

Efficiency of IPv4 - IPv6 Translators

Se-Young Yu
Prof Brian Carpenter
Assoc-Prof Nevil Brownlee
November 2010

What Do We Want to Find Out

Efficiency of translators for

Simultaneous connections

Different packet sizes

Compare 5 scenarios for HTTP access:

1. native IPv4

2. native IPv6

3. NAT64

4. NAT-PT

5. dual stack HTTP proxy

How NAT64 works

Clients send an IPv6 request packet to the address within NAT64 prefix.

NAT64 records session state and 5 tuple information for the session.

NAT64 translates source and destination address according to the translation mechanism

NAT64 sends translated IPv4 packet to the destination.

For rest of the session, translation is performed according to the recorded state.

Experiment Plan

A client sends 10000 packets for a connection

A client establishes 1-100 simultaneous connections.

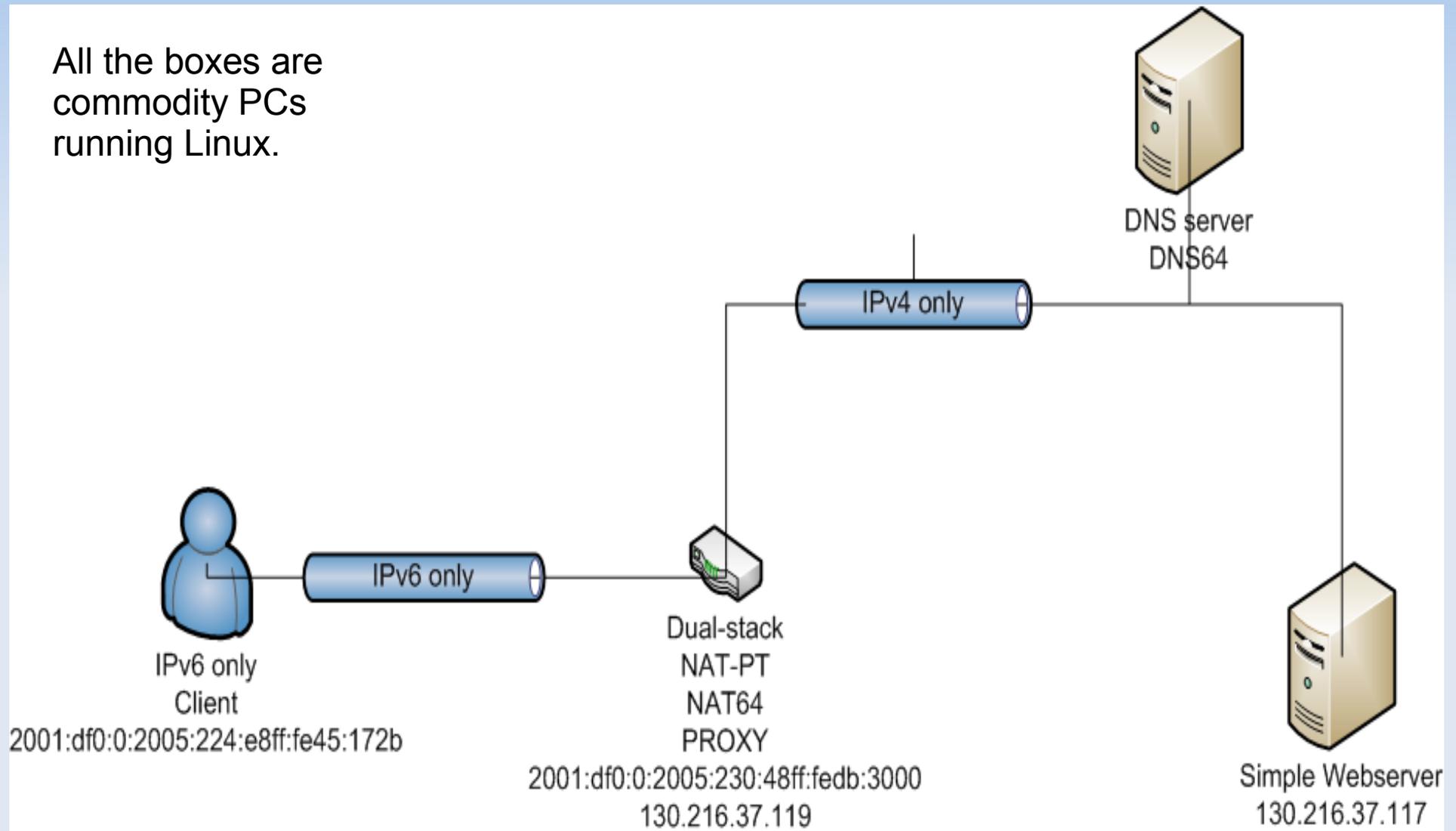
A client sends simple or large packet size HTTP requests.

A Linux router is implemented with NAT-PT, NAT64 (Viagenie) and HTTP Proxy (apache web server), as well as forwarding native IPv4 and IPv6.

Simple apache webserver is deployed to be tested as the target.

Experiment environment

All the boxes are commodity PCs running Linux.



Experiment Results: Simplest case

Median RTT

Native IPv4 :

631 μ sec

Native IPv6:

745 μ sec

NAT64:

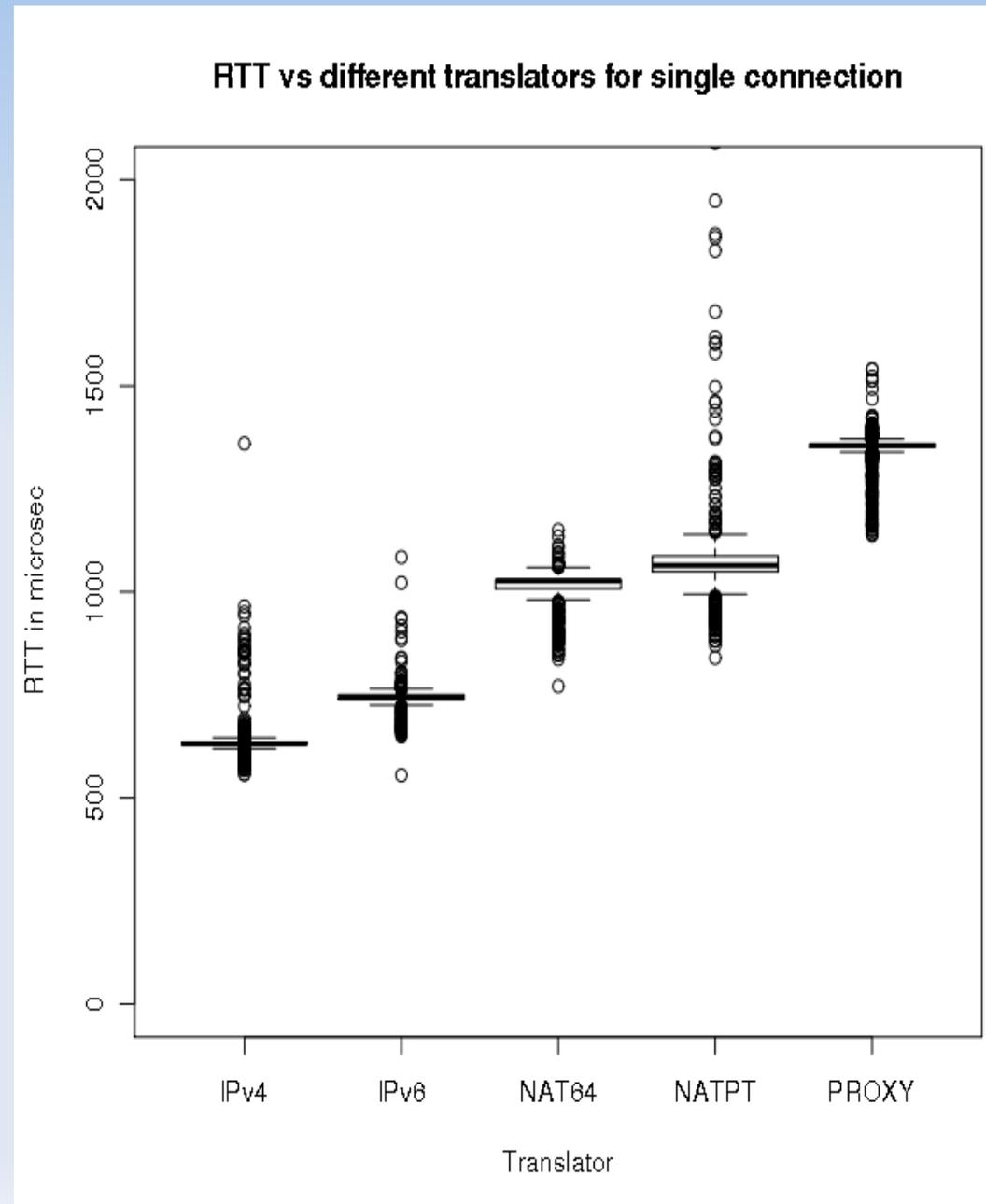
1027 μ sec

NAT-PT:

1064 μ sec

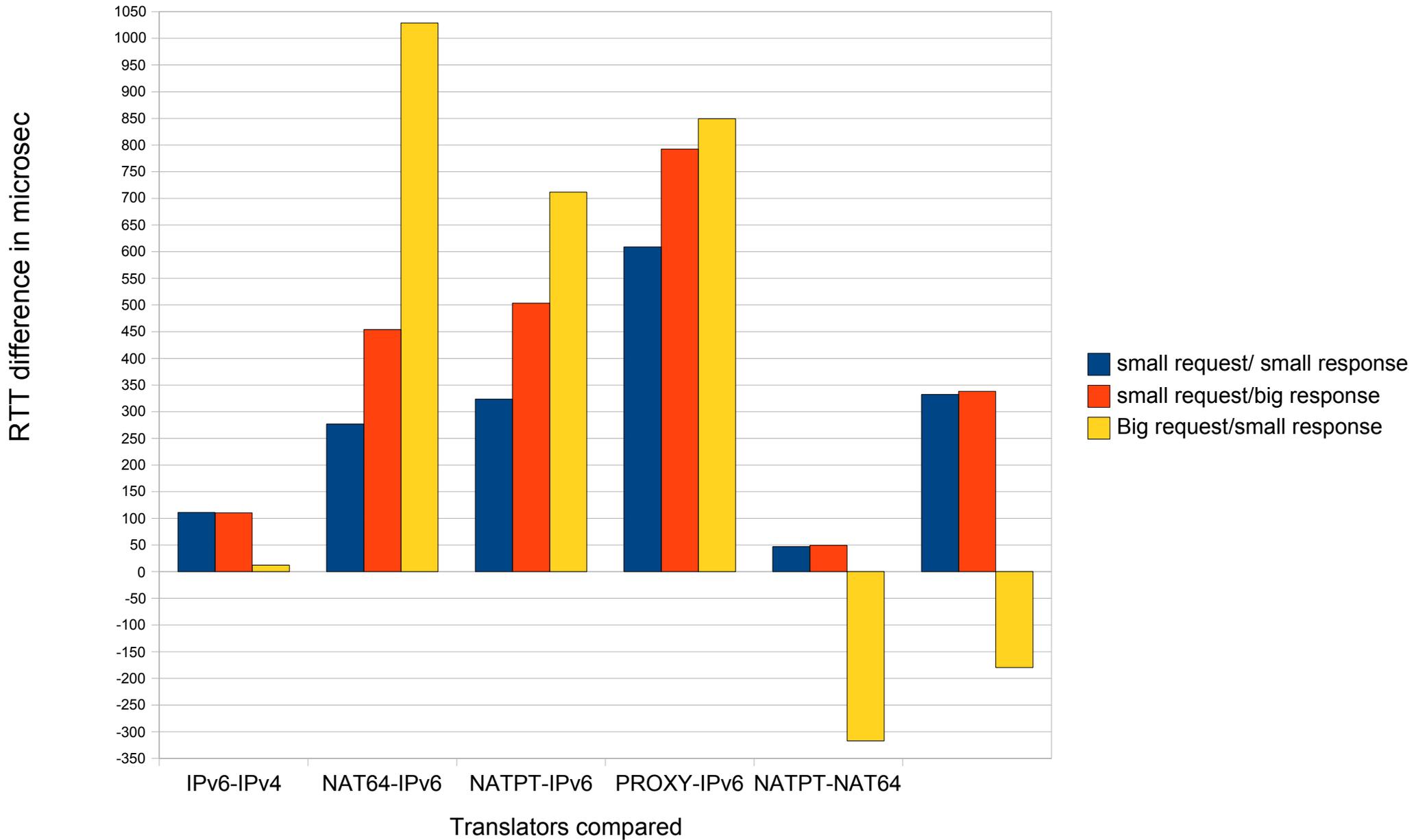
HTTP Proxy:

1355 μ sec



Experiment Results

RTT compared between different translators for a single connection



Conclusion

NAT64 is clearly a reasonable choice for a small size packet communication.

For large outbound packet NAT64 seems to perform badly (NAT64 code authors can't explain yet).

NAT-PT is also reasonable choice (from the performance view).

HTTP Proxy might not be too bad.

Please remember we are comparing implementations, not protocols!