8309804 Mobile Internet Technical Architecture

IPv6 and transition to IPv6

Heikki Vatiainen <hessu@cs.tut.fi>

Agenda

- MITA and IPv6
- IPv6 basics
  - packet format, addressing architecture
  - myths about security and Quality of Service
- Neighbor Discovery (ND) and autoconfiguration
  - this is where the new stuff is
  - the many duties of ND
  - about the security of ND
- Transition from IPv4 to IPv6
  - the three categories of transition techniques
  - Dual Stack, 6to4 and NAT-PT

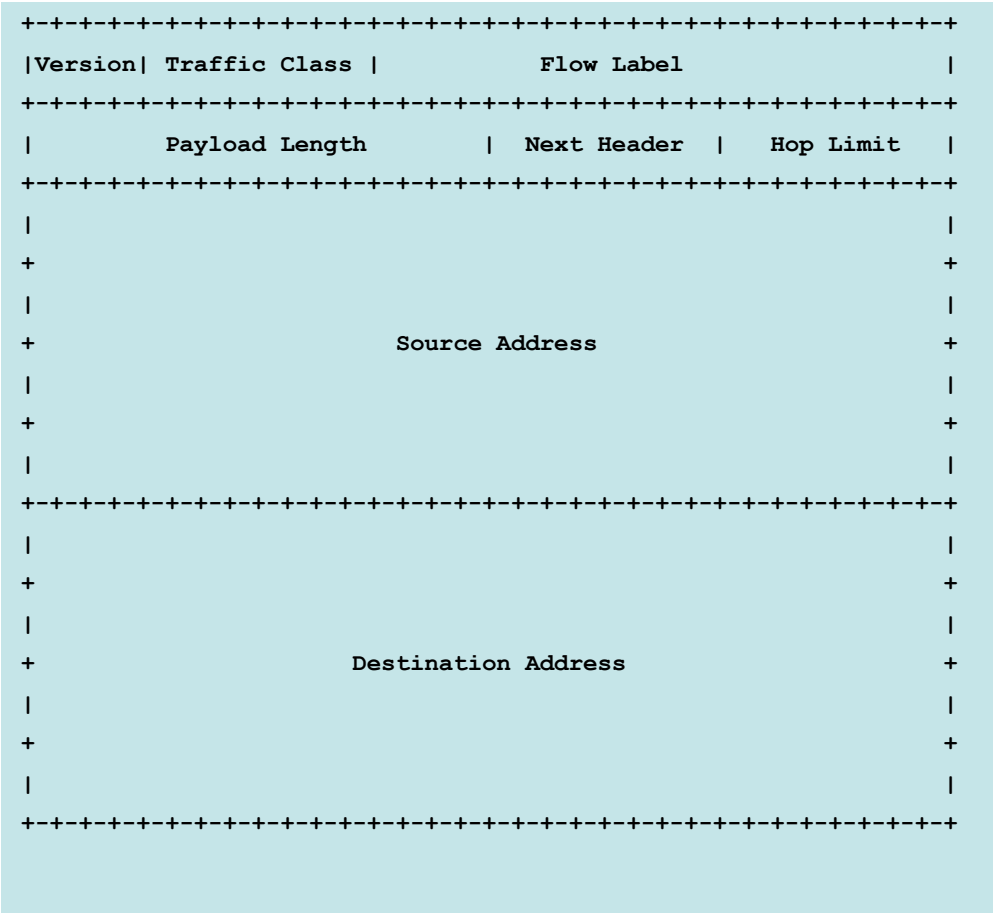
MITA and IPv6

- Mobile Internet Technical Architecture
- IPv6 is good for mobility
  - Enormous address space gives us room to concentrate on other things than saving address space.
  - Mobile IPv6 has learned from mobile IPv4
- 3GPP has chosen IPv6 for their IP Multimedia Subsystem architecture
- MITA aims for the future
  - Who *really* wants to bet that IPv4 will last for ever?

IPv6 basics

- Packet format
- Changes and improvements to IPv4
- Addressing architecture

Internet Protocol, Version 6



- RFC 2460, 39 pages
- Compared to IPv4:
  - Enhanced addressing
  - Simpler header
  - Better support for extensions and options
  - Flow Label and Traffic Class fields for QoS
    - Specs not quite complete yet
  - Mandatory IPSec
    - But does work without it

Enhanced addressing

- 128 bit address

  4 milliard * 4 milliard * 4 milliard * 4 milliard addresses or $(2^{32})^4$
- Address autoconfiguration
  - Stateless: RFC 2462 Stateless Address Autoconfiguration, 25 pages
  - Stateful: RFC 3315 DHCPv6, 101 pages
- New address type: anycast
  - Packet to anycast address is routed to the ”nearest” in the anycast group
  - IPv4 has it too
- Scoped multicast addresses
  - Scope examples: subnet, organization, world
  - The address includes the intented scope

Simpler header

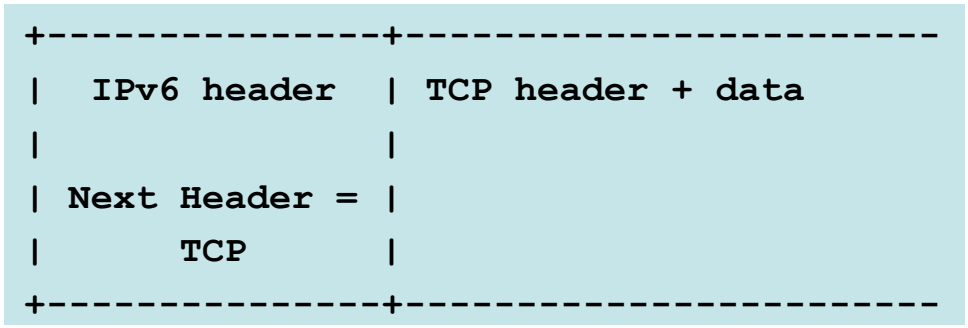
- Always 40 bytes, IPv4 *usually* 20 bytes
- Less fields
- No checksum
  - Next layer, TCP, UDP, ICMPv6 checksum contains parts from IPv6 header. Also, lower layers also do checksumming: why bother the routers with checking and recalculating IP layer checksum?
- Some missing fields have their own extensions headers, such as fragmentation and source routing
- IPv6 does have options, but they are carried in the Hop-by-Hop or Destination Options extension header

Extension headers

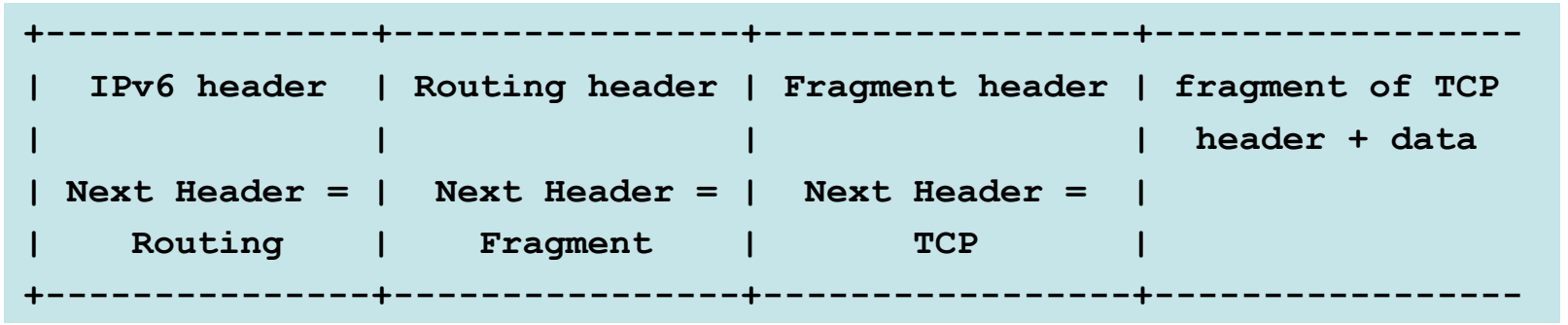
- Every extension header starts with 8 bit Next Header field
- RFC 2460 specifies the following extension headers
  - Hop-by-Hop options
  - Routing
  - Fragment
  - Destination Options
- Only the destination node examines the extension headers
- Except Hop-by-Hop Options
  - The name tells it all

Headers by example

1. The usual case: IPv6 + TCP. No extension headers



2. Fragmented TCP-segment using specified route



– IPv4 header always carries Identification, Fragment Offset and More Fragments information. IPv6 only does it when required.

QoS

- Enabling IPv6 does not automatically enable QoS
  - IPv6 protocol header has fields that can be used for QoS
  - Yes, this is play with words but notice the difference the words make
- Traffic Class field
  - Use of Traffic Class field was defined by IETF DiffServ (Differentiated Services) working group
  - RFC 2474 ”Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”
- Flow Label field
  - Definition and use of Flow Label field is still at draft state
  - For reasons such as IPsec encryption a flow is defined as a triplet of source and destination addresses and Flow Label
  - How to make use of Flow Label within a network is still a mystery

IPsec (1)

- Enabling IPv6 does not automatically enable security
  - IPsec has been defined to support both IPv4 and IPv6
  - IPv6 has been defined to work with IPsec from the start
- RFC 2401 ”Security Architecture for the Internet Protocol” tells more
- IPsec extension headers are
  - Authentication header (AH)
  - Encapsulation Security Payload header (ESP)
- Security Association (SA)
  - SA is a cryptographically secured one way connection
  - SA is identified by triplet <SPI, destination address, AH or ESP>
  - SPI is included in AH and ESP headers
- Just where is the global PKI (Public Key Infrastructure)?

IPsec (2)

- An example from ICMPv6, RFC 2463, section 5, ”Security Considerations”

```
ICMP protocol packet exchanges can be authenticated using the
IP Authentication Header [IPv6-AUTH].  A node SHOULD include an
Authentication Header when sending ICMP messages if a security
association for use with the IP Authentication Header exists for
the destination address.

The security associations may have been created through manual
configuration or through the operation of some key management
protocol.

...
```

Other comparison to IPv4

- Requires links to support minimum MTU of 1280 bytes
  - If the links has MTU less than 1280, the interfaces must still support MTU of 1280 bytes
- There are no broadcast addresses
- Multiple address prefixes on the same link are supported
  - Enables configuring of addresses from multiple ISP:s
- There is **no** change in UDP ja TCP
  - In practice some changes in code is needed because programs use IP addresses. Need to accommodate the bigger addresses
  - RFC 3493 ”Basic Socket Interface Extensions for IPv6”, 39 pages
  - RFC 3542 ”Advanced Sockets API for IPv6”, 77 pages

Addressing architecture (1)

- RFC 3513 ”IP Version 6 Addressing Architecture”, 26 pages
  - Defines the textual format of address and address prefixes
  - Defines unicast, anycast and multicast addresses
  - Defines the required addresses of IPv6 nodes
- Textual format of address
  - 16 bit groups, separated with a colon, possibly packing a group of zeroes with double colon (::)
  - For example ::1 and 2001:708:310:52:203:baff:fe08:fdd
- Textual format of address prefix
  - Similar to what IPv4 CIDR uses. For example:
    - TUT IPv6 prefix is 2001:708:310::/48
    - Node address and subnet 2001:708:310:52:203:baff:fe08:fdd/64

Addressing architecture (2)

- Each network interface has:
  - At least one link-local unicast address
  - Possibly **multiple** unicast and anycast addresses
  - Membership of multiple multicast groups
- The addresses belong to network interfaces
  - Just like in IPv4
  - We still have not separated the node identifier and locator

Addressing architecture (3)

- The type of IPv6 address is given in the most significant bits

Address type	Binary prefix	IPv6 notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-local unicast	1111111010	FE80::/10
Site-local unicast	1111111011	FEC0::/10
Global unicast	(everything else)	

Dead

- Anycast addresses come from unicast address space
  - One can not tell if an address is anycast or unicast address just by looking at it. Anycast address is configured to a node.
  - Anycast addresses can come from any scope
- Site locals were recently deprecated by IETF IPv6 WG

Addressing architecture (4)

- IANA (Internet Assigned Numbers Authority) is responsible for the IPv6 address space
  - http://www.iana.org/assignments/ipv6-address-space
- Currently all global IPv6 unicast addresses start with bits 001
  - The rest of the global unicast addresses (85%) are reserved for future use
- The table on the next page is from RFC 3513

Addressing architecture (5)

Allocation	Prefix (binary)	Fraction of Address Space	
Unassigned (see Note 1 below)	0000 0000	1/256	
Unassigned	0000 0001	1/256	
Reserved for NSAP Allocation	0000 001	1/128	[RFC1888]
Unassigned	0000 01	1/64	
Unassigned	0000 1	1/32	
Unassigned	0001	1/16	
Global Unicast	001	1/8	[RFC2374]
Unassigned	010	1/8	
Unassigned	011	1/8	
Unassigned	100	1/8	
Unassigned	101	1/8	
Unassigned	110	1/8	
Unassigned	1110	1/16	
Unassigned	1111 0	1/32	
Unassigned	1111 10	1/64	
Unassigned	1111 110	1/128	
Unassigned	1111 1110 0	1/512	
Link-Local Unicast Addresses	1111 1110 10	1/1024	
Site-Local Unicast Addresses	1111 1110 11	1/1024	
Multicast Addresses	1111 1111	1/256	

link-local addresses

- FE80::/10
- Mandatory for each node
- Always available
- Not routeable
- Useful for:
  - autoconfiguring other addresses
  - communicating with other nodes on the same link when stateful or stateless address configuration for global addresses is not available
  - address of default gateway is its link local address

```
|   10     |
|  bits    |         54 bits           |           64 bits              |
+----------+---------------------------+--------------------------------+
|1111111010|            0              |       interface ID             |
+----------+---------------------------+--------------------------------+
```

global unicast addresses

- RFC 3513 general format is given below
- To make a long story short, what we have now in practice is:

n == 48, m == 16

- Size of n being 48 is the most common case by far
- n+m being 64 i.e. 64 bit size subnet is practically immutable
- If needed, blocks of size /64 and /128 can be given to e.g. by ISP to customers

```
|         n bits          |   m bits   |       128-n-m bits          |
+-------------------------+------------+-----------------------------+
| global routing prefix   | subnet ID  |       interface ID          |
+-------------------------+------------+-----------------------------+
```

Addressing example: Solaris 8

```
%sudo ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
eri0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 130.230.52.79 netmask ffffff80 broadcast 130.230.52.127
        ether 0:3:ba:8:f:dd
lo0: flags=2000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 1
        inet6 ::1/128
eri0: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
        ether 0:3:ba:8:f:dd
        inet6 fe80::203:baff:fe08:fdd/10
eri0:1: flags=2080841<UP,RUNNING,MULTICAST,ADDRCONF,IPv6> mtu 1500 indx 2
        inet6 2001:708:310:52:203:baff:fe08:fdd/64
```

- Interface IDs of link-local unicast and global unicast addresses are derived from node's 48 bit MAC address

Neighbor Discovery (ND)

- RFC 2461 ”Neighbor Discovery for IP version 6 (IPv6), 93 p.
  - Replaces ARP
  - Works on top of ICMPv6.
  - Enables address autoconfiguration
  - Performs duplicate address detection
- Most important functional change in IPv6 as compared to IPv4
  - IPv6 is plug-and-play/zeroconf to end nodes (hosts)
  - The most important change in IPv6 vs IPv4 are the big addresses
- ND is common name for many different protocols
- Only some of ND’s functionality can be found from IPv4
- ND uses link-local addresses in many ways
  - ND is the most important user of l-l addresses
  - For example: the default gateway address the end node uses is the gateways l-l address

ND's functionality (1)

- Router Discovery
  - How end nodes (hosts) find the routers
- Prefix Discovery
  - How hosts learn the prefixes their interfaces must use
- Parameter Discovery
  - For example link MTU (Maximum Transmit Unit) and IP TTL (Time To Live)
- Address Autoconfiguration
  - Should stateless or stateful or combination of both be used when configuring the interface addresses
- Address Resolution
  - If the IP address of an on-link peer is known how its link level address is found
  - This is what replaces ARP that IPv4 uses

ND's functionality (2)

- Next-hop determination
  - Algorithm for finding the next hop IP
- Neighbor Unreachability Detection (NUD)
  - Method for finding out if the link neighbor is still alive
  - Useful e.g. for switching to another default gateway
- Duplicate Address Detection (DAD)
  - Before an address is used DAD is run
  - Done for stateless, stateful and manually configured addresses
- Redirect
  - For finding a better on-link next hop
  - Usually redirect is used for telling the host to use another router for packets leaving the link

ND, autoconfig and security

- ND is based on nodes sending and replying to advertisements
  - Simple to advertise wrong information or answer with false replies
- Anyone can send router advertisements and pretend to be a router
  - After that the packets from gullible hosts can be inspected and forwarded to a real router
- Anyone can reply to Duplicate Address Detection probes
  - Denial of service attack
- Anyone can advertise false address prefixes
  - Denial of service attack
  - Seen a couple of times in TUT/ICE test network when someone booted up a router configured to a different network
- IPsec is not a cure-all (at least now)
  - How do you establish SA with IKE (i.e. automatically without extensive manual keying), if you first need to have SA so that the packets are accepted by the link peers (chicken-egg problem)
  - IETF SEND (Secure Neighbor Discovery) WG has started work on securing ND without extensive manual keying

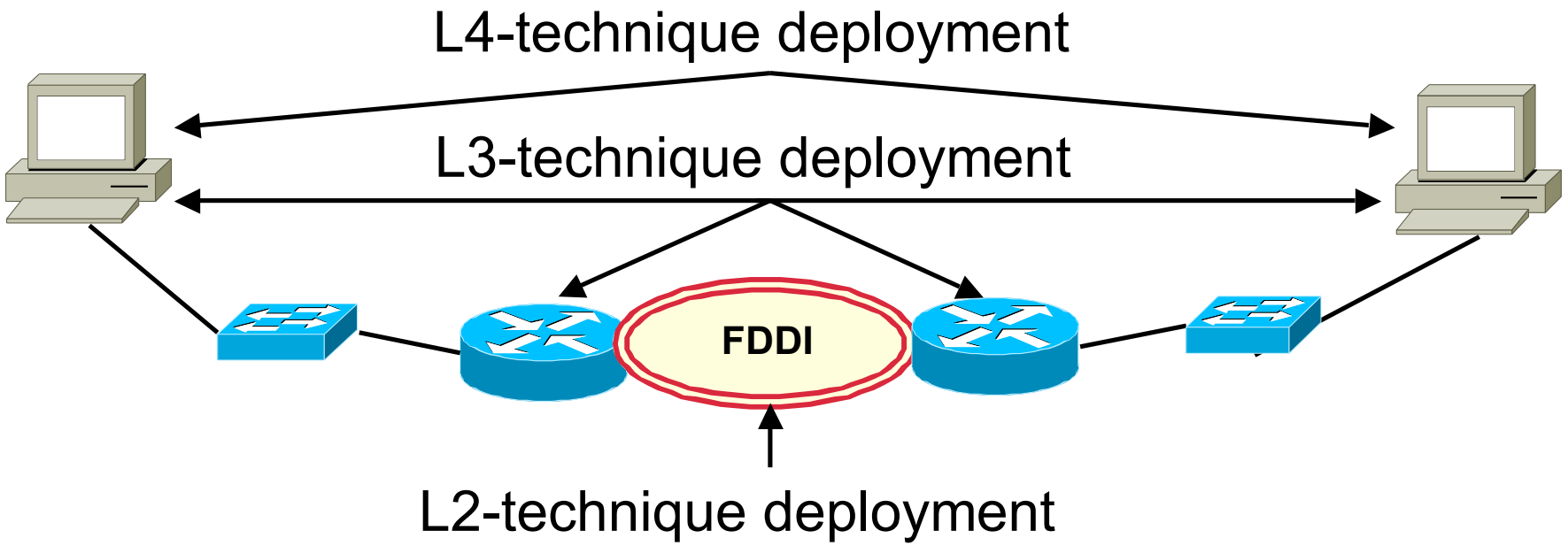
Transition to IPv6

- Reasons
- Problems
- Techinques
  - Dual Stack
    - Running both IPv6 and IPv4 simultaneously
  - Tunneling
    - Configured and automatic
    - Example: 6to4
  - NAT-PT
    - Network Address Translation - Protocol Translation

About IPv6 transition

- Purpose to make transition from IPv4 to IPv6 as painless as possible
- IETF did have a WG for specifying transition techniques, ngtrans, which concluded Feb 2003
- The work is now continued by IPv6 operations (v6ops) WG
  - ngtrans specified the techniques, v6ops is there to work on making the deployment and operations in mixed v4/v6 environment feasible
- Co-operation with IPv4 is important
  - the two must co-exists easily
  - many transition techniques use the existing IPv4 infrastructure
- The transition will last tens of years
  - maybe forever

What makes transition difficult



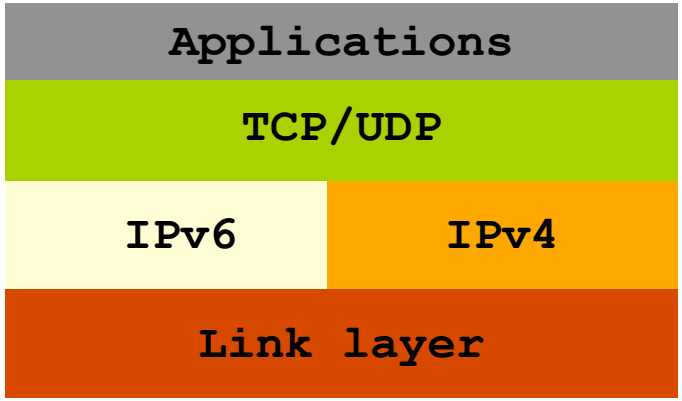
- IP is on OSI layer 3. Change layer 3 and the changes touch all layer 3 devices (read: the whole Internet)
  - L4 is for end nodes only
  - L2 is one link and its attached devices
  - L3 is the endpoints and routers (and firewalls, NAT boxes, ...)

Types of techniques

- Three types of transition techniques
- Dual Stack
  - Both IPv4 and IPv6 are available in a node
  - May or may not know about each other
- Tunneling
  - IPv6 in IPv4 in the beginning
  - Maybe IPv4 in IPv6 later
- NAT-PT
  - Conversion of IPv6 packets to IPv4 and vice versa
  - Done by the packet sender or in a router between the sender and receiver

Dual Stack

- IPv6 and IPv4 running simultaneously in the same box
- Dual stack router can route IPv6 and IPv4 concurrently
- Dual stack end node can choose its poison based on e.g. a DNS response
  - this is a mixed blessing, choose carefully
  - for example *BSD, Mac OS X, Linux, Solaris, Win2K and WinXP have dual stacks. This lets them to tunnel IPvX over IPvY.



Tunnels: Configured tunneling

- RFC 2893 ”Transition Mechanisms for IPv6 Hosts and Routers”, 29 pages
  - Specifies e.g. manually configured tunnels (Configured Tunneling)
  - New version draft-ietf-v6ops-mech-v2-00.txt removes many obsoleted mechanisms
- Enable connecting IPv6 islands across IPv4 internetwork
  - Also IPv4 over IPv6 is possible
- High management overhead, both tunnel ends must be manually configured
- Stable and well understood
- RFC 3053 ”IPv6 Tunnel Broker”, 13 pages
  - TB service providers provide tunnels, DNS and semiautomated method for setting up a tunnel. Currently widely available.

Tunnels: 6to4 (1)

- RFC 3056 ”Connection of IPv6 Domains via IPv4 Clouds”, 23 pages
- Enables big and small organisations to interconnect between other 6to4 sites and IPv6 Internet
  - Big could be e.g. an university and small a standalone host
- Requires one global IPv4 address
  - This can belong to a NAT box which is IPv4 NATting the organisation
  - IPv6 hosts within the organisation have global IPv6 addresses
  - IPv4 hosts may or may not have global IPv4 addresses, depending NAT’s presence
- 6to4 user gets a /48 sized IPv6 network
  - That is what ISPs generally give
  - Makes subnetting works just like with ”real” (non 6to4) prefixes

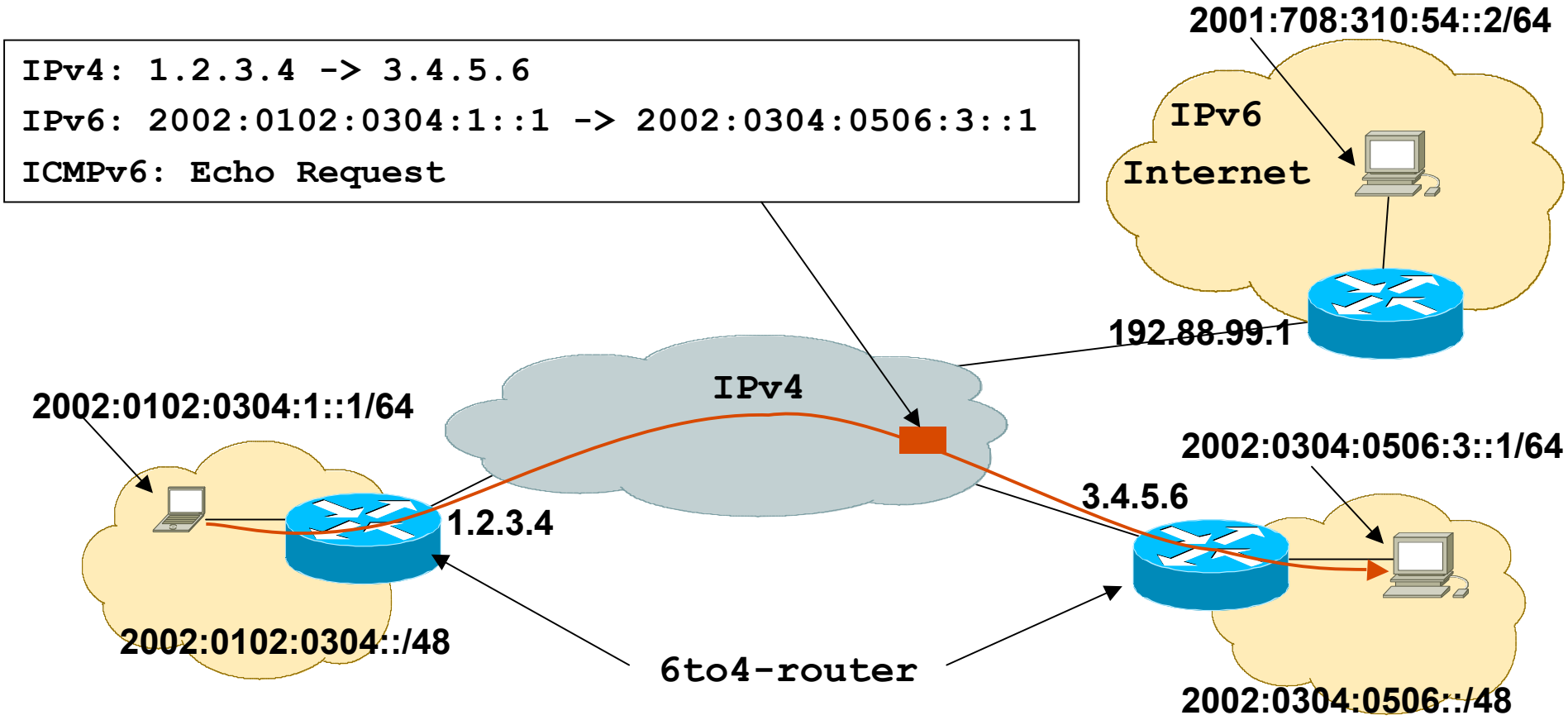
6to4 (2)

- 6to4 in four steps
  - Get a 6to4 capable router and choose one of its global IPv4 addresses. Call it **V4ADDR**
  - Compose a 6to4 prefix by concatenating 2002:: with **V4ADDR**, see the picture below
  - You now have a global /48 size IPv6 prefix for your 6to4 site
  - Create subnets using the /48 prefix
- For example from 130.230.54.113 one gets 2002:82E6:3671::/48
- Terms FP, TLA ja SLA are from RFC 2374 which is Historic

```
| 3 |  13  |    32     |   16   |          64 bits                |
+---+------+-----------+--------+---------------------------------+
|FP | TLA  | V4ADDR    | SLA ID |         Interface ID            |
|001|0x0002|           |        |                                 |
+---+------+-----------+--------+---------------------------------+
```

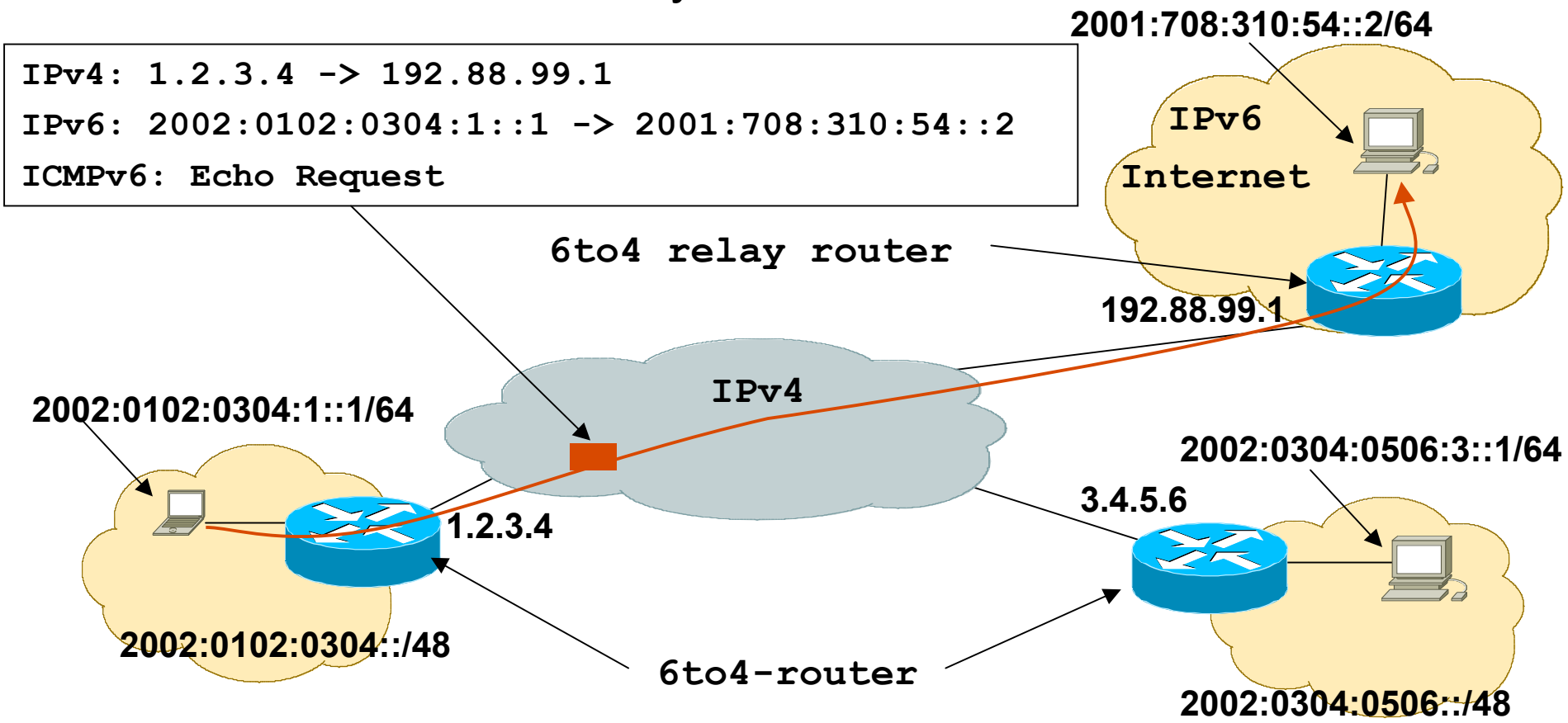
6to4 example (1)

- The simple case: both connection endpoints are using 6to4
- 6to4 routers tunnel IPv6 packets encapsulated in IPv4



6to4 example (2)

- 6to4 relay router connects 6to4 site to IPv6 Internet
- 6to4 routers use the relay as their default router



NAT-PT

- RFC 2766 ”Network Address Translation - Protocol Translation (NAT-PT)”, 21 pages
- Defines protocol translation between IPv6 and IPv4
  - Aimed at devices that only know about one or the other, but need interprotocol connectivity (IPv6 <--> IPv4 or IPv4 <--> IPv6)
  - Converts (mangles) IPv6 headers to IPv4 headers and vice versa
  - Based on IPv4 NAT
- Has problems with DNS
  - NAT box must also do application level DNS mangling
  - IP addresses within DNS data must also be translated
- Has the same problems as IPv4 NAT (and some of its own)
  - problems for IPsec
  - problems with applications that exchange IP addresses
  - http://www.cs.utk.edu/~moore/what-nats-break.html

Conclusions

- IPv6 works better than IPv4 with mobility
  - More addresses
  - Better mobility support
- IPv6 has working implementations
- IPv6 has working multicasting, routing, DNS, IPsec, etc.
- IPv6 needs to co-exist with IPv4
  - A lot of IPv4 has been deployed
- IPv6 can co-exist with IPv4
  - Transition techniques for the rescue
- It is unlikely IPv4 becomes the only technique that will never die
  - But how long will it live on?