

IPV6

Lorenzo Bracciale

Marco Bonola

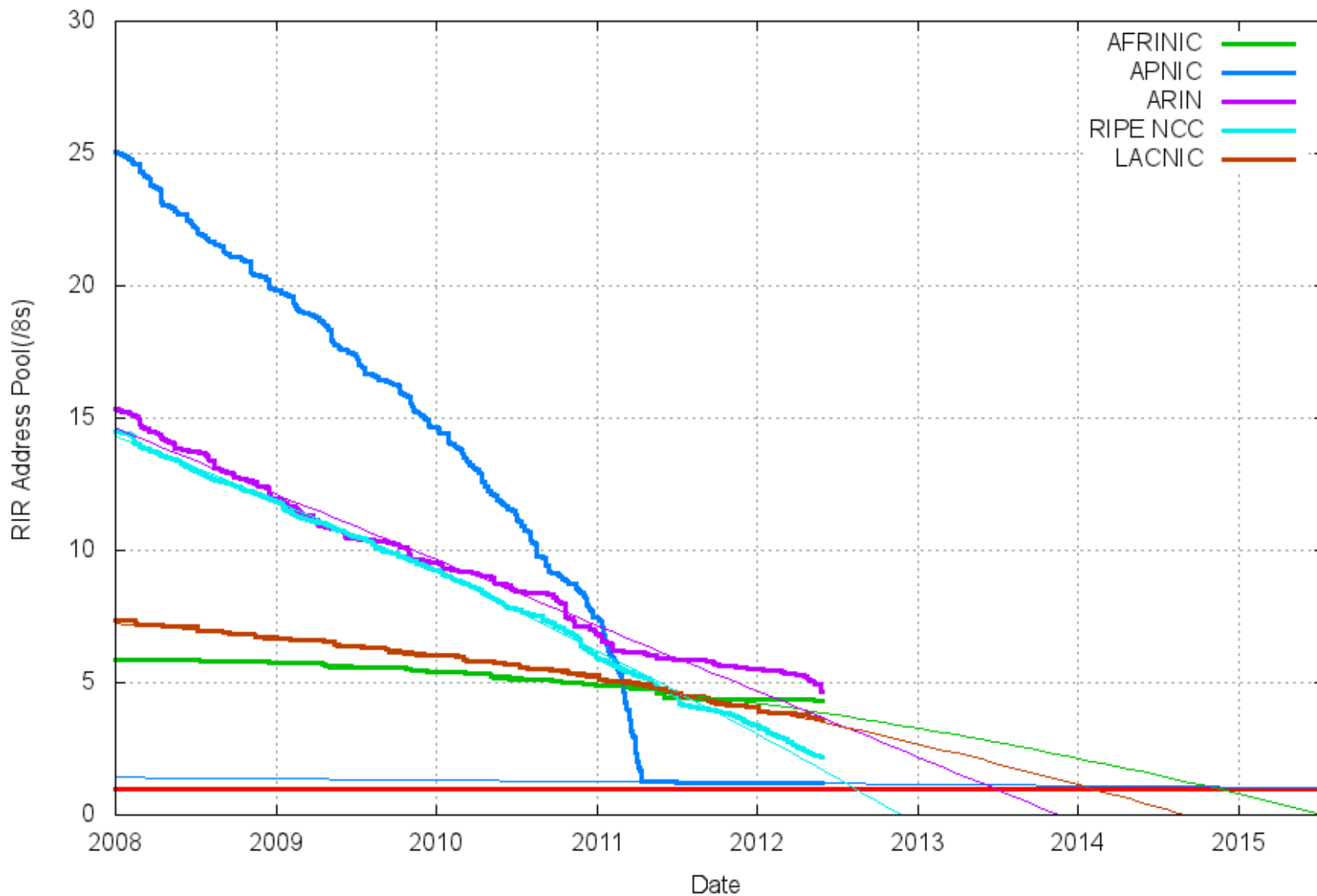
Study here!

<http://www.ip6.com/us/book/>

Oops...

- IPv4 is described in RFC 791 (September 1981)
 - address space: 2^{32}
- IPv4 allocation pool has been a concern since the late 1980s:
 - Mobile devices
 - Internet of the things
 - Always-on connections
 - Internet demographics
 - Inefficient address use. With classfull, three sizes:
 - Class A: 16 millions of addresses
 - Class B: 65536 addresses
 - Class C: 256 addresses

RIR IPv4 Address Run-Down Model



Shortcoming solutions/ mitigations

- Classless Inter-Domain Routing (**CIDR**) in 1993
- Virtual Hosting
- Tighter control by regional Internet registries
- **Network address translation (NAT)**
 - popular in mid-1990s, RFC standard (!) 1999
 - statefull (limited scalability)
 - Broke L4 end-to-end
 - Communication (e.g. VPN, VoIP) between natted endpoints requires an external publicly accessible node (for signalling and maybe also for data)

IPV6

- RFC2460–RFC3513–RFC4193
 - December 1998 !!
 - RFC 3363 and 3364 for DNS support (AAAA records)
- More Addresses
 - **128 bit** IP addresses
- *Stateless* autoconfiguration
- No checksums
- No router fragmentation
- Multicast/Anycast
- Headers extension (e.g. to use IPSec)

IPV6 address notation

- 128 bit → up to 8 group of 4 hex char
 - e.g. 2001:0db8:3c4d:0015:0000:0000:abcd:ef12
- Shorten as much as possible
 - e.g. 2001:0db8:3c4d:0015:0000:0000:abcd:ef12
 - becomes 2001:db8:3c4d:15::abcd:ef12
- In URI:
 - [https://2001:db8:3c4d:0015::abcd:ef12\]:443/](https://2001:db8:3c4d:0015::abcd:ef12]:443/)
- Use CIDR
 - e.g. 2001:cdba:9abc:5678::/64

Unicast and Anycast addr type

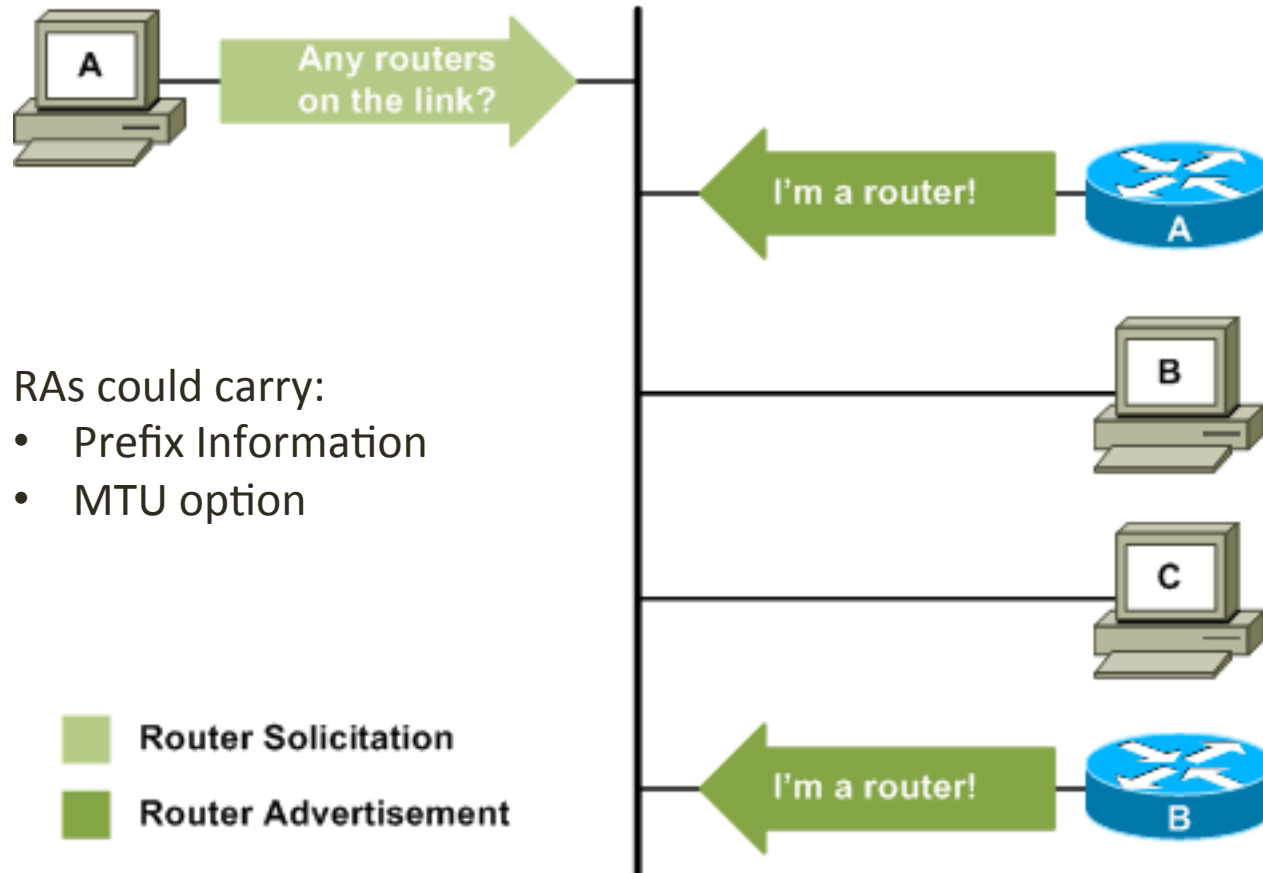
48 bit	16 bit	64 bit
routing prefix	subnet id	interface id



- Interface ID could be:
 - manually assigned
 - automatically generated (EUI-64, often derived by expanding mac address)
 - automatically assigned (DHCPv6)
 - randomly assigned

Stateless Autoconfiguration

- Rely on Neighbor Discovery Protocol (NDP) that use ICMPv6
 - and in particular on “Router discovery”

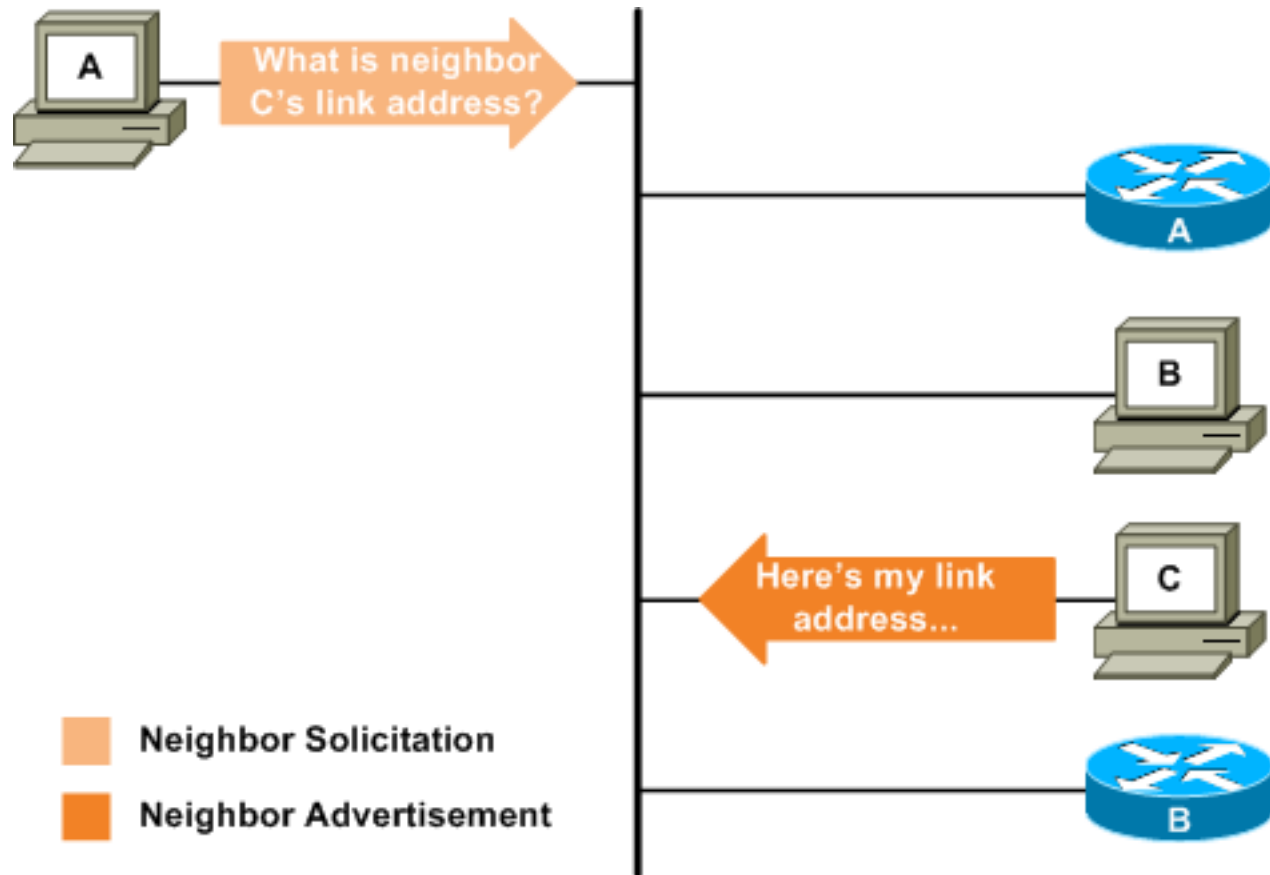


RAs could carry:

- Prefix Information
- MTU option

NDP: Address Resolution

- Carries:



Multicast and Neigh solicitation

- For each unicast address it responds to, each host listens on a *solicited-node multicast address*.
- The solicited-node multicast address for a given unicast address is constructed by taking the last three octets of the unicast address and prepending *FF02::1:FF00:0000/104*.
- Thus, the solicited-node multicast address of *2001:630:200:8100:02C0:4FFF:FE68:12CB* is *FF02::1:FF68:12CB*.
- It's the solicited-node multicast address that a node uses as the destination of a neighbour solicitation packet.
- This use of multicast means that most hosts don't get disturbed by neighbour solicitations that aren't either for them or for a host with a very similar IPv6 address.

NDP can also

- do redirection (as ICMPv4)
- do Duplicate Address detection (as gratuitous ARP)
- do Neighbor Unreachability Detection
- (as ICMPv4 host unreachable)

Link local / site local

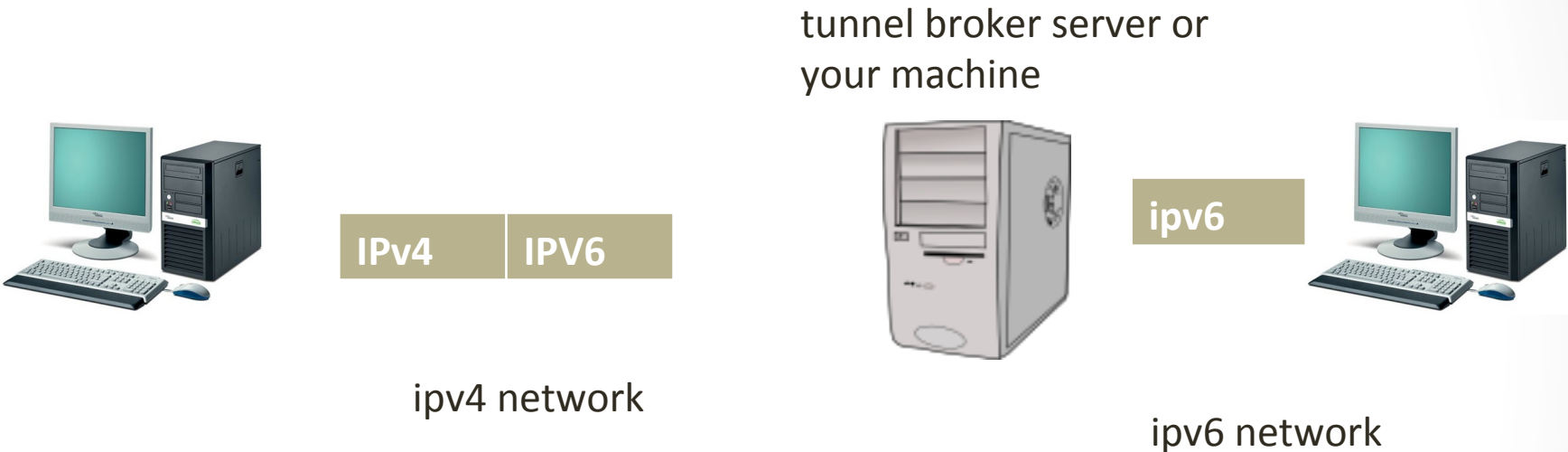
address	meaning
fe80::/10	link-local prefix are only valid and unique on a single link. Within this prefix only one subnet is allocated (54 zero bits)
fd00::/8	Unique local address (same role as IPv4 private address) (site local have been deprecated in RFC 3879)
ff00::/8	Multicast
::1/128	Loopback
::/96	IPv4-compatible IPv6 addresses.

How to change the IP version?

- Not “flag day” ...
- Worldwide testing and evaluation in the 6bone (1996-2006)
 - 3FFE::/16 addresses
- IPv6 has been in active production deployment since June 2006
- Now a lot of hosts/router are Dual stack
 - this does not means they are properly configured! 😊

6in4

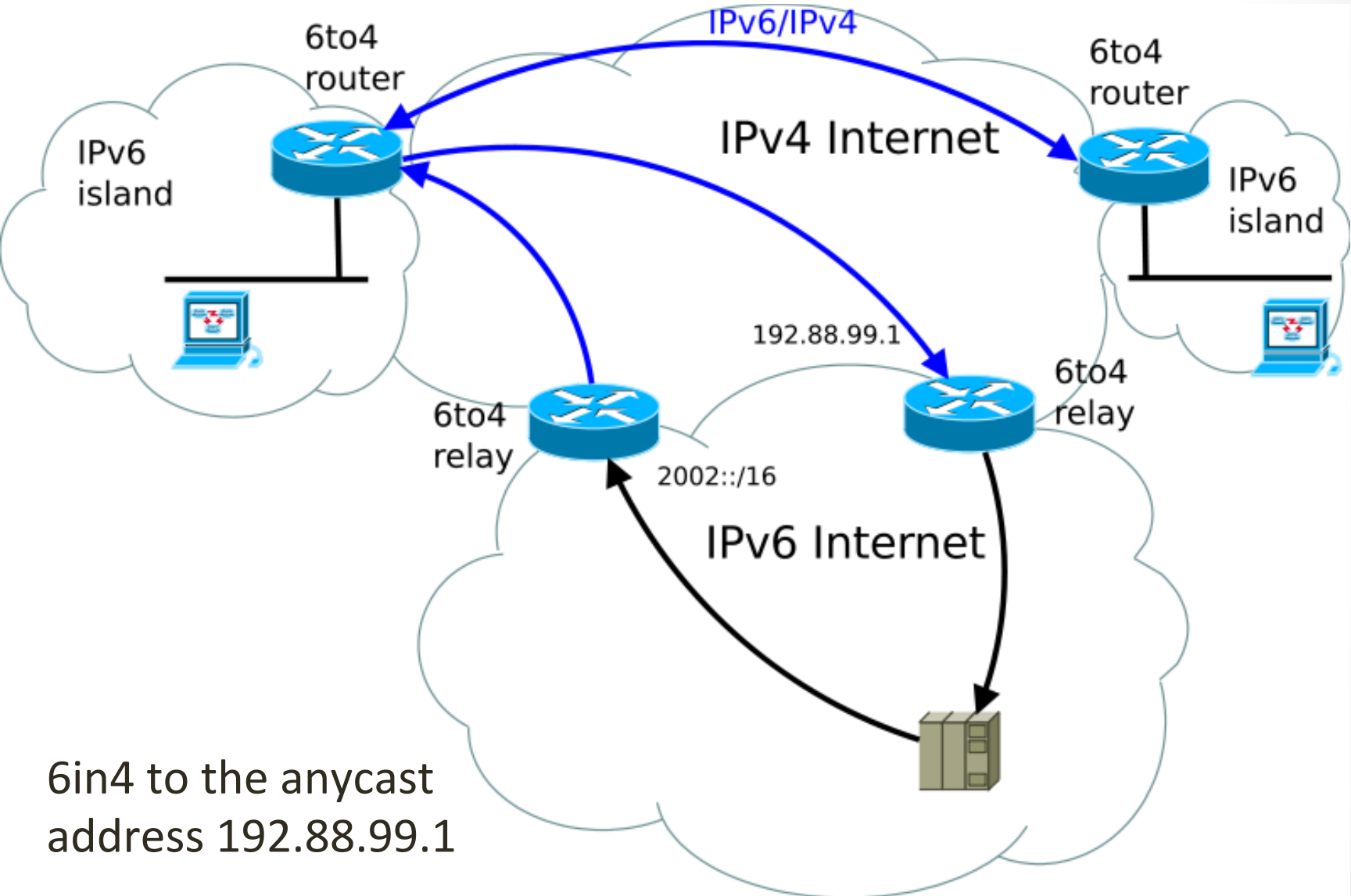
Allows ipv6 hosts to communicate using IPv4 network



- manually configure endpoints
 - require end-hosts configurations (some utilities make the job easier e.g. AICCU)
 - used by Tunnel Broker as SIXXs or Hurricane Electric
 - 6in4 is the IPv4 protocol 41
 - could use GRE (e.g. Hurricane Electric) or IPSec
 - **NAT unfriendly!!**

6to4

2002::/16



6in4 to the anycast address 192.88.99.1

6in4: Linux example

- `#!/bin/bash`
- `### Get the global IPv4 address for your host from the command line:`
- `GLOB_IP4=$1`
- `### Compute the 6TO4 tunnel IPv6 address:`
- `GLOB_IP6TO4=$(printf "2002:%x%02x:%x%02x::1" $(echo $GLOB_IP4 | tr . ' '))`
- `### Setup the tunnel`
- `ip tunnel add tun6to4 mode sit remote any local $GLOB_IP4 ttl 64`
- `ip link set dev tun6to4 up`
- `ip addr add $GLOB_IP6TO4/16 dev tun6to4`
- `ip route add 2000::/3 via ::192.88.99.1 dev tun6to4 metric 1`

Teredo

2001:0::/32

- Encapsulation on IPv4 – UDP – IPv6
 - uses port 3544
 - RFC 4380
 - “Microsoft Solution”, enabled by default from Windows Vista

Teredo

```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\administrator>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : mshome.net
    Link-local IPv6 Address . . . . . : fe80::2d2f:d205:7b93:edeax11
    IPv4 Address. . . . . : 192.168.137.147
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.137.1

Tunnel adapter isatap.mshome.net:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : mshome.net

Tunnel adapter iphttpsinterface:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Tunnel adapter Teredo Tunneling Pseudo-Interface:

    Connection-specific DNS Suffix  . :
    IPv6 Address. . . . . : 2001:0:c9b:a602:cfdbfe:f364:59fa
    Link-local IPv6 Address . . . . . : fe80::cfdbfe:f364:59fax14
    Default Gateway . . . . . : ::

C:\Users\administrator>_
```

... and more alternatives

- Anything in Anything (AYIYA) encapsulates IPv6 in UDP, TCP or even SCTP.
- NAT friendly
- used by SiXXs

IPv6 Tunnel Calculator

IPv6 is getting under steam recently - and IPv4-IPv6 tunnels see increasing usage. This little JavaScript page enables you to convert tunneled addresses to original ones (and vice versa).

6to4 tunnels ([Wikipedia DE](#) / [EN](#))

2002:

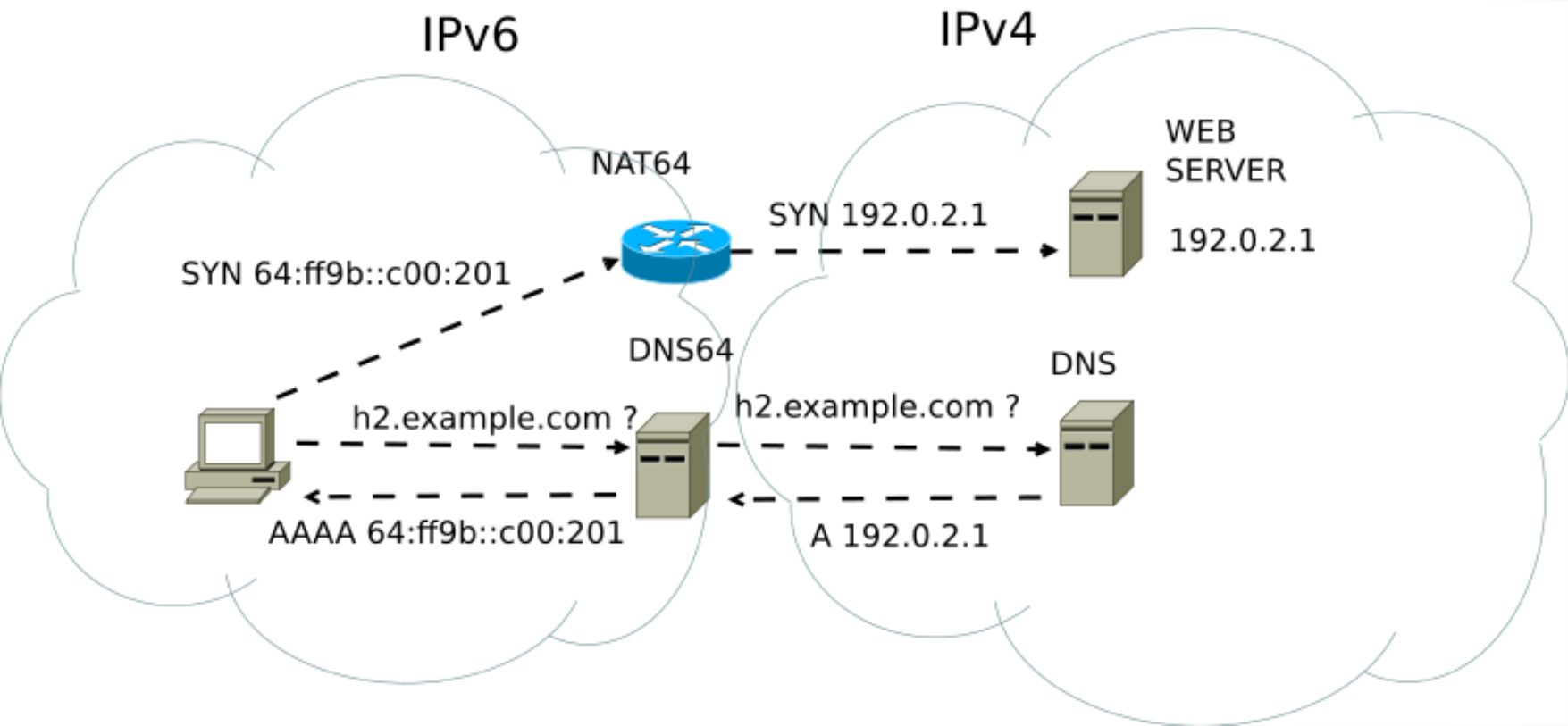
Teredo tunnel ([Wikipedia EN](#))

<http://www.wyae.de/docs/ipv6calc/>

2001:0: **Server:**
Client:
Port:

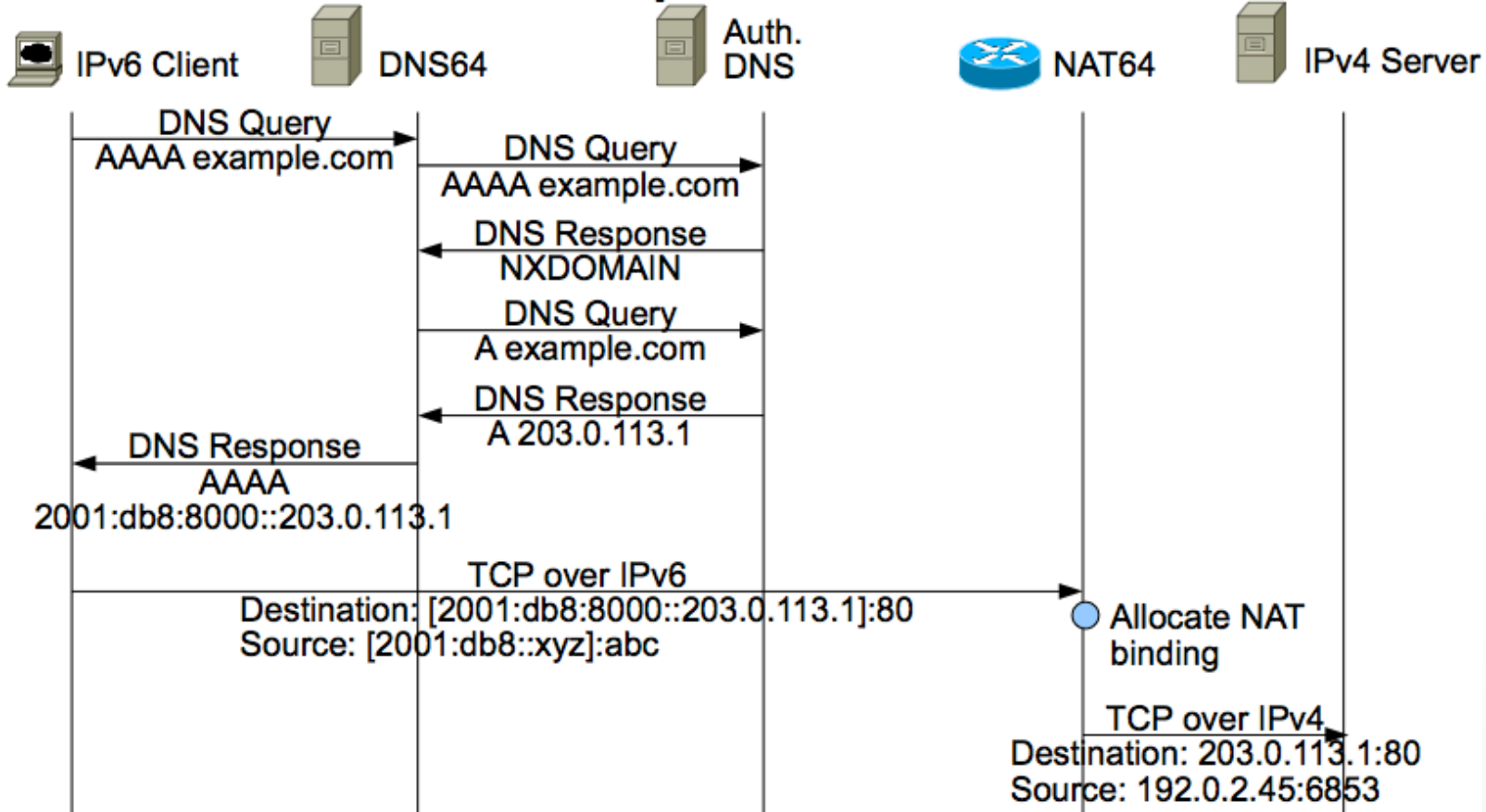
How IPv6 client could connect to IPv4 server?

- NAT64
 - allow IPv6 hosts to communicate with IPv4 servers
 - stateful or stateless



DNS64 and NAT64

Example



IPv6 Packets are simpler

- No checksum
 - Avoid not necessary processing
 - Avoid checksum recalculation when changing options
 - ...even from 1990 hardware technology has changed (now efficient hardware techniques for checksum)
- No fragmentation
- VERY big benefit where a state is needed
- Think of firewalls
 - Need to wait for all fragments before forwarding adding latency
 - Not all fragment are expected to flight on the same path

RADVD

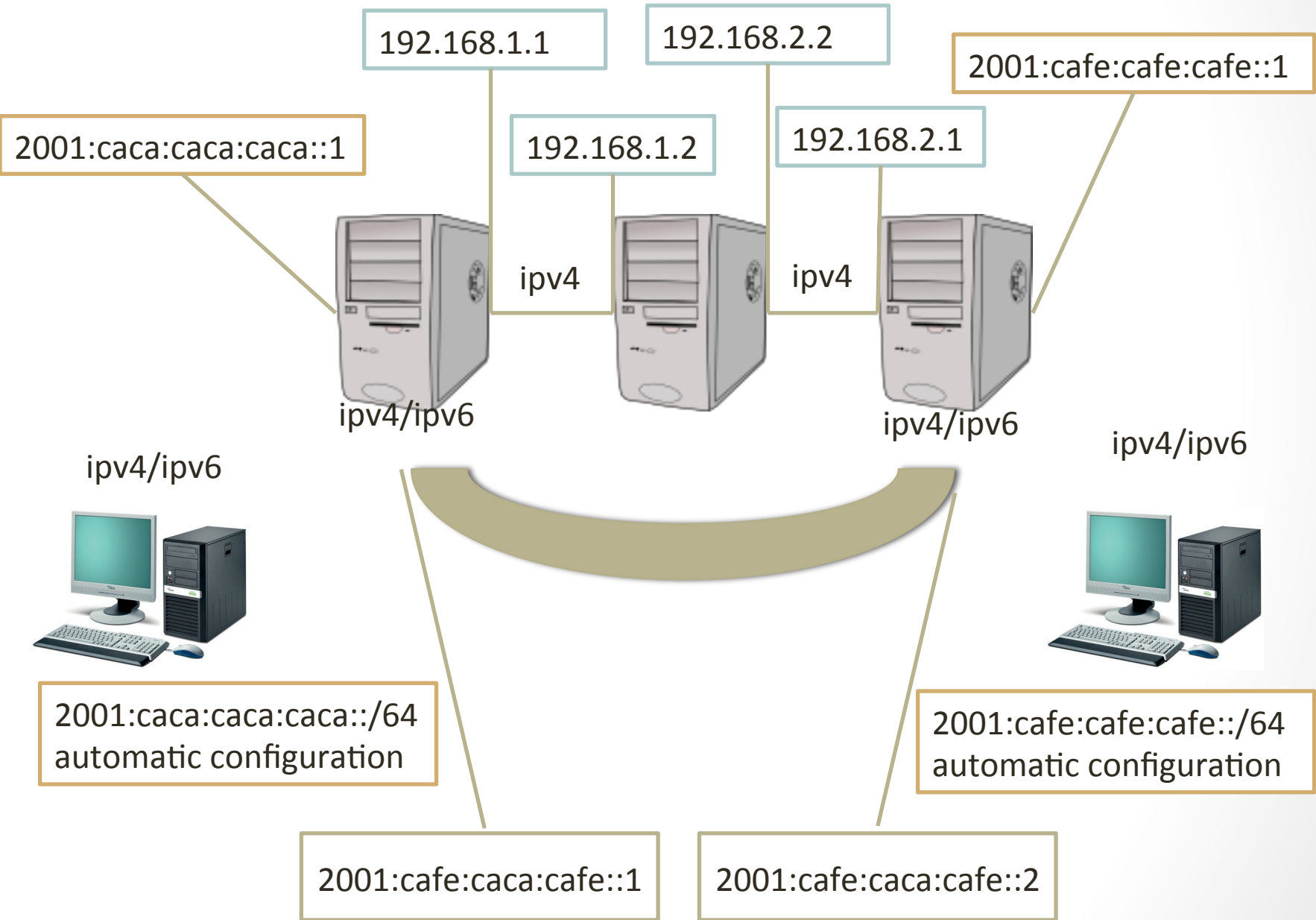
- Routing Advertisement Daemon in linux
- When IPv6 hosts configure their network interfaces, they broadcast router solicitation (RS) requests onto the network to discover available routers. The radvd software answers requests with router advertisement (RA) messages. In addition, radvd periodically broadcasts RA packets to the attached link to update network hosts. The router advertisement messages contain the routing prefix used on the link, the link maximum transmission unit (MTU), and the address of the responsible default router.

iproute2

- ip -6 *SAME_COMMANDS*

```
r1 login: root (automatic login)
Last login: Sun May 27 18:23:00 UTC 2012 on tty1
r1:~# ip -6 address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 2001:caca:caca:caca::1/128 scope global tentative
        valid_lft forever preferred_lft forever
    inet6 fe80::cab:f8ff:fe0c:104b/64 scope link
        valid_lft forever preferred_lft forever
4: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 fe80::f8de:dcff:fe30:9657/64 scope link
        valid_lft forever preferred_lft forever
r1:~#
```


Lab



Traces

- r1_ra.pcap
- encap_ipv6.pcap

```
r1:~# tcpdump -i any -vvv
tcpdump: WARNING: Promiscuous mode not supported on the "any" device
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size 96 bytes
L7:25:55.031557 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 56) fe80:
cab:f8ff:fe0c:104b > ip6-allnodes: ICMP6, router advertisement, length 56
    hop limit 64, Flags [none], pref medium, router lifetime 30s, reachable
time 0s, retrans time 0s[ndp opt]
L7:25:59.891377 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 56) fe80:
cab:f8ff:fe0c:104b > ip6-allnodes: ICMP6, router advertisement, length 56
    hop limit 64, Flags [none], pref medium, router lifetime 30s, reachable
time 0s, retrans time 0s[ndp opt]
L7:26:06.361376 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 56) fe80:
cab:f8ff:fe0c:104b > ip6-allnodes: ICMP6, router advertisement, length 56
    hop limit 64, Flags [none], pref medium, router lifetime 30s, reachable
time 0s, retrans time 0s[ndp opt]
```