Lightweight IPv6 and IPv4 traffic offloading using IPv6 neighbor discovery

WiBrA Workshop

15-Oct-2012

Jouni Korhonen

Background

- Original idea: use native IPv6 Neighbor Discovery (ND) for network controlled offloading of traffic:
  - Network tells preferences using standardized means, which every compliant end host should then adhere.
  - ND is mandatory to implement in 3GPP.
  - Host does the hard work i.e. selects what packets go where – which also then works for encrypted traffic.
  - Does not guarantee 100% correct decision every time but majority of cases are ok. Everything works as long as there are no walled gardens.

- Basic work & PoC was done spring/summer 2010.

Basic principle

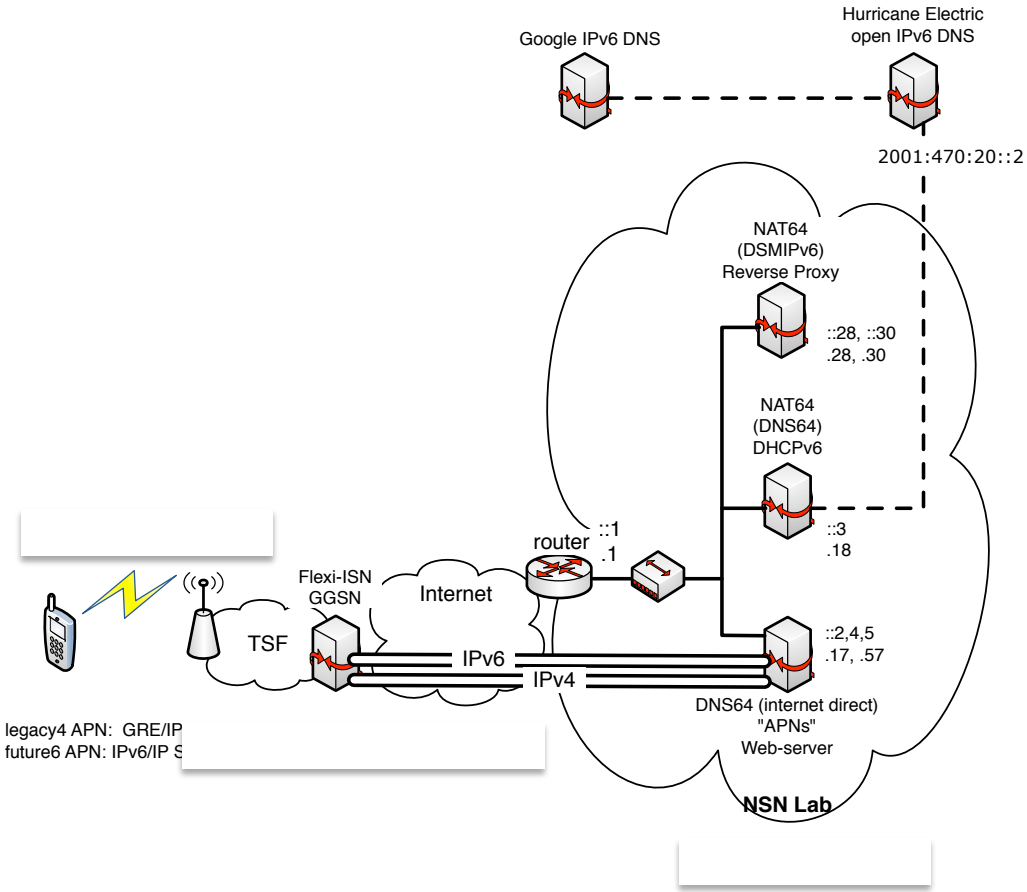
- RFC 4191 defines ‘default router preferences and more specific routes’ extension to neighbor discovery :
  - End host implementations for Windows since XP, Linux including Androids(!), *BSD, etc.. code is out there already.

- Allows a GGSN/PGW to say simple rules e.g.,:
  - “I am the lowest preferred router. Use anything else except me **if possible**” -> e.g., pickup WLAN.
  - “I still want to **get traffic** going to these prefixes”.

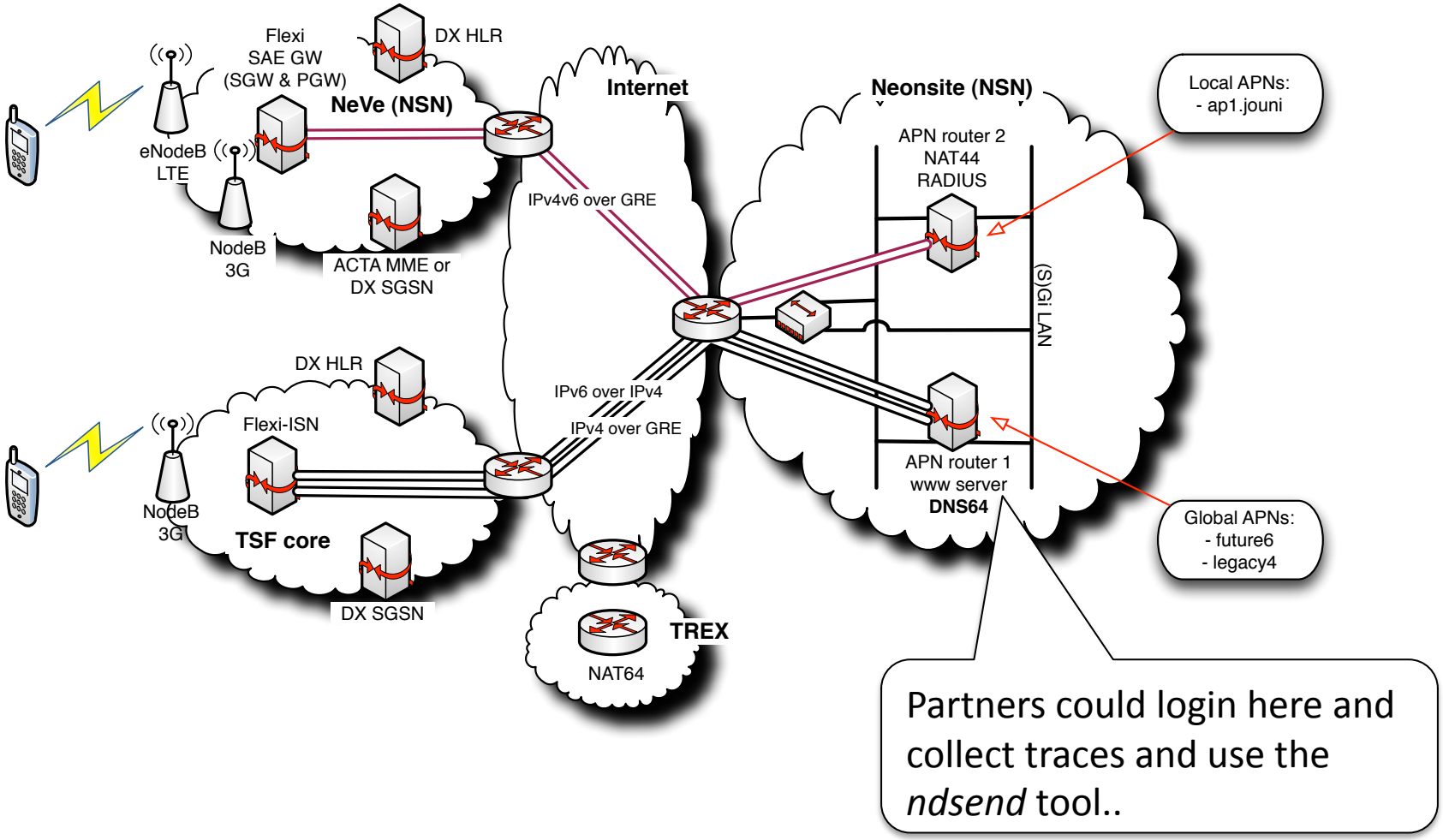
- Has a **push model** – the GGSN/PGW can initiate a Router Advertisement at any time.
- Not meant for offloading but works just fine!
- Natural to use 3GPP link as the **only source** for rules!

Test network since summer 2010 – Courtesy of a helping operator partner

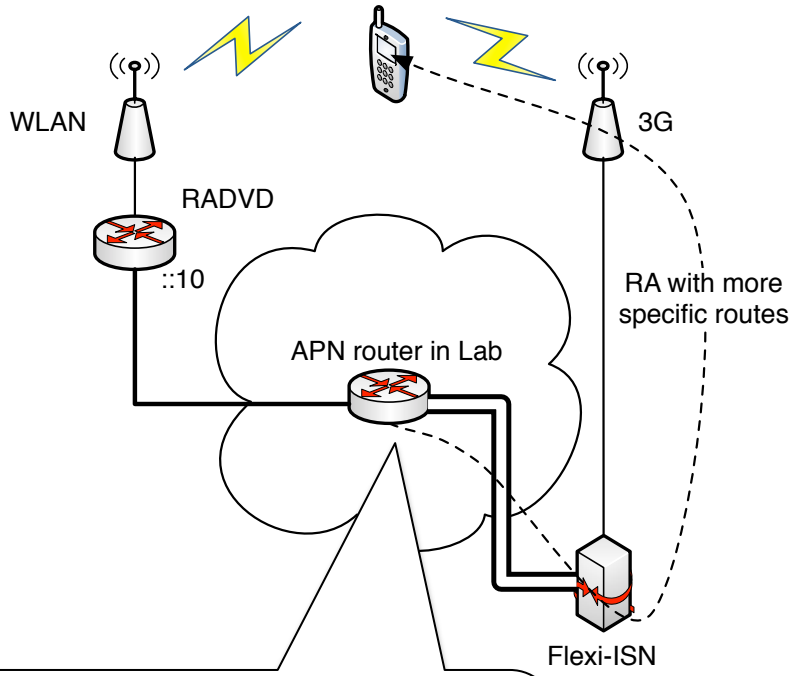
- A live network GPRS access and a bag of SIM cards (with roaming enabled).
- Normal transition gear:
  - NAT64 and DNS64 (open source obviously).
- Public IPv4 and IPv6.
- Best 3G dongles ever from Nokia.
- All partners (and some additional folks outside the project had access).



And test network evolved to this..



Basic IPv6-only offloading..



Project implemented *ndsend* tool that allows sending RAs and NAs to the mobile device.. spoofing to be the Flexi-ISN..

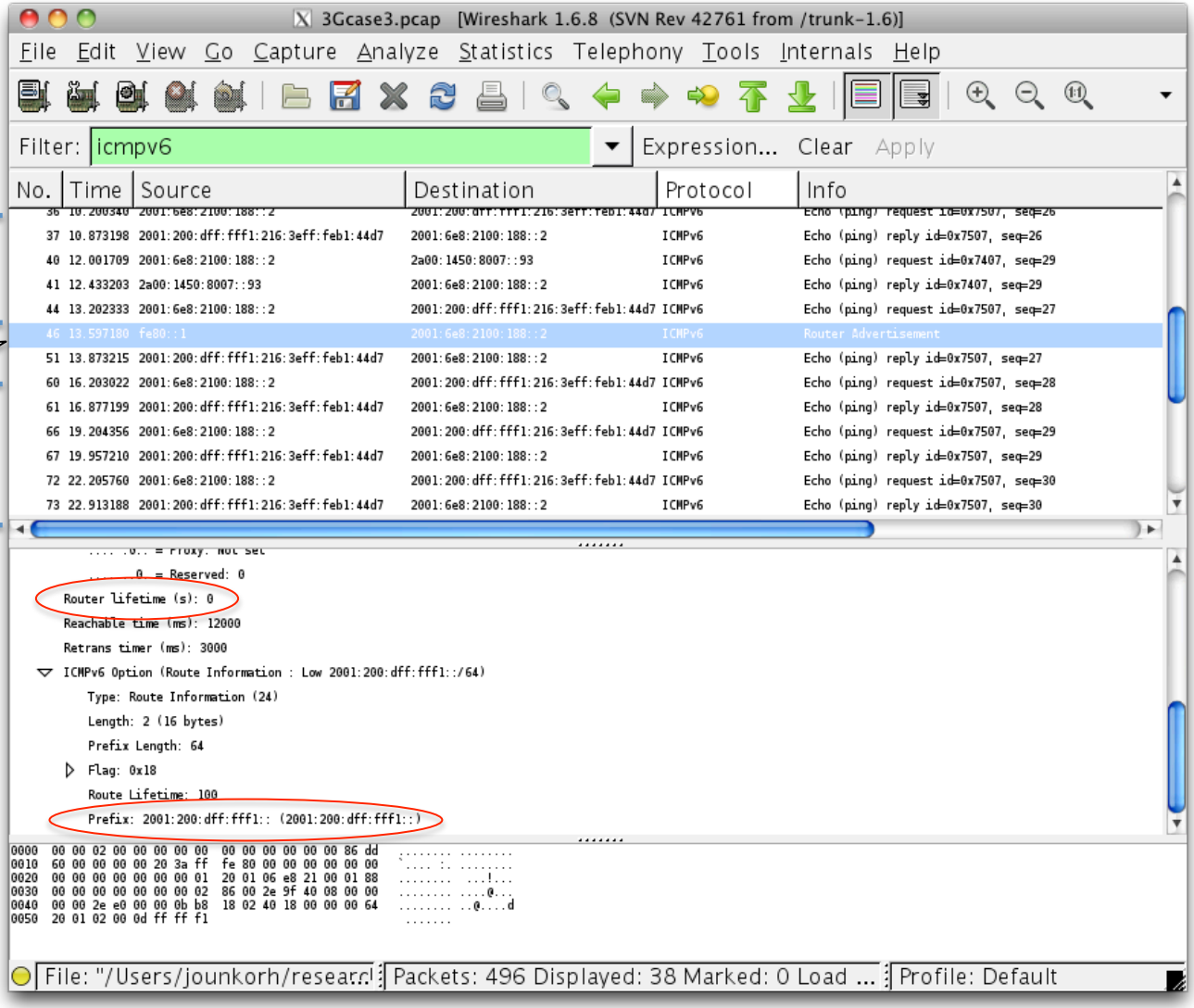
- Traffic steering affects only new connections i.e. there is no mobility and established connections remain unaffected.
- The GGSN may also remove itself from default routers and then no traffic except those with more specific routes are routed to it. (next slide)

ping6 www.google.com and www.kame.net and switch between 3G and WLAN

Both google and kame are ping6:ed happily over 3G

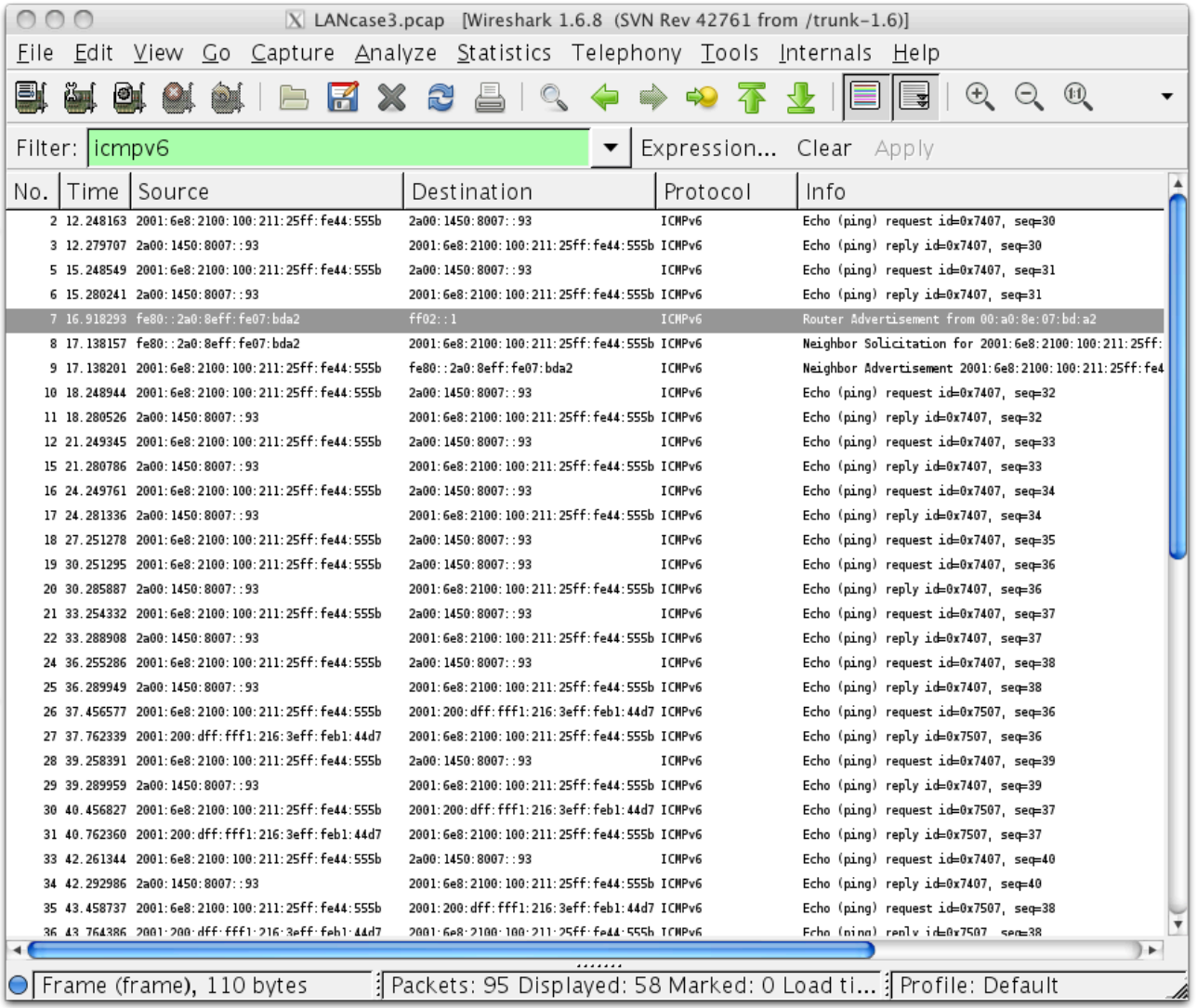
RA from GGSN:
- Removes itself from default routers (NO traffic will be routed to in unless..)
- GGSN adds a more specific route for kame

Google's traffic moves to other interface (WLAN)



ping6 www.google.com and www.kame.net and switch between 3G and WLAN

Google's traffic appeared to WLAN interface..

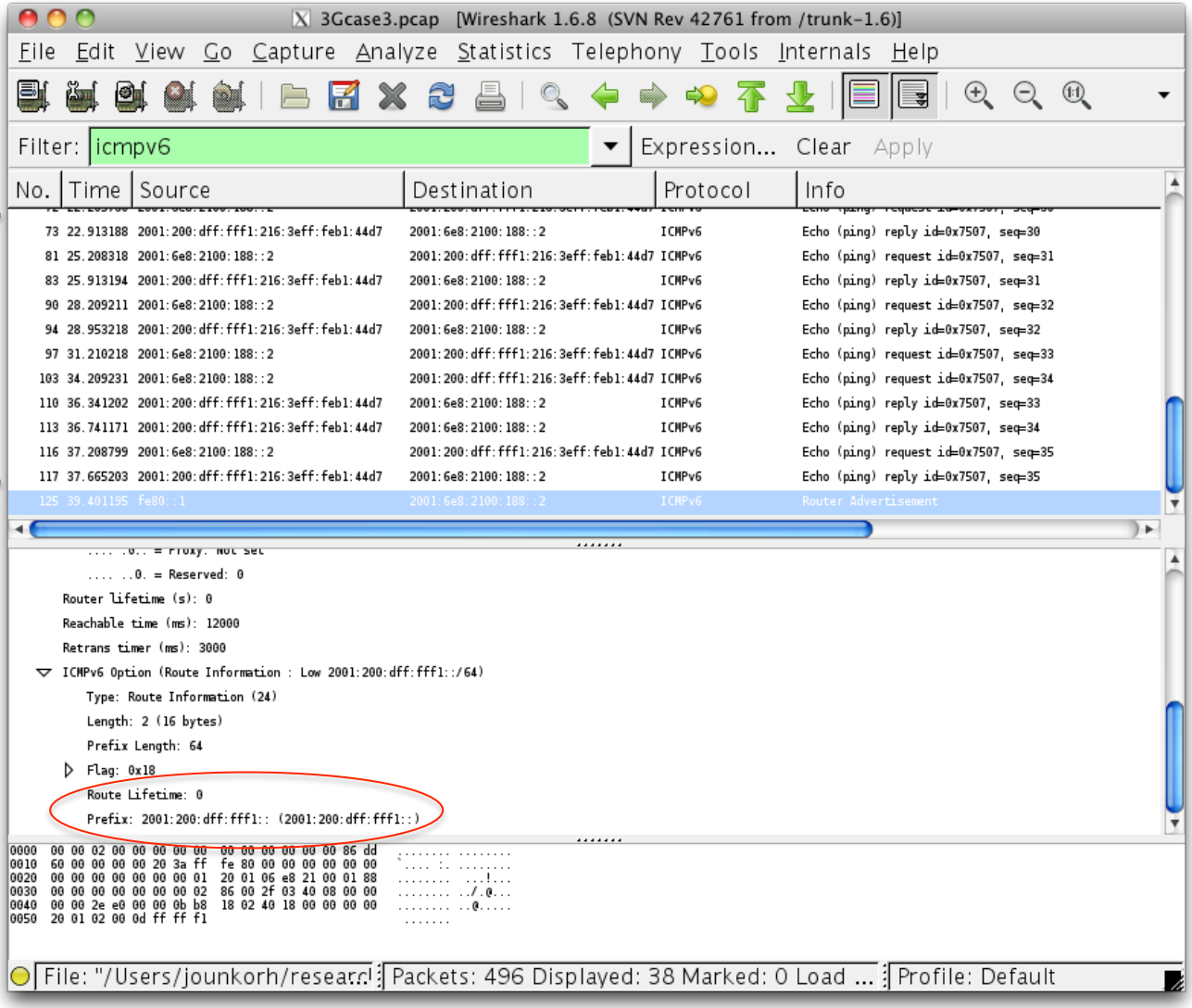


ping6 www.google.com and www.kame.net and switch between 3G and WLAN

Only kame is over 3G.. google moved to WLAN..

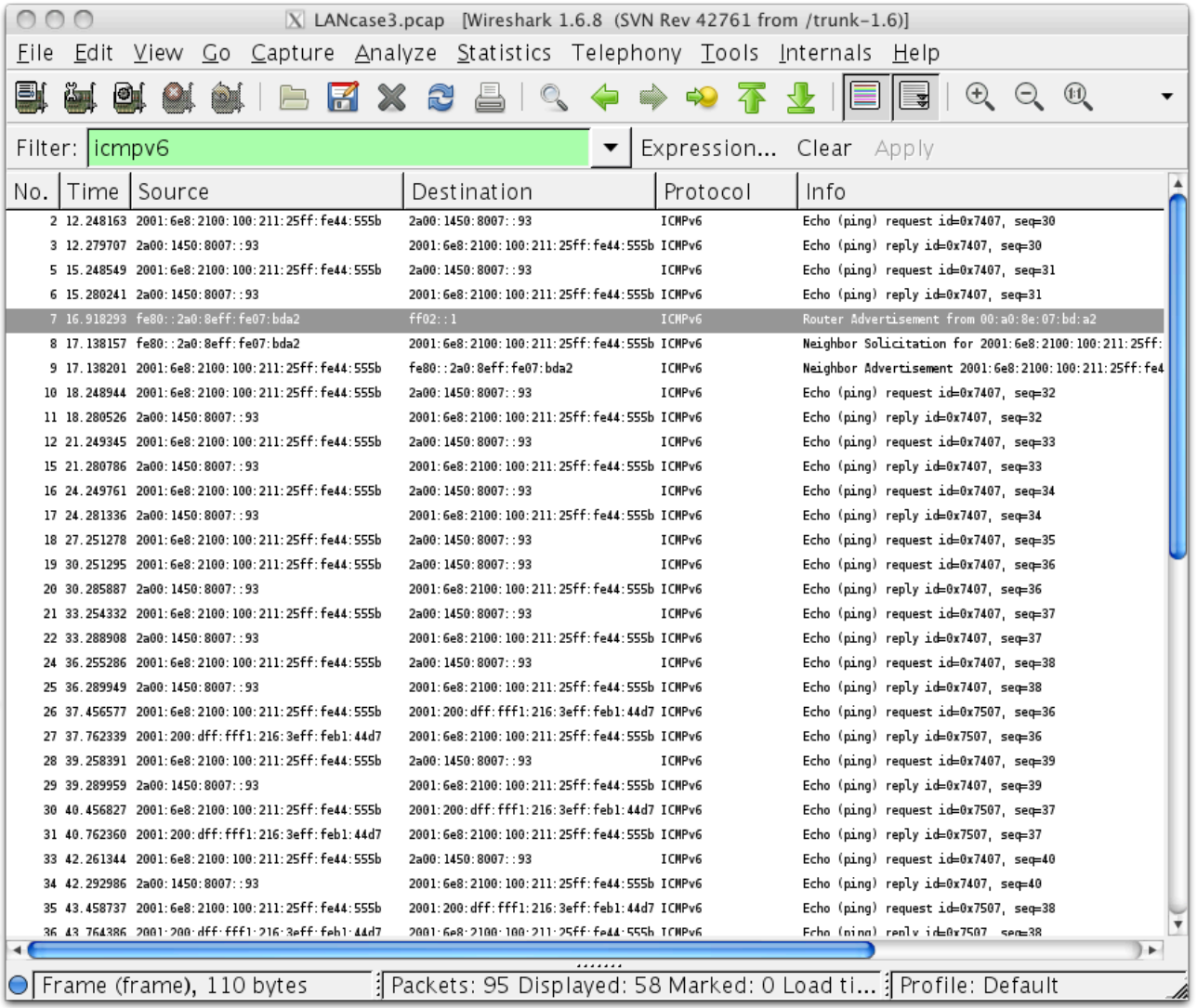
RA from GGSN:
• Removes the more specific route for kame

All traffic moves to WLAN. GGSN is not a default router and all new connections go to other accesses..



ping6 www.google.com and www.kame.net and switch between 3G and WLAN

Now also Kame's traffic appeared to WLAN interface.. both google & kame are over WLAN.



Issues – what next

- That worked as expected with unmodified end host. All we did was end host configuration.
- Alternatively GGSN could have lowered its preference to LOW and have the same effect but still keeping itself as a default router.
  - We do not accept RFC 4191 from WLAN so its default router is implicitly always has a MEDIUM preference.

- BUT! This works only for IPv6! WHAT ABOUT IPv4?
  - You can specify IPv4-mapped IPv6 routes in RFC 4191 but those routes get installed into IPv6 routing table not IPv4.

Implement RFC 4191 equivalent for IPv4 and piggyback it on the Neighbor Discovery

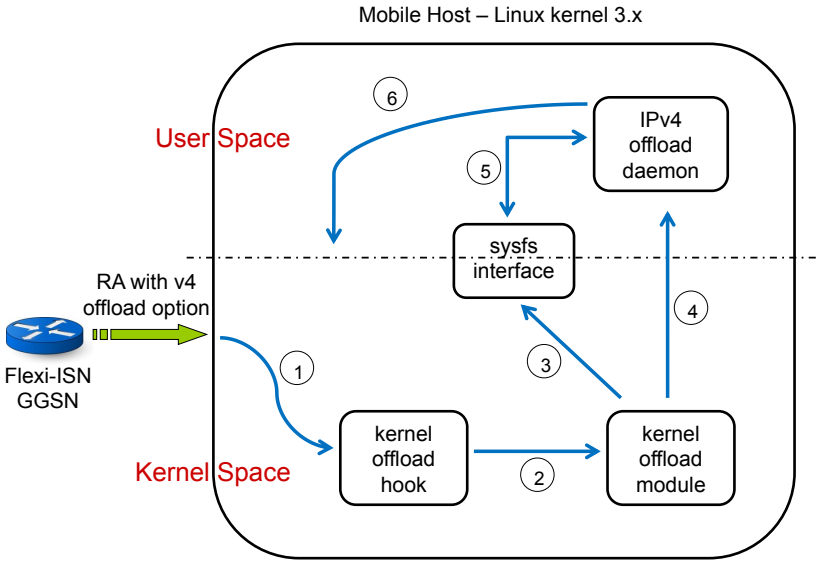
- Solution described in:
  - J. Korhonen, T. Savolainen, A. Y. Ding, "Controlling Traffic Offloading Using Neighbor Discovery Protocol", Internet-Draft "draft-korhonen-mif-ra-offload-05.txt", August 31, 2012 (work in progress).
  - Aaron Yi Ding, Jouni Korhonen, Pan Hui, Teemu Savolainen, Sasu Tarkoma, Markku Kojo, "NAO: A Framework to Enable Efficient Mobile Offloading", In Proc. ACM/IFIP/USENIX International Middleware Conference - Middleware PDT Workshop, Lisbon, Portugal, December 2011.
  - Jouni Korhonen, Teemu Savolainen, Aaron Yi Ding, Markku Kojo, "Towards Network Controlled IP Traffic Offloading", IEEE Communication Magazine COMMAG. (submitted for review)

- Adds IPv4 more specific routes (old) using IPv4-Mapped IPv6 addresses or (new) as its own options; also adds IPv4 default gateway address, which must point to the sending router.

IPv6 and IPv4 offloading implementation

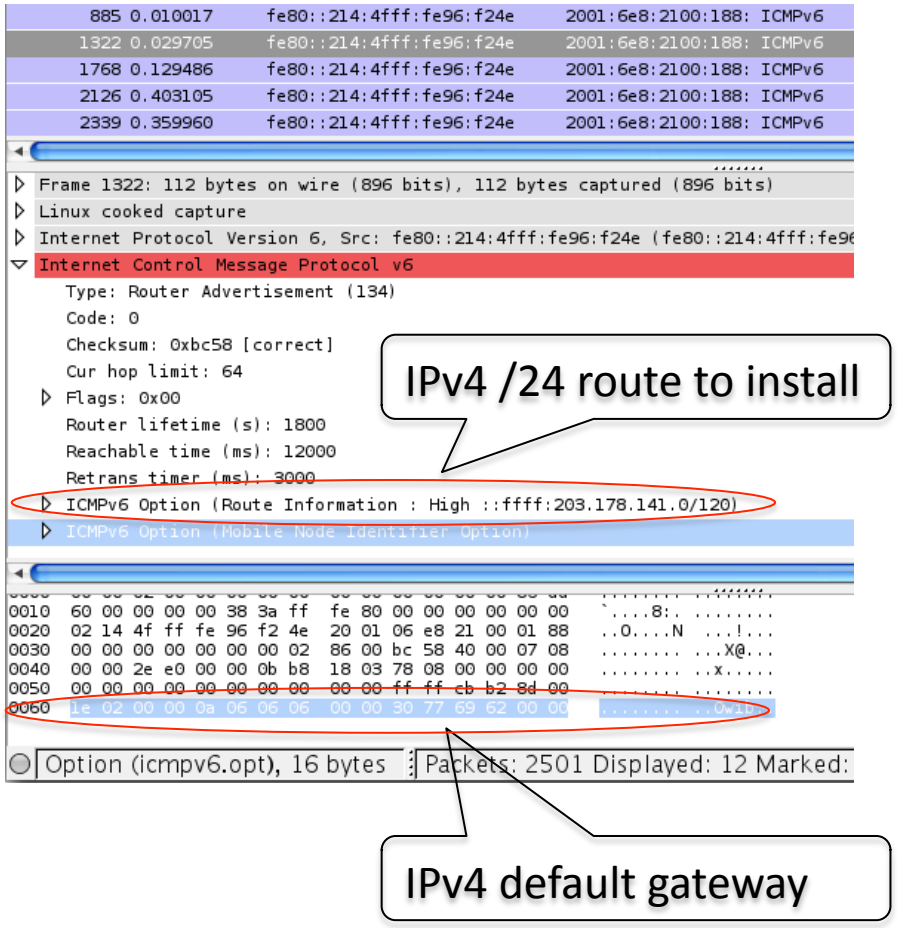
- Modifies the end host, which is generally bad.. but..

- Implementation mainly in user space with a kernel hook.
- Kernel pushes new RA options to (3) via sysfs to user space and/or (4) using netlink to the listening daemon (we could have just hook into ndisc_is_useropt() but this was a generic exercise anyway.. ;-)
- Daemon takes care of installing and handling IPv4 routes (scripting mostly..)



Test case and example of a RA

- For mixed a test sequence: 6 flows (pings)
  - IPv6/IPv4 to google.com, kame.net, funet.fi
- All traffic go over 3G (IPv6 and IPv4)
  - This is done by sending RAs to client, to add specific routes for each destination, lifetime 500s, preference as high (could be all in one RA) .
  - Router pref is MED but Linux prefers WLAN over 3G as a default.. to be sure router pref should be LOW.
- Move all IPv4 to WLAN and keep all IPv6 in 3G
  - 3 RAs to change the routes for IPv4 traffic, by setting lifetime to 0 -
- Move IPv6 funet.fi from 3G to WLAN
  - 1 RA to change the route to wlan, by setting lifetime to 0
- Move IPv6 kame.net from 3G to WLAN
  - 1 RA to change the route to wlan, by setting lifetime to 0
- Move IPv4 funet.fi back to 3G
  - 1 RA to add specific route to 3G, lifetime 500s, preference as high
- Now we should have the following:
  - IPv6 google.com over 3G
  - IPv6 kame.net and funet.fi over WLAN
  - IPv4 google.com and kame.net over WLAN
  - IPv4 funet.fi over 3G

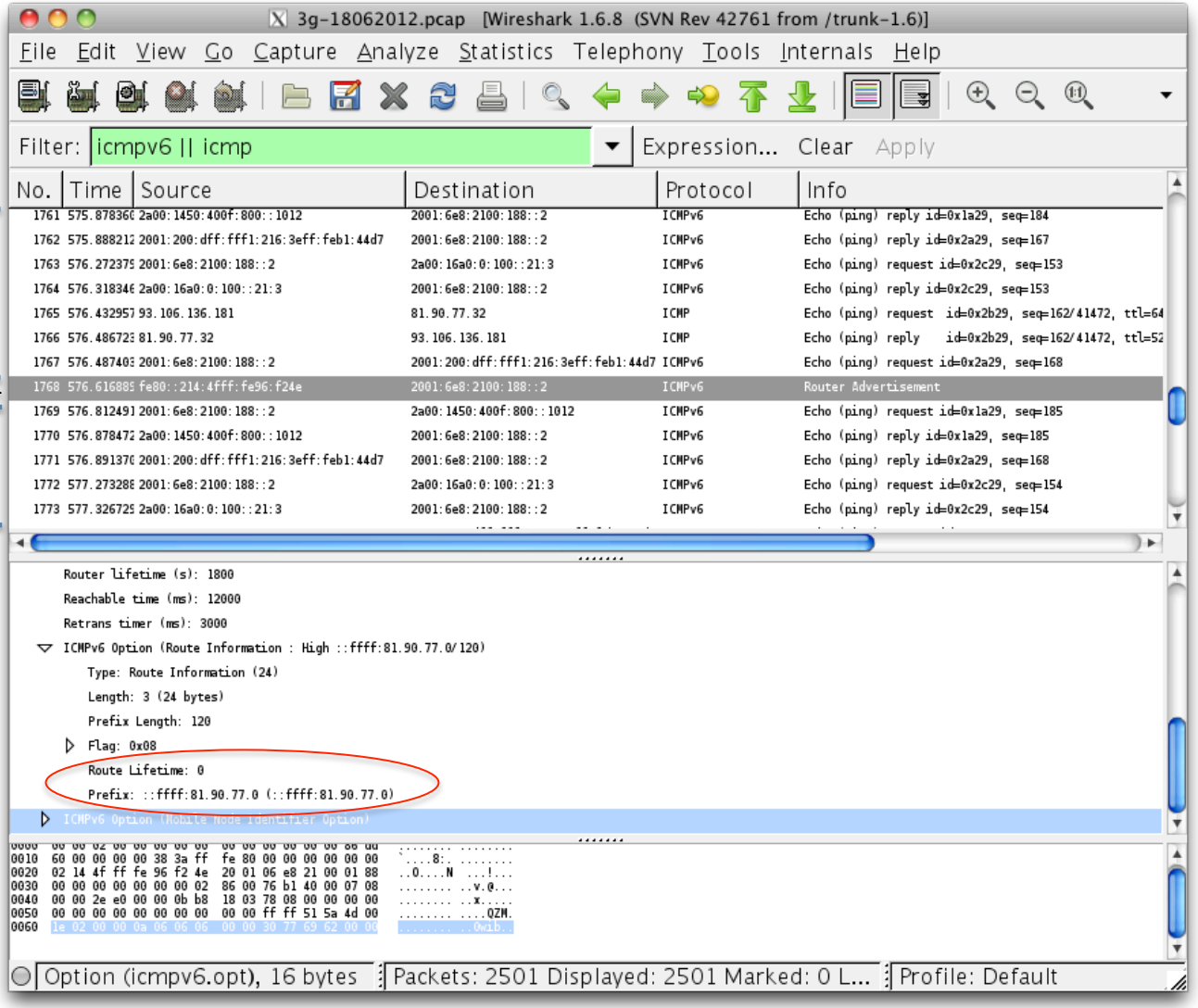


ping6 and switch between 3G and WLAN – move the last IPv4 flow to WLAN

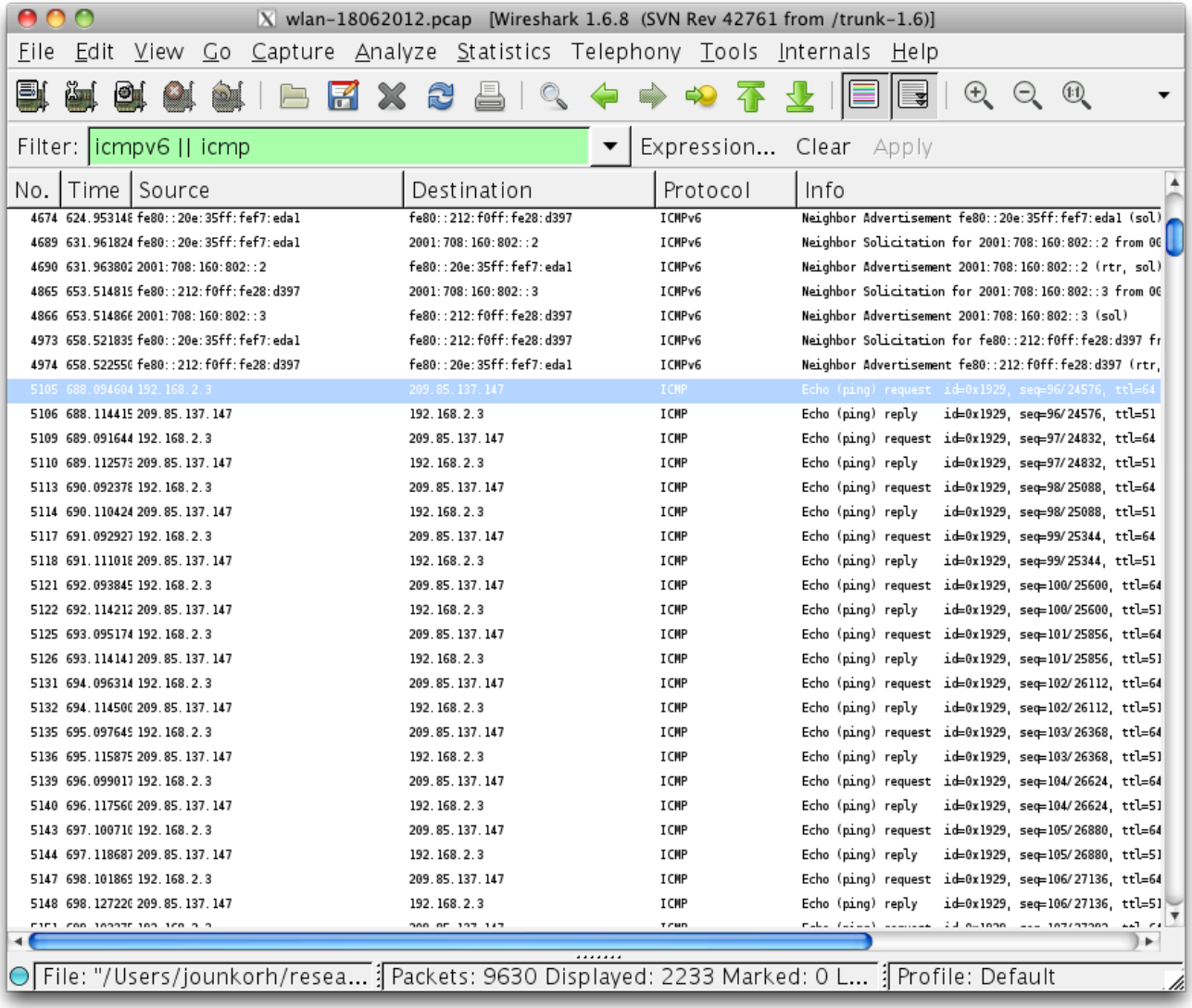
IPv6 google & kame over 3G.. IPv4 funet also over 3G

RA from GGSN:
• removes more specific route for www.funet.fi

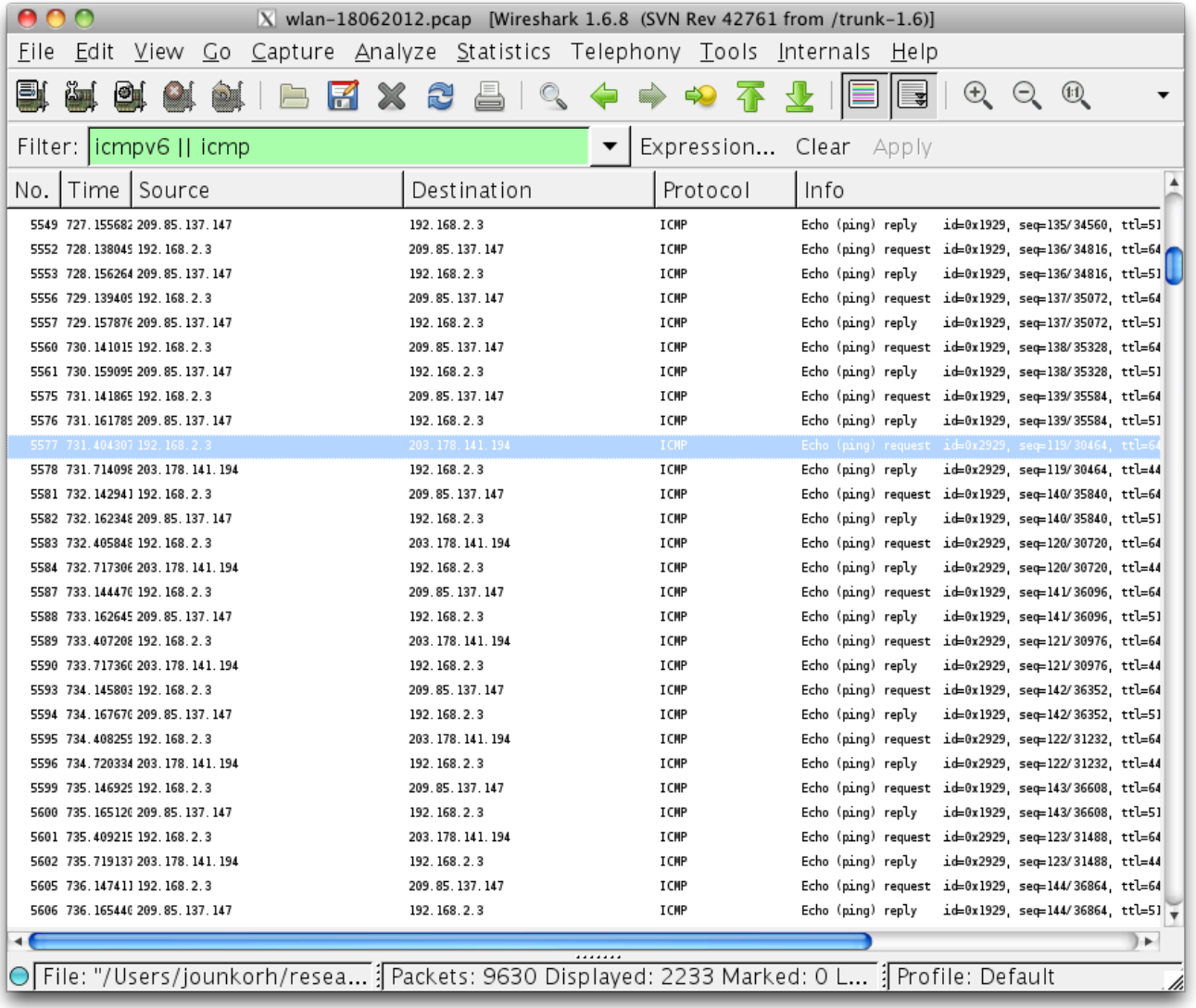
Only IPv6 left on 3G.



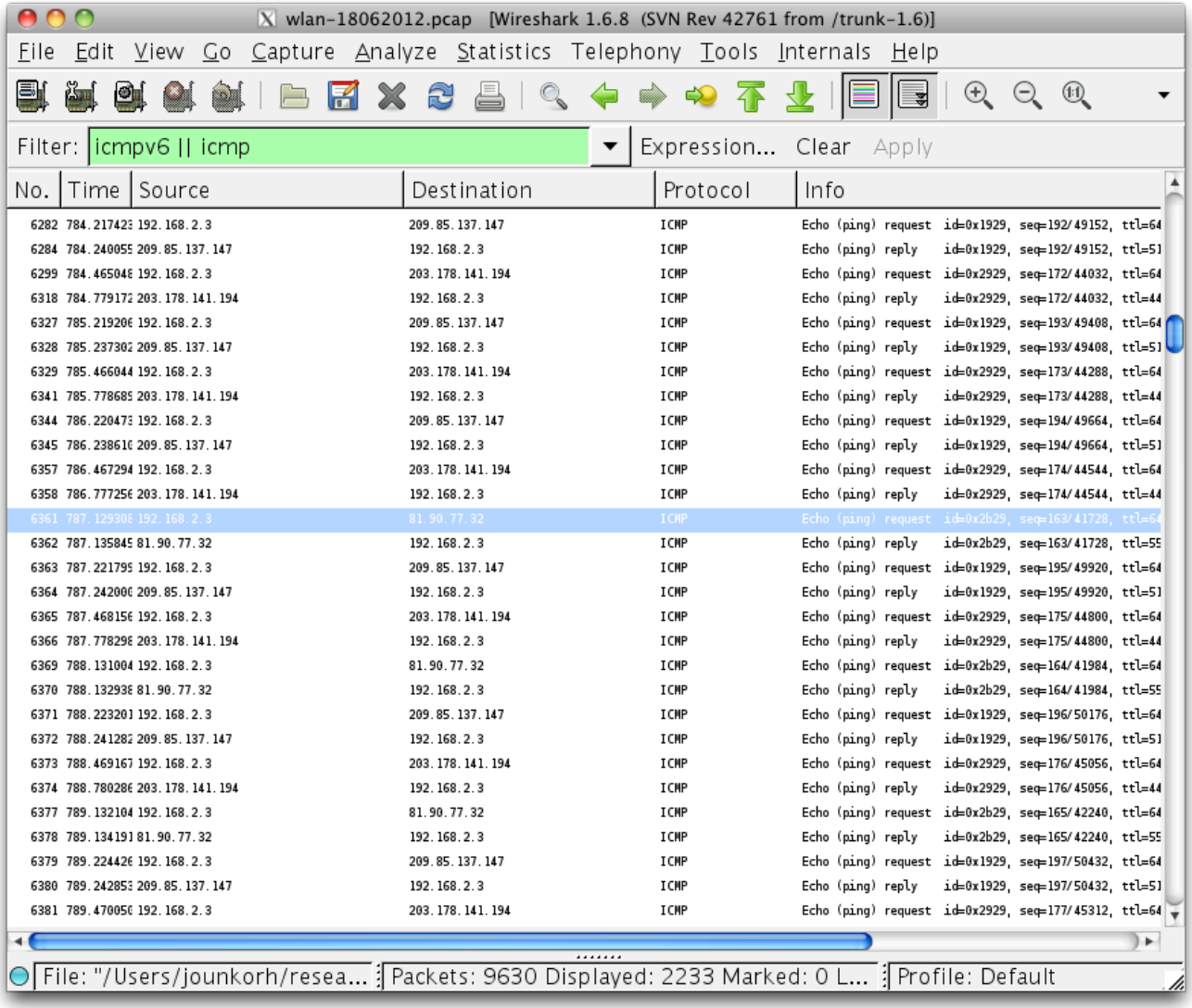
GGSN moved one IPv4 flow from 3G to WLAN



GGSN moved second IPv4 flow from 3G to WLAN

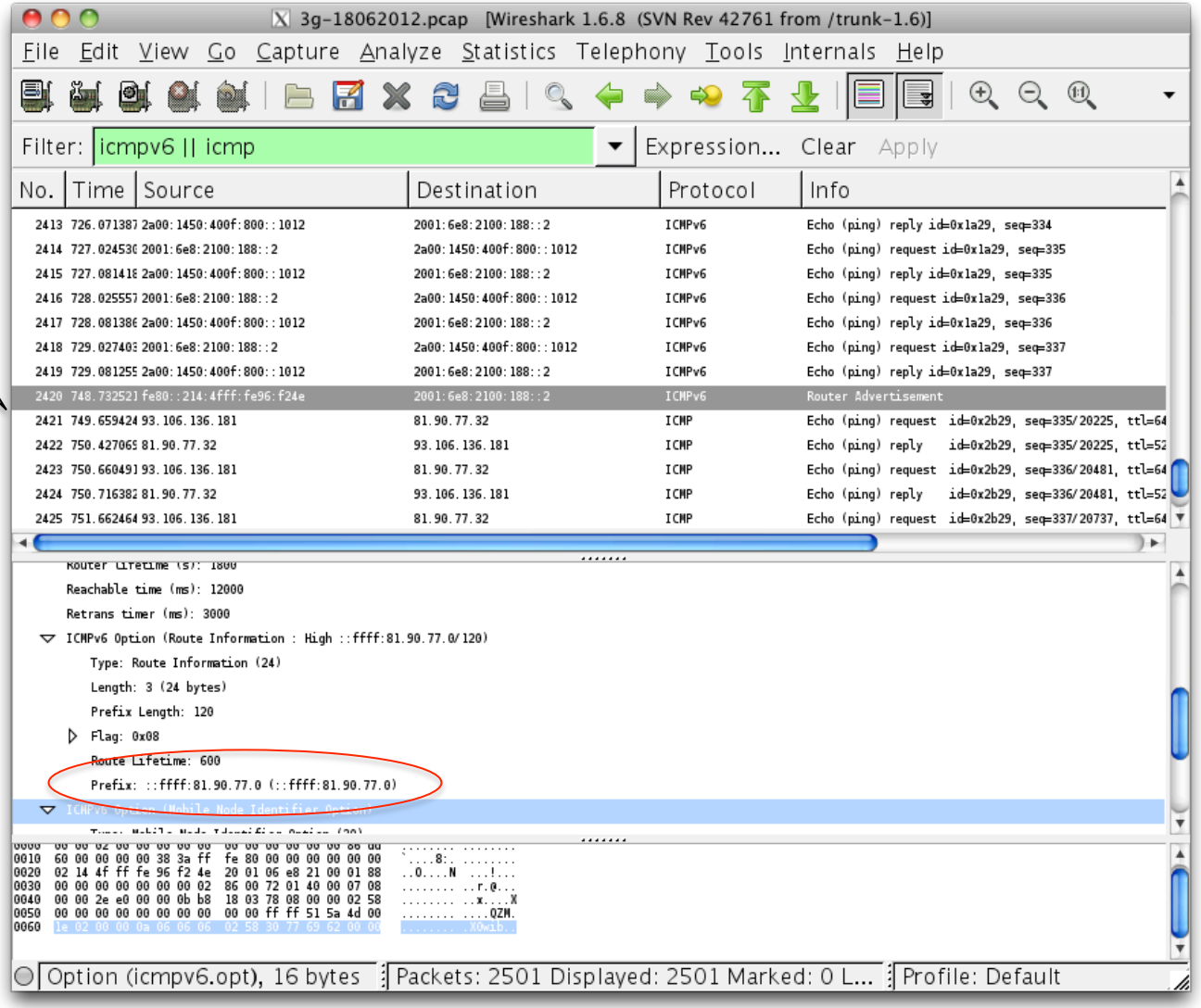


GGSN moved third IPv4 flow from 3G to WLAN



GGSN gets one IPv4 flow back to 3G

GGSN installs a route for funet..



Last comments..

- Both IPv4 and IPv6 offloading is doable.. but requires end host modification and standards push due the lack of IPv4 support. (it is there but not quite..)
- Note that the GGSN/PGW can only speak for itself. It cannot directly point traffic to other node/router. **The implicit "I don't want traffic" approach only available.**
- Opposition in IETF due mixing IPv4 into IPv6..
- DHCP not that much better since there are own version for both IPv4 and IPv6; and lacks easy push semantics.
- There is no ANDSF/DM/MOs/etc in these experiments/solutions..
- GGSN/PGW would require a policy/configuration interface for dynamic triggering of RAs with routing information.