SIIT-DC: IPv4 Service Continuity for IPv6 Data Centres

Tore Anderson
Redpill Linpro AS
RIPE69, London, November 2014
Stop Thinking IPv4; IPv6 is Here

• IPv4 is a dying and cramped protocol
• IPv6 is the exact opposite
• Which is the best choice to build a scalable and durable data centre infrastructure on?
• IPv6-only is the long-term goal, but how to get there in the easiest way possible?

A) IPv4-only -> IPv4+IPv6 compat -> Dual-Stack -> IPv6+IPv4 compat -> IPv6-only?
B) IPv4-only -------------------------------> IPv6+IPv4 compat -> IPv6-only?
IPv4? There's an “app” for that!

• Delivering IPv4 as a legacy backwards compatibility service on top of IPv6 is not only doable – it's fashionable
  – Data centres: Facebook, yours truly, ...
  – Mobile: Orange PL, Telenor NO, T-Mobile USA, ...
  – Wireline: Kabel Deutschland, ...

• For the data centre, SIIT-DC provides such an “app”, which runs on the outer network edge
An IPv6 data centre

The IPv6 Internet

IPv6 data centre infrastructure
An IPv6 data centre with SIIT-DC

The IPv6 Internet

The IPv4 Internet

SIIT-DC gateway(s)

IPv6 data centre infrastructure
So how does it work?

• IPv4 packets are *statelessly* translated to IPv6 and vice versa by the SIIT-DC GWs [RFC 6145]

• The end user's IPv4 source address is 1:1 mapped into a 96-bit IPv6 prefix [RFC 6052]

• The service's IPv4 destination address is rewritten according to a 1:1 IPv4:IPv6 mapping configured in the SIIT-DC gateways
• An IPv6 /96 prefix is assigned as the translation prefix representing the IPv4 internet and routed to the SIIT-DC gateways.
- An IPv6 /96 prefix is assigned as the translation prefix representing the IPv4 internet and routed to the SIIT-DC gateways
- A pool of IPv4 service addresses is assigned and routed to the SIIT-DC gateway
The SIIT-DC gateway is configured with static IPv4 mappings for each IPv6 service.

The IPv6 /96 prefix is configured as a default rule (used if no static map match).

IPv4 (IN A) records are added to DNS.
The client looks up the service's IPv4 address in DNS and connects to it like it would with any other IPv4 address.

The IPv4 packet is routed to the SIIT-DC gateway's IPv4 interface.
The SIIT-DC gateway translates the packet to IPv6
- DST address is rewritten according to static map
- SRC address gets the /96 prefix prepended (as it does not match any static maps)

Layer 4 payload is copied verbatim
- The server (or load balancer) responds to the packet just as it would with any other IPv6 packet
  - The server / LB requires no SIIT-DC support or awareness
- The original IPv4 source address is not lost
- Response packet is routed back to the SIIT-DC GW
The SIIT-DC gateway translates back to IPv4:
- SRC address according to static mapping rule
- DST address doesn't match any static map, so it only gets the /96 prefix stripped

Response packet is routed back to client
SIIT-DC highlights

- **Facilitates native end-to-end IPv6 everywhere!**
- Stateless per-packet operation
  - You can use anycast, ECMP load balancing, etc.
  - Flows are not required to bidirectionally traverse a single SIIT-DC gateway
  - Does not need to be the data centre's default IPv6 route, gateways may be located anywhere in the network
  - Concurrent flow count and fps are irrelevant for performance – scales similarly to an standard IP router
- The original IPv4 address remains known
  - Applications may geolocate IPv4 users, handle abuse, ...
Anycast, high availability, ECMP (load balancing)

No problem thanks to the stateless nature of SIIT-DC
SIIT-DC highlights, cont.

• Maximum conservation of IPv4 addresses
  – 1 address used per public service, none lost to infrastructure or subnet $^2$ overhead, etc.
  – A typical DC might have thousands of servers for each service that's publicly available from the internet

• Single-stack applications and server LANs
  – Avoid the dual-stack complexity of running two protocols

• Application stack is independent of IPv4
  – Forget about further IPv6 migration projects
  – When IPv4 has become irrelevant, remove IN A DNS records and shut down your SIIT-DC gateways - done
Application requirements

• If the application does work through NAT44, it will likely work with SIIT-DC as well
  – e.g., HTTP and HTTPS

• If the application does **not** work through NAT44, it will likely not work with plain SIIT-DC
  – e.g., FTP (uses IP literals in Layer 7 payload)

• The servers' OS and application stacks must fully support IPv6
Supporting IPv4-only applications

• A Host Agent reverses the SIIT-DC Gateway's translations before passing data to the application
• Application handles IPv4 traffic on an IPv4 socket
• Very similar to the CLAT component in 464XLAT
  – End-to-end IPv4 address transparency; referrals work
There's even running code

- TAYGA for Linux (open source)
  - http://www.litech.org/tayga/
- Cisco ASR1K
  - Requires IOS XE v3.10
- Brocade ServerIron ADX (not tested by me)
- F5 BIG-IP LTM (not tested by me)
- https://github.com/toreanderson/clatd
  - SIIT-DC Host Agent for Linux (uses TAYGA)
Questions?
Thank you for listening!

Further reading:

RFC 6052 - IPv6 Addressing of IPv4/IPv6 Translators
RFC 6145 - IP/ICMP Translation Algorithm
draft-anderson-v6ops-siit-dc - Stateless IP/ICMP Translation in IPv6 Data Centre Environments
draft-anderson-v6ops-siit-dc-2xlat – SIIT-DC: Dual Translation Mode
http://toreanderson.no - My personal home page (contact info, social media links, slides from this and earlier talks)
http://redpill-linpro.com - My employer and sponsor of this project

Note: IPv4 traffic to both of the above URLs is routed through a SIIT-DC gateway (eating my own dog food)