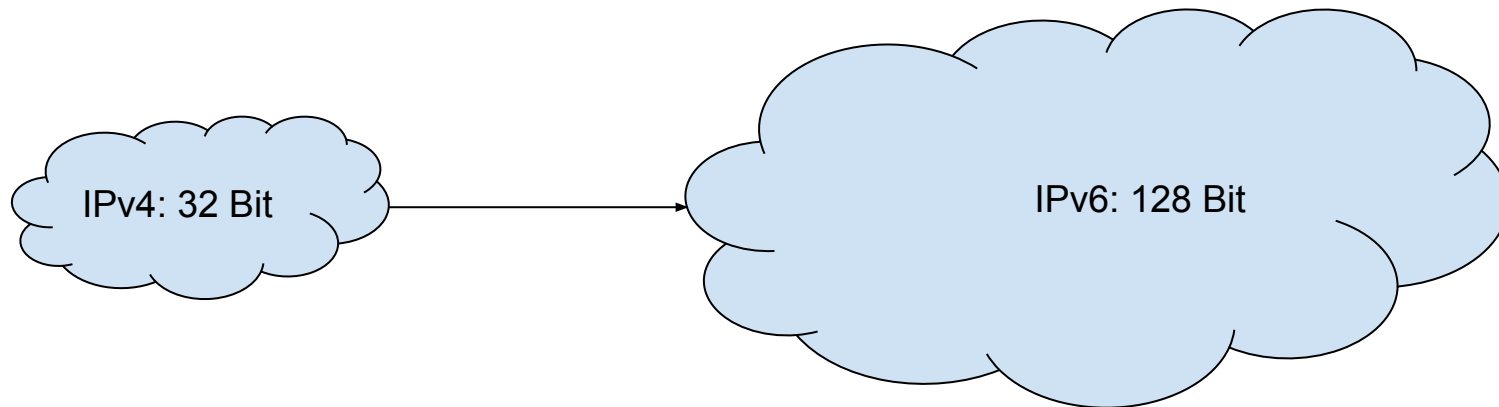
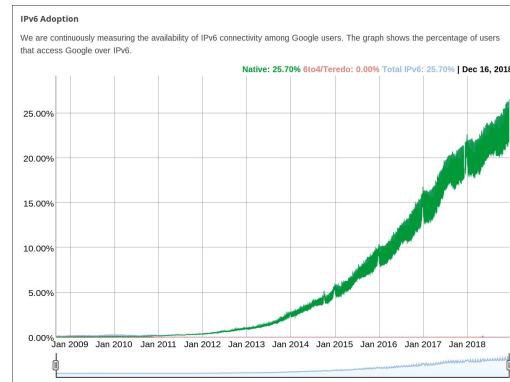


Implementation of a Layer 7 IPv4 to IPv6 Reverse Proxy

Nico Schottelius, Sarah Plocher

Motivation

- IPv4 addresses have run out
- IPv6 is here, today
- Smart mechanism for bridging during transition time
- Need 1:N mappings

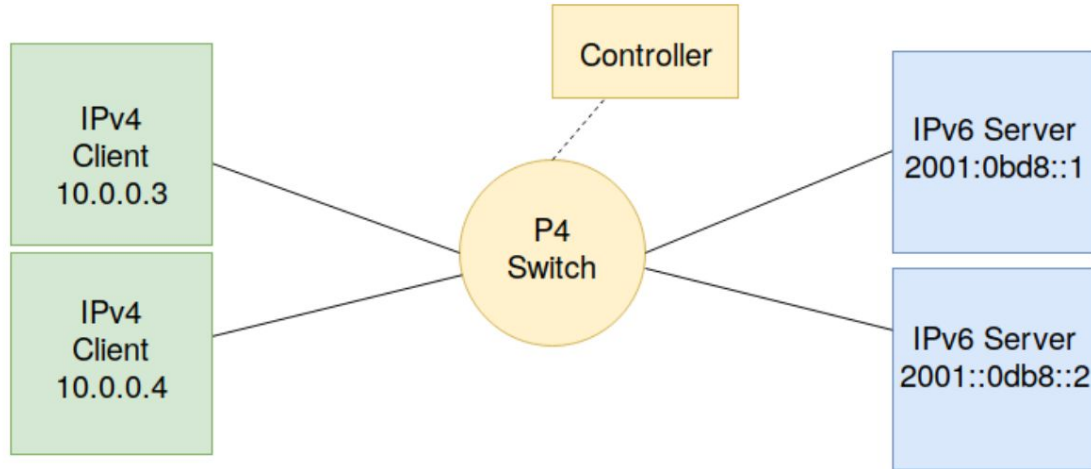


Real world example

- ipv6onlyhosting.com
- Product of ungleich glarus ag
- Using **ssh** jump host, **nginx** and **haproxy** in various modes



Idea



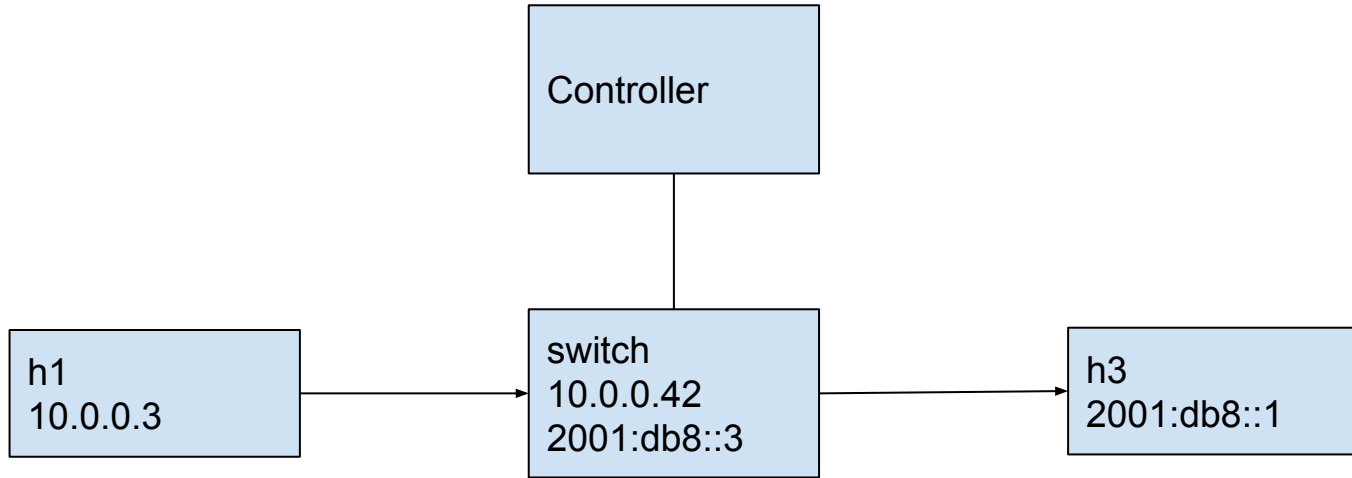
Real world problems

- No Neighbor discovery protocol
 - Not even plain IPv6 <-> IPv6 communication possible
 - Solution: static neighbor entries
- Sending incorrect packets with scapy
 - Strange mac addresses (“missing layer”)
- (Not) Handling of FIN
 - Socket in use errors

Real world problems (2)

- TCP Checksums
 - Changing seq/ack requires new checksum
 - Scapy does not recalculate by default
 - IPv6 is different than IPv4 (for TCP!!!)
- Buffering / Ordering
 - IPv4 client already sends FIN prior to IPv6 backend established!
 - Need to buffer packets!

The solution



IPv4 connection

```
INFO:main:INCOMING: 10.0.0.3:33526 => 10.0.0.42:80: flags=S seq=2605695646 ack=0 raw=None  
INFO:main:OUTGOING: 10.0.0.42:80 => 10.0.0.3:33526: flags=SA seq=42 ack=2605695647 raw=None  
INFO:main:INCOMING: 10.0.0.3:33526 => 10.0.0.42:80: flags=A seq=2605695647 ack=43 raw=None  
INFO:main: --- CLIENT CONNECTION ESTABLISHED
```


IPv6 connection setup

```
INFO:main:INCOMING: 10.0.0.3:33526 => 10.0.0.42:80: flags=PA seq=2605695647 ack=43 raw=A
INFO:main:OUTGOING: 2001:db8::3:33526 => 2001:db8::1:80: flags=S seq=2605695646 ack=0 raw=None
INFO:main:BUFFER: <Ether  dst=00:00:0a:00:00:42 src=00:00:0a:00:00:03 type=0x800 |<IP  version=4 ihl=5 tos=0x0
INFO:main:INCOMING: 10.0.0.3:33526 => 10.0.0.42:80: flags=FA seq=2605695649 ack=43 raw=None
INFO:main:BUFFER: <Ether  dst=00:00:0a:00:00:42 src=00:00:0a:00:00:03 type=0x800 |<IP  version=4 ihl=5 tos=0x0
INFO:main:INCOMING: 2001:db8::1:80 => 2001:db8::3:33526: flags=SA seq=3769728775 ack=2605695647 raw=None
INFO:main:OUTGOING: 2001:db8::3:33526 => 2001:db8::1:80: flags=A seq=2605695647 ack=3769728776 raw=None
INFO:main: --- BACKEND CONN ESTABLISHED
```

NAT64 in action

```
INFO:main:Emptying buffer
INFO:main:NAT46 IN : 10.0.0.3:33526 => 10.0.0.42:80: flags=PA seq=2605695647 ack=43 raw=A
ck=2647
INFO:main:NAT46 OUT: 2001:db8::3:33526 => 2001:db8::1:80: flags=PA seq=2605695647 ack=3769728776 raw=A
ck=None
INFO:main:NAT46 answer: <Ether dst=00:00:0a:00:00:01 src=00:00:0a:00:00:42 type=0x86dd |<IPv6 nh=TCP src=2001:db8::3 dst=2001:db8::1
INFO:main:NAT46 IN : 10.0.0.3:33526 => 10.0.0.42:80: flags=FA seq=2605695649 ack=43 raw=None ck=19304
INFO:main:NAT46 OUT: 2001:db8::3:33526 => 2001:db8::1:80: flags=FA seq=3769728776 raw=None ck=None
INFO:main:NAT46 answer: <Ether dst=00:00:0a:00:00:01 src=00:00:0a:00:00:42 type=0x86dd |<IPv6 nh=TCP src=2001:db8::3 dst=2001:db8::1
INFO:main:INCOMING: 2001:db8::1:80 => 2001:db8::3:33526: flags=PA seq=3769728776 ack=2605695647 raw=ok
INFO:main:NAT64 IN : 2001:db8::1:80 => 2001:db8::3:33526: flags=PA seq=3769728776 ack=2605695647 raw=ok
INFO:main:NAT64 OUT: 10.0.0.42:80 => 10.0.0.3:33526: flags=PA seq=43 ack=2605695647 raw=ok
INFO:main:INCOMING: 2001:db8::1:80 => 2001:db8::3:33526: flags=FA seq=3769728779 ack=2605695647 raw=None
INFO:main:NAT64 IN : 2001:db8::1:80 => 2001:db8::3:33526: flags=FA seq=3769728779 ack=2605695647 raw=None
INFO:main:NAT64 OUT: 10.0.0.42:80 => 10.0.0.3:33526: flags=FA seq=46 ack=2605695647 raw=None
INFO:main:INCOMING: 2001:db8::1:80 => 2001:db8::3:33526: flags=A seq=3769728780 ack=2605695649 raw=None
INFO:main:NAT64 IN : 2001:db8::1:80 => 2001:db8::3:33526: flags=A seq=3769728780 ack=2605695649 raw=None
INFO:main:NAT64 OUT: 10.0.0.42:80 => 10.0.0.3:33526: flags=A seq=47 ack=2605695649 raw=None
INFO:main:INCOMING: 2001:db8::1:80 => 2001:db8::3:33526: flags=A seq=3769728780 ack=2605695650 raw=None
INFO:main:NAT64 IN : 2001:db8::1:80 => 2001:db8::3:33526: flags=A seq=3769728780 ack=2605695650 raw=None
INFO:main:NAT64 OUT: 10.0.0.42:80 => 10.0.0.3:33526: flags=A seq=47 ack=2605695650 raw=None
INFO:main:INCOMING: 10.0.0.3:33526 => 10.0.0.42:80: flags=A seq=2605695650 ack=46 raw=None
INFO:main:NAT46 IN : 10.0.0.3:33526 => 10.0.0.42:80: flags=A seq=2605695650 ack=46 raw=None ck=19301
INFO:main:NAT46 OUT: 2001:db8::3:33526 => 2001:db8::1:80: flags=A seq=2605695650 ack=3769728779 raw=None ck=None
INFO:main:NAT46 answer: <Ether dst=00:00:0a:00:00:01 src=00:00:0a:00:00:42 type=0x86dd |<IPv6 nh=TCP src=2001:db8::3 dst=2001:db8::1
INFO:main:INCOMING: 10.0.0.3:33526 => 10.0.0.42:80: flags=A seq=2605695650 ack=47 raw=None
INFO:main:NAT46 IN : 10.0.0.3:33526 => 10.0.0.42:80: flags=A seq=2605695650 ack=47 raw=None ck=19300
INFO:main:NAT46 OUT: 2001:db8::3:33526 => 2001:db8::1:80: flags=A seq=2605695650 ack=3769728780 raw=None ck=None
INFO:main:NAT46 answer: <Ether dst=00:00:0a:00:00:01 src=00:00:0a:00:00:42 type=0x86dd |<IPv6 nh=TCP src=2001:db8::3 dst=2001:db8::1
```

Demo notes

- Open up terminals and show
 - h1
 - ip addr
 - ip -6 neigh
 - h3
 - ip addr
 - ip neigh
- Try to ping
 - h3 ping 2001:db8::1 # fails, mention other way also does not work

Demo notes 2: Show NAT64

- Open additional terminals
 - mininet / switch
 - controller
 - mx h1 tcpdump -lni any
 - mx h3 tcpdump -lni any
- Use existing terminal h1
 - echo "OK" | mx h1 socat TCP6-LISTEN:80 -
- Drum rolls
- Use existing terminal h3
 - socat A | socat - TCP:10.0.0.42:80
- Explain that A/B are used to decide where to proxy to

Questions?