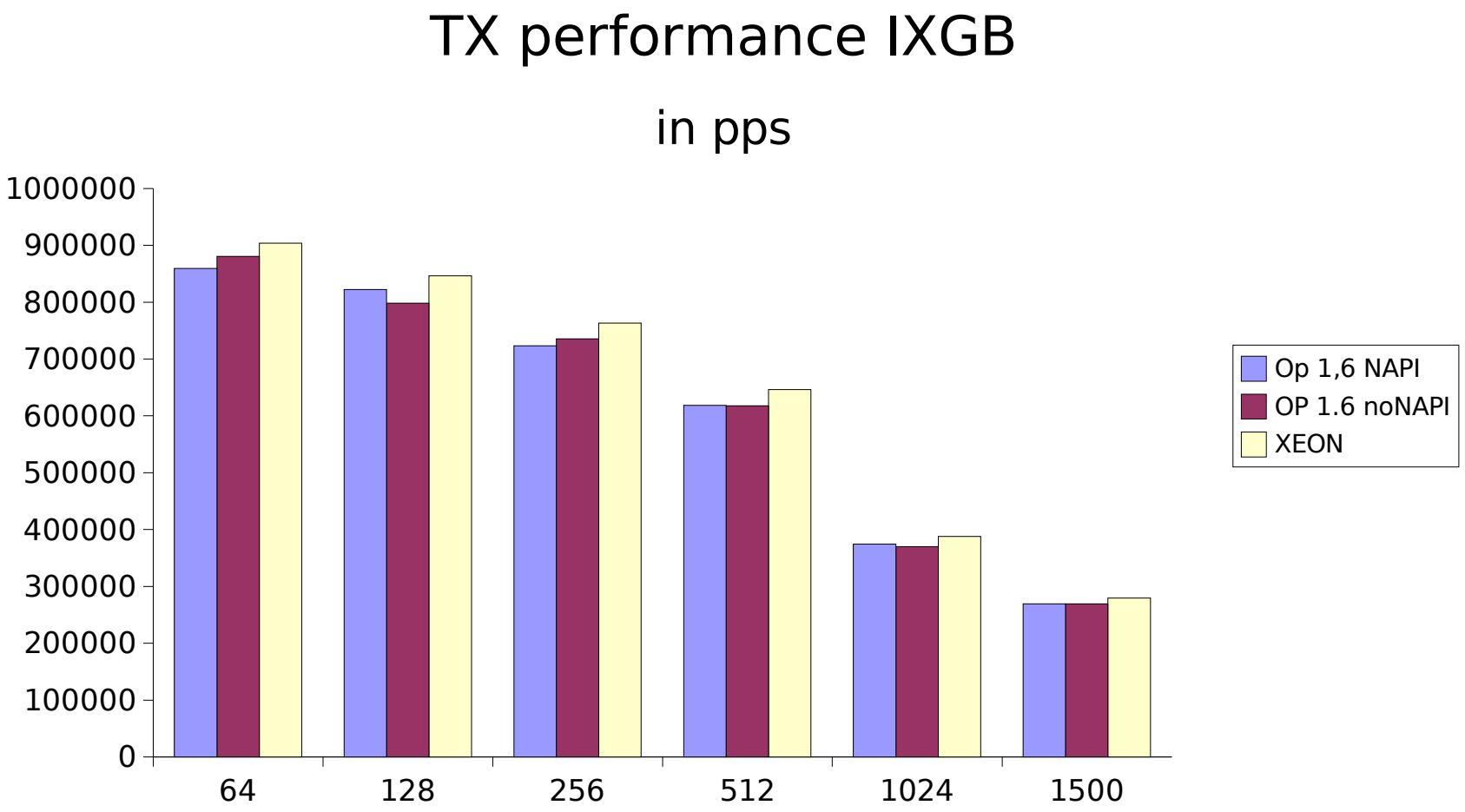


How rDoS work on sparse routing table?

Goodbye to old friends?

FASTROUTE
HW-FLOWCONTROL

10 GbE early days



Hi-perf filtering

Need for hi-pref stateless filtering netfilter API

hi-pac?

tc-stuff?

netfilter API

share fib_semantic_match()