Mapping IPv6 to/from IPv4

Technical challenges
with mapping IPv6 to/from IPv4

Adrian Kennard
FireBrick / Andrews & Arnold

Mapping IPv6 to IPv4

- What?
- Why?
- How?
- Other ways to do it
- Issues
- Security, reliability, legal

What? A mapping device

- One side has IPv4 packets
  - IPv4 source address
  - IPv4 destination address
- Other side has IPv6 packets
  - IPv6 source address
  - IPv6 destination address
- Sounds simple, doesn't it!

Why? General mapping

- The main reason we started this is as a general mapping facility
- FireBrick FB105 has IP and port mapping
- Next generation FB are IPv6 from the ground up
- They have to allow simple mapping between IPv4 and IPv6 worlds

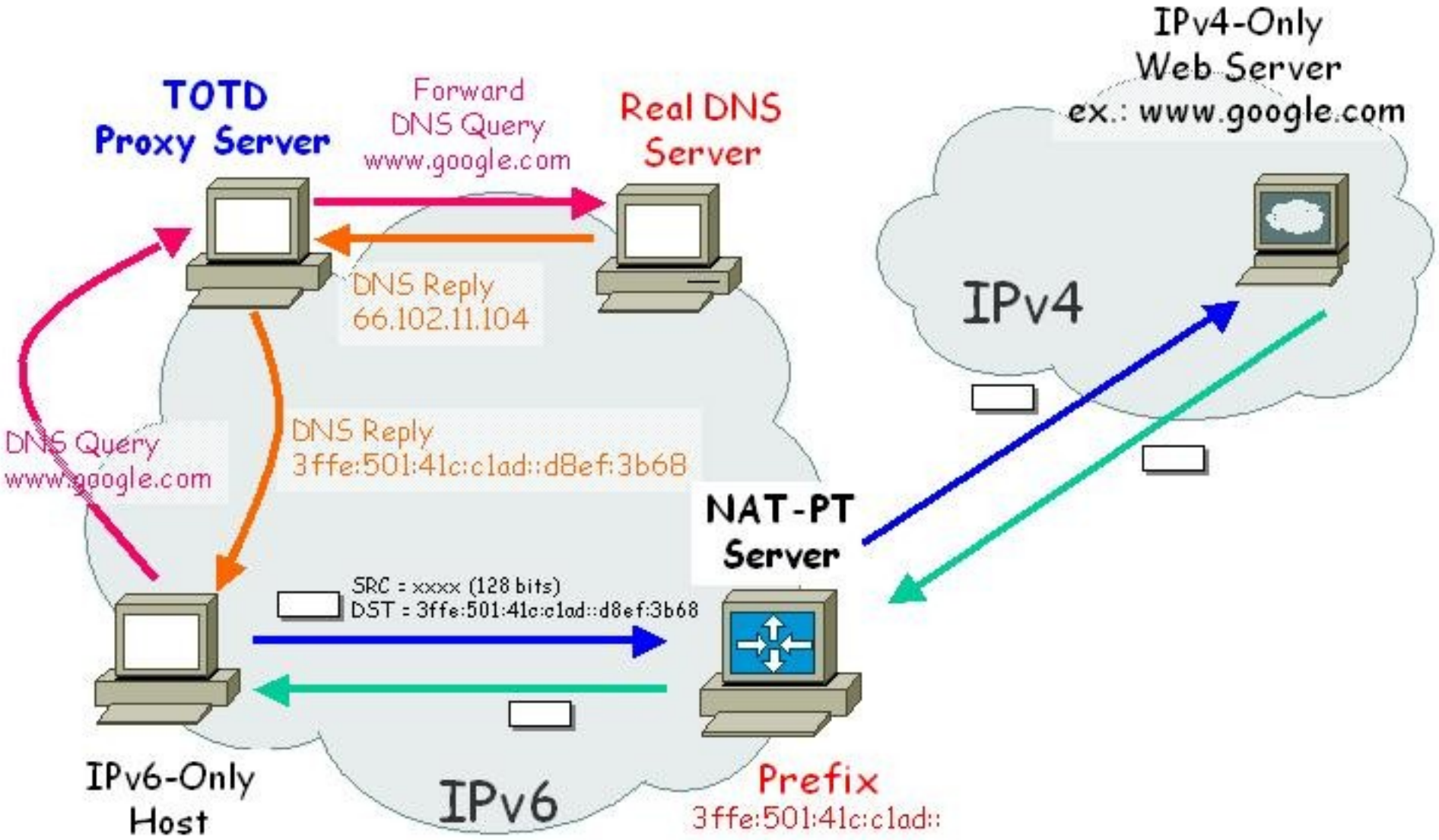
Why? IPv6 only hosts

- Side effect of doing this is supporting IPv6 only hosts talking to the IPv4 world.
- Several techniques for this need support that maps IPv4 to IPv6, e.g. NAT-PT, NAPT-PT.
- With DNS ALG support, allows IPv6 only hosts to seamlessly access IPv4 world for TCP and UDP traffic.

How? DNS ALG

- IPv6 only hosts have to be able to look up a target IP (IPv6) via DNS to connect
- DNS ALG (e.g. TOTD) provides fake replies which include IPv6 target address for IPv4 only target hosts
- Fake address sends traffic to the IPv6/IPv4 mapping gateway
- Fake address has embedded target IPv4 address so gateway can relay to IPv4
- What about DNSSEC?

TOTD for DNS-ALG



How? Change IP header

- You just need to change the IPv4 header to an IPv6 header – simples!
- How do you pick the source address?
- What about packet size? – it is bigger now
- What about protocol checksums?
- What about protocol ports?
- General NAT issues...

How? Mapping headers

- Change IP addresses, obviously
- ToS/DSCP ↔ Traffic class
- TTL ↔ Hop limit
- Frag offset / ID / more ↔ Frag header
- IPv4 options? Drop them or reject?
- Flow label? Ignore it?
- Packet length ↔ Packet length (–20...)
- Protocol ↔ Next header (ICMP/ICMPv6)

How? Pick a new source address

- One to one mapping (NAT) IPv4 ↔ IPv6?
  - Needs a lot of addresses
- NAPT where we change port as well
  - IPv4 from small pool of addresses
  - Single IPv4 address
    - Overload ports based on target address
- Mapping ports means limited protocols
  - UDP, TCP, ICMP and a few others

How? Protocol header

- Change ports for TCP/UDP (ID for ping)
- Adjust header checksum
  - Changed IPs in pseudo header
  - Changed ports
- ICMP (e.g. ping)
  - Different protocol
  - Different type/code
  - Different checksum algorithm

How? ICMP errors

- Change protocol, type, code as any ICMP
  - Type/code is not one to one mapping
- Quoted packet needs mapping
  - IP and port mappings and checksums
  - ICMP (e.g. ping) changes
  - Quoted packet is other way around
- Wrapper checksum adjust for changes including change of payload checksum
  - ICMP and ICMPv6 checksums
    - Pseudo header has length, add up frags!

General mapping issues

- Must session track anyway
  - Tracking TCP not that hard
  - Tracking UDP needs time-outs
    - Configurable time-outs?
  - Special ICMP error handling
- Hiding source identity
  - Is this a feature or a problem?
- ALG (assisting some protocols)
  - FTP, ident, thousands of others

General mapping issues

- Single point of failure unless multi device state tracking
- Resource limit (ports) creates point of DoS attack
- Loss of some semantics as IP headers not one to one mapping of all parameters
- Lack of support for protocols without any demultiplexing (ports), e.g. IPSec
- Issues with embedded IPs (FTP, etc)

How? Fragments and MTU

- Collate fragments (no actual reassembly)
  - Needed for fire-walling anyway
- IPv6 and IPv4 have different header sizes
- IPv4 to IPv6 treat as 1280 MTU at IPv4
  - ICMP error for DF on IPv4 side if too big
  - Else fragment and then send IPv6 frags
- IPv6 to IPv4 don't set DF
  - 20 extra bytes helps avoid fragments
- TCP MSS fix to reduce fragments

Ways to handle IPv6 only hosts

- SIIT: Stateless IP mapping, very limited
- NAT-PT: IP only mapping, limited
- NAPT-PT: IP and port mapping, as above
- TRT: Transport Relay Translator
  - Protocol stack each side
  - Avoids MTU issues
  - Can provide ALGs
  - Can mix NAT64 with TRT for difficult protocols that need ALGs

Implementations

- Open source solutions
  - natptd
    - Allows ALG plug-ins
  - pTRTd
  - Kame faith
  - totd for DNS ALG
- Commercial solutions
  - Major vendors like CISCO
  - FireBrick FB6000, FB2700, FB2500

Redundancy / scaling

- Needs session tracking, so hash based load sharing routing to multiple boxes can scale and replies use IP from whichever box go the traffic
- Failure of a box would drop sessions, unless some co-ordinated session tracking to backup and handover, like VRRP. Can be done

Security

- Sensible to lock down sources to specific customer IP blocks, obviously
- May want to lock down specific targets
  - We already block 0.0.0.0/8 127.0.0.0/8 224.0.0.0/3 as targets in IPv6/4 prefix
- Logging is a good idea, but logging every session is a lot
- Careful coding to avoid bogus packets breaking things – consider attacks!

Legal

- Do we have to be able to trace the real source from an IP?
  - Tricky to do, accurate clocks, IP addresses and even TCP/UDP ports needed
  - Lots of data to log, for a year? Is it actually legally required?
  - Depends on legislation and to some extent what is practical
  - Run the relay box in another country?
  - DEA notice needs *subscriber* IP address

Does it work?

- Yes – we turned it on and customers are using it (mostly experimentally)
  - Good feedback (just works)
- You can try it now – set DNS to 2001:8b0:6464::1 as we have not locked down sources yet
- TCP, UDP, ping, and even traceroute works
- Large (fragmented) TCP, UDP and even ICMP pings work

Traceroute works... (even from windoze!)

```
C:\>tracert bbc.co.uk
Tracing route to bbc.co.uk [2001:8b0:6464:0:666:616:d43a:e08a]
 over a maximum of 30 hops:
   1     *        *        *     Request timed out.
   2     3 ms     1 ms     1 ms  eclink.a.homeless.aaisp.net.uk
                                  [2001:8b0:0:31::51bb:1ffa]
   3     3 ms     2 ms     2 ms  a.armless.thn.aaisp.net.uk
                                  [2001:8b0:0:53::5a9b:3506]
   4     4 ms     3 ms     3 ms  nat64.discard.me.uk
                                  [::90.155.46.46]
   5     3 ms     3 ms     3 ms  a.armless.thn.aaisp.net.uk
                                  [::90.155.53.6]
   6     *      134 ms     8 ms  rt-lonap-a.thdo.bbc.co.uk
                                  [::193.203.5.90]
   7     4 ms     5 ms     3 ms  ::212.58.238.129
   8     4 ms     4 ms     4 ms  virtual-vip.thdo.bbc.co.uk
                                 [2001:8b0:6464:0:666:616:d43a:e08a]
```