

NTP over IPv6

tf-ngn

Pozman, 8 May 2003

Laura Serrano – RedIRIS

Is NTP over IPv6 ready for use?

⌘ Current status

- ☑ NTP version 3
- ☑ Synchronization over native IPv4

⌘ Objectives

- ☑ NTP version 4
- ☑ How IPv6 stack affects to NTP synchronization and NTP convergence?

Network and software requirements

- ⌘ NTP client
- ⌘ NTP server
- ⌘ IPv6 network infrastructure
- ⌘ NTP version with IPv6 support
 - ☑ NTP version 4

Network Scenario (I)

⌘ 2 NTP servers -> stratum 1

☑ hora.rediris.es

☑ IPv4 address -> 130.206.1.163

☑ IPv6 address -> 3ffe:3328:5:1:2a0:24ff:fe4b:74a8

☑ pulsar.rediris.es

☑ IPv4 address -> 130.206.0.153

☑ IPv6 address -> 2001:720:419::2

⌘ 1 NTP client -> stratum 2

☑ picachu.rediris.es

☑ IPv4 address -> 130.206.204.14

☑ IPv6 address -> 2001:720:1214:1

Network Scenario (II)

⌘ 3 different network environments

- ☑ Native IPv4 -> full connectivity to both servers

Similar to any normal production environment

- ☑ Mixed native IPv6/tunnelled IPv6 -> hora.rediris.es

Most of the experimental setups follow this schema

- ☑ Native IPv6 -> pulsar.rediris.es

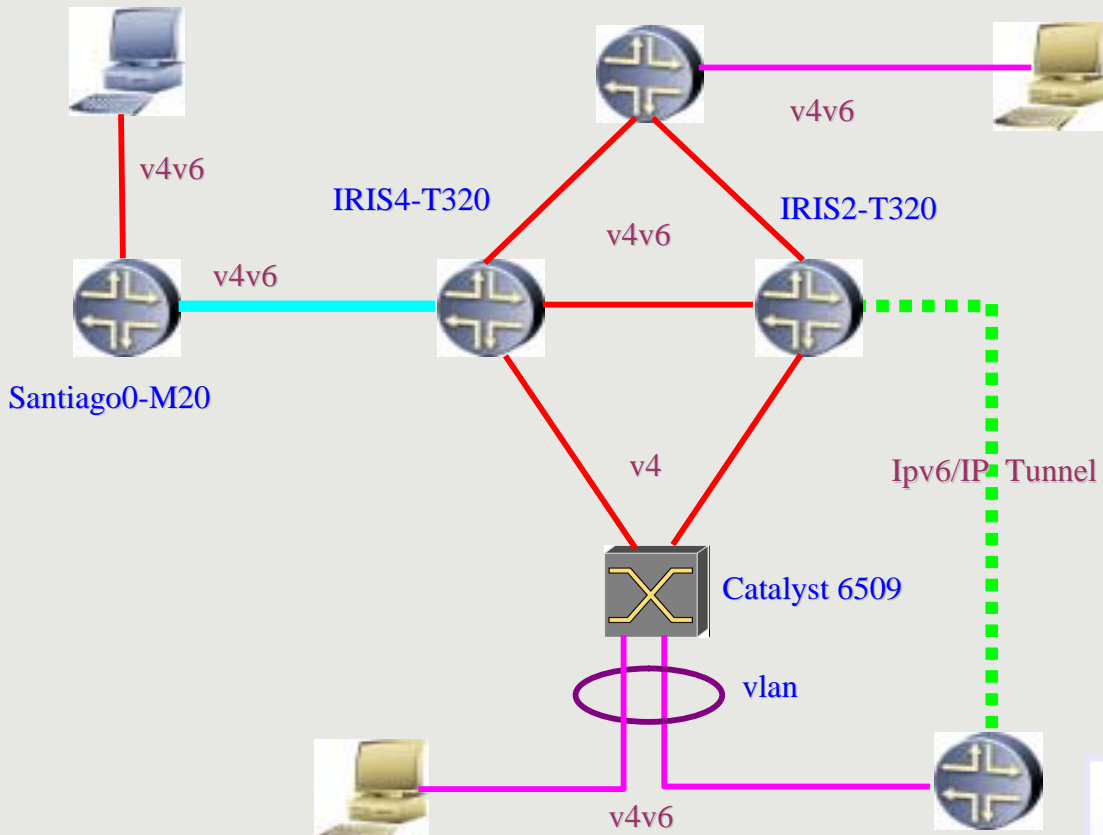
The whole path is IPv6-native between client and server.

Network Scenario (III)

picachu-NTP Client





Madrid0-M40

pulsar-NTP Server



hora-NTP Server

Cisco 2505

	STM-16 SDH
	G/E , 1000 BASE-SX
	Ipv6/IP Tunnel
	F/E 100 BASE-TX

Server settings

⌘ hora.rediris.es

- ☑ Linux RedHat 7.3 i386/2.4.19-NANO
- ☑ NTP daemon version 4.1.73

⌘ pulsar.rediris.es

- ☑ Linux RedHat 7.3 i386/2.4.19-NANO
- ☑ NTP daemon version 4.1.74

-NANO stands for nanoseconds support and PPS API for the Linux kernel:

`ftp://ftp.kernel.org/pub/linux/daemons/ntp/PPS/PPSkit-2.1.1.tar.bz2`

Client settings

⌘ picachu.rediris.es

☑ Sun Solaris 8

☑ NTP version 4.1.74

☑ /etc/ntp.conf file

```
server 3ffe:3328:5:1:2a0:24ff:fe4b:74a8 minpoll 4 maxpoll 6
```

```
server 130.206.1.163 minpoll 4 maxpoll 6
```

```
server 2001:720:419::2 minpoll 4 maxpoll 6
```

```
server 130.206.0.153 minpoll 4 maxpoll 6
```


Tunnelled vs. native IPv6

```
ntpdc> peer
      remote                local      st poll reach  delay  offset  disp
=====
*130.206.1.163    130.206.204.14    1   16   377 0.00923  0.000092  0.00044
=3ffe:3328:5:1:2  2001:720:1214:1   1   16   377 0.01424 -0.000213 0.00113
=130.206.0.153    130.206.204.14    1   16   377 0.00917 -0.000047  0.00101
=2001:720:419::2  2001:720:1214:1   1   16   377 0.00917 -0.000033 0.00082
```

- ⌘ Tunnel delay ~ 53% higher than native IPv6
- ⌘ This delay affects to the dispersion value
- ⌘ An IPv6 server in this enviroment is not a goot election as server reference

Native IPv4 vs. native IPv6

```
ntpdc> peer
      remote          local      st poll reach  delay  offset  disp
-----
=130.206.1.163      130.206.204.14  1   64   377 0.00941 -0.000005 0.00487
=3ffe:3328:5:1:2    2001:720:1214:1  1   64   377 0.01433 -0.000174 0.00507
=130.206.0.153      130.206.204.14  1   64   377 0.00919 -0.000002 0.00391
*2001:720:419::2    2001:720:1214:1  1   64   377 0.00920 0.000000 0.00378
```

- ⌘ IPv6 delay is the same that IPv4 delay
- ⌘ The synchronization values seem to be better to IPv6 than IPv4 (offset and dispersion)
- ⌘ An IPv6 server in a native environment is a good election as a server reference

Conclusions

- ⌘ NTP peerings over IPv6 should be more reliable because of the superior capabilities of the IPv6 stack.
- ⌘ Data obtained in this experiment maybe is not enough for extrapolating the in-theory better behaviour of NTP over IPv6.
- ⌘ We want interested parties for testing our setting and are seeking contributions from

YOU

**That's all,
Folks!!**

