IPv6 Labor

IPv6 Gateways
(TRT)

Stephan Uhlmann <su@su2.info>
02.10.2002

Copyright (c) 2002 Stephan Uhlmann

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts and no Back-Cover Texts. A copy of the license can be obtained from http://www.gnu.org/licenses/fdl.html.

Motivation

- verbinden reine IPv6 Netze mit IPv4 Netzen

- wenn nicht alle Hosts (Server) Dual-Stack fähig sind oder sein sollen (da IPv4 + IPv6 Adressen benötigt werden)

- bester Ansatzpunkt: Router

Modelle

- SOCKS64
  - erweitertes SOCKS
- ALGs (Application Layer Gateways)
  - Proxies auf Dual-Stacks
- SIIT (Stateless IP/ICMP Translation Algorithm, RFC 2765)
  - allg. Methode zum Übersetzen von IPv4 und IPv6 Paketen

Modelle

- NAT-PT (Network Address Translation / Protocol Translation, RFC 2766)
  - NAT: temp. IPv4 Adresse für IPv4 Traffic, PT == SIIT
  - Linux-Implementation: nat-pt (Electronics and Telecommunications Research Institute, Korea)

- TRT (Transport Relay Translator, RFC 3142)
  - DNS: IPv4 -> spezielle IPv6 Adresse (dummy Prefix), TRT fängt diese ab und setzt sie in IPv4 um
  - Implementation: faith/faithd (BSD), pTRTd (portabel)

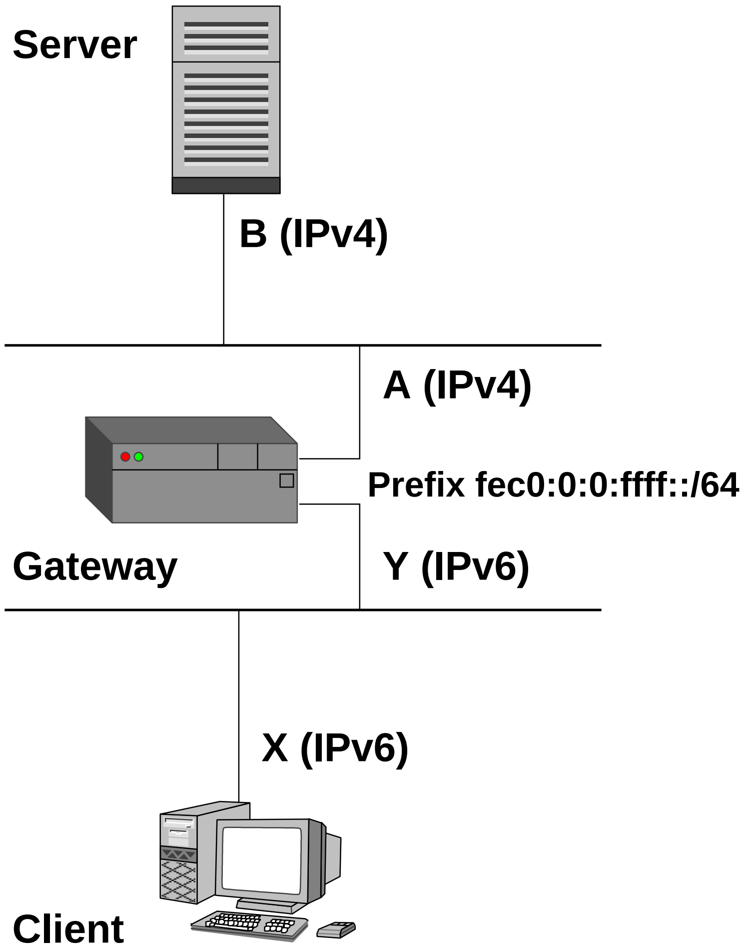
TRT

- RFC 3142

- ähnlich NAT-PT, aber: Transport Layer

- kein Header rewriting sondern “hartes” Relaying,
TRT kappt Verbindung und baut eine zweite zum Zielsystem auf

- benötigt DNS-Proxy


Server
B (IPv4)
A (IPv4)
Prefix fec0:0:0:ffff::/64
Gateway
Y (IPv6)
X (IPv6)
Client

TRT

- Vorteile
  - einfach zu implementieren und zu verstehen
  - keine Modifikationen an den Hosts notwendig
  - Path MTU discovery / Fragmentation unkomplizierter (IPv4 und IPv6 Teile können getrennt betrachtet werden)
  - ALGs können im TRT implementiert werden

TRT

- Nachteile:
  - Informationsverlust bei IPv6<->IPv4 (wie bei SIIT)
  - wie NAT-PT: benötigt auch ALGs für NAT-unfreundliche Protokolle (FTP, DNS), IPSec nicht möglich
  - nur IPv6 -> IPv4
  - nur bidirektionale Verbindungen, kein unidirektionaler (Multicast)
  - TRT ist “stateful” -> alle Pakete einer Verbinding müssen also über den gleichen TRT gehen (Single Point of Failure)

pTRTd

- relayt IPv4 -> fec0:0:0:ffff::/64 (default)

- portable Implementation (eigener IPv6 Stack)

- benötigt TUN/TAP device
  `mknod /dev/net/tun c 10 200`
  `modprobe tun`

totd

- DNS-Proxy, leitet Anfragen an Forwarder weiter

- Aufgabe: Umwandlung IPv4 Adressen in “Fake” IPv6 Adressen

- Bsp.: `dig @::1 -p 5005 www.google.com aaaa`
```
;; ANSWER SECTION:
www.google.com. 122 IN AAAA fec0::ffff:0:0:d8ef:2365
```

Praxis

- totd funktioniert gut

- pTRTd ist Alpha, scheint aber in der Grundfunktionalität zu gehen

- Problem:
  ```
  [root@bender stephan]# ping6 fec0::ffff:0:0:d8ef:2365
  connect: Cannot assign requested address
  ```

- connect() Systemaufruf liefert `EADDRNOTAVAIL`

- Gelöst durch Verwendung einer "global scope" Adresse

```
# ./totd -c totd.conf
# ./ptrtd -p 2001:7a0:101:888::
# ping6 2001:7a0:101:888::1 (zurückgepingt von ptrtd)
# dig @::1 -p 5005 lisa.cs.uni-potsdam.de aaaa

; ANSWER SECTION:
lisa.cs.uni-potsdam.de. 172800 IN AAAA 2001:7a0:101:888::8d59:30de

# ssh -6 -l uhlmann 2001:7a0:101:888::8d59:30de
uhlmann@lisa:~ >
```

Links

- RFC 3142 (TRT): `http://www.ietf.org/rfc/rfc3142.txt`
- pTRTd: `http://www.litech.org/ptrtd/`
- totd: `http://www.vermicelli.pasta.cs.uit.no/ipv6/software.html`
- Modellvergleich IPv6->IPv4 Gateways: `http://web.mit.edu/peilei/www/nats2.htm`