The IPv6 Protocol

Mauro Tortonesi
<mauro@deepspace6.net>

Limits of good old IPv4

- The current version of IP protocol presents some serious limits:
  - Address space exhaustion
  - Explosion of routing tables
  - Mobility
  - Performance and scalability
- These are structural problems and cannot be fixed!!!

Address space exhaustion

- Extremely inefficient address assignment policy
- Until 1993 address assignment was in fixed-sized blocks: 16.776.214 (class A), 65.534 (class B) or 254 (class C) units
- These blocks are too big or too small for the effective needs of organizations
- Enormous address waste
- Once there will be no addresses available, the Internet could not grow anymore, except using...

Network Address Translation

- Don't do it!!! NAT is plain evil!!!
- Single point of failure
- Breaks end-to-end topology of TCP connections
- Problems with Mobile IP, VPN, FTP, TCP TIME_WAIT state etc...
- Complicates the use of multi-homing
- NATs enable casual use of private addresses
- False sense of security (NAT is not a packet filter)
- And much more!!! (see RFC 2993 - Architectural Implications of NATs)

Explosion of routing tables

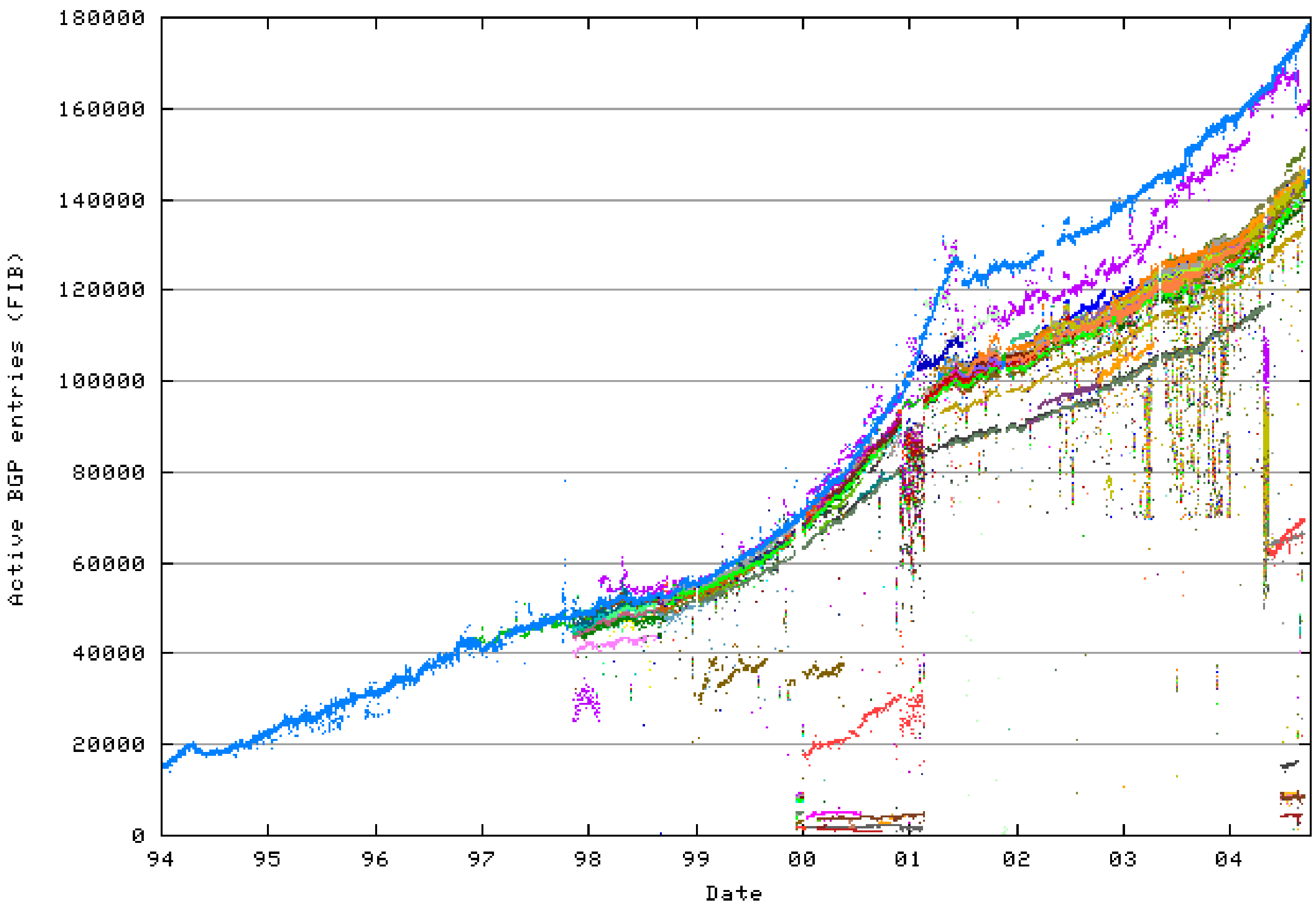
- IPv4 address space is not aggregatable
- Size of routing tables depends from the number of networks connected to the Internet
- As the routing tables grow larger, complexity of routing process increases and performance drops
- May lead to routing stability problems (often very hard to fix!!!)

Classless Inter-Domain Routing

- More efficient address assignment: contiguous class C blocks allocated on geographic basis
- Creation of an aggregatable subspace in the IPv4 address space
- Routing decisions are made according to “longest matching prefix” rule
  - if packet dest is 5.6.7.8 and router has both 5.6.0.0/16 and 5.6.7.0/24 routes, the packet will be sent to the 2nd one
- Smaller routing tables
- Short-term solution that extends IPv4 lifetime

BGP RT size

- Picture taken from http://bgp.potaroo.net



(Data Gathered from AS1221 and Route-Views)

Problems in Mobility and Scalability

- Mobile IP is not a good solution for mobile computing
  - Performance problems (triangle routing)
  - Need to deploy FAs in visited networks
  - No security
  - Problems with NAT
- IPv4 has not been designed to be scalable
  - Complex header w/o common case optimization
  - Fragmentation
  - Inadequate min (576) and max (64KB) packet size

Future scenario

- New markets are developing in the ICT sector:
  - Personal / Mobile Devices
  - Networked Entertainment
  - Device Control
- IPv4 cannot satisfy the requirements posed by these new markets

The IPv6 solution

- New version of the Internet Protocol
- Well-defined evolution of IPv4
- Devised by IETF to replace IPv4
- It solves many of the problems with IPv4
- It is ready for mainstream adoption

The IPv6 protocol

- Enormous address space
- Aggregatable address space
- Mandatory IPSEC support
- Advanced autoconfiguration functions
- Improved mobile networking
- High levels of performance and scalability

Enormous address space

- IPv6 addresses are 128 bits long
- 2^128 (~ 340*10^36) possible addresses
(340,282,366,920,938,463,463,374,607,431,768,211,456)
- Address space 2^96 (~ 79*10^27) times bigger than IPv4 one (79,228,162,514,264,337,593,543,950,336)
- 665,570,793,348,866,943,898,599 addresses per square meter on Earth!!!
- IPv6 allows the Internet to grow with exponential pace for the next 30 years (RFC1715, RFC3194)

More on IPv6 addresses - 1

- Hexadecimal packed representation:
  - FEDC:BA98:7654:3210:FEDC:BA98:7654:3210
  - Sequences of zeros are shortened to "::" (e.g. 1080:0:0:0:0:0:200C:417A == 1080::200C:417A)
  - "::" can be used only once for each address
- In mixed IPv4 and IPv6 environments it may sometimes be useful an hybrid notation:
  - ::FFFF:5.6.7.8 (IPv4-mapped address)
  - ::5.6.7.8 (IPv4-compatible address)
  - 2002:5.6.7.8::1 (6to4 address)

More on IPv6 addresses - 2

- IPv6 addresses are assigned to network interfaces
- Three different address tipologies:
  - Unicast
  - Anycast
  - Multicast
- Four addressing scopes:
  - Link local (FE80::/10)
  - Site local (FEC0::/10) NOW DEPRECATED!!!
  - Global, Unicast & Routable (2000::/3)
  - Multicast (FF00::/8)

More on IPv6 addresses - 3

- 6 (out of 16) multicast scopes:
  - interface-local (useful only for loopback)
  - link-local
  - admin-local
  - site-local
  - organization-local
  - global
- Examples
  - FF05::2 (all routers on site)
  - FF02::1 (all nodes on link)

More on IPv6 addresses - 4

- Special type addresses:
  - Unspecified address (::/128)
  - Localhost address (::1/128)
  - IPv4-mapped (::FFFF:V4ADDR/96)
  - IPv4-compatible (::V4ADDR/96)
  - 6to4 (2002:V4ADDR::/48)
  - Solicited node multicast address (used for DAD)
  - Subnet-router anycast address
  - Many, MANY, ***__MANY__*** more...

Aggregatable address space

- Routing decisions are made according to the "longest matching prefix" rule
  - if packet dest is 2001:abcd::1 and router has both 2001::/16 and 2001:ab::/24 routes the packet will be sent to the 2nd one
- Routing table size optimization
- Format of global routable unicast IPv6 address:

3	13	8	24	16	64 bits
FP	TLA ID	RES	NLA ID	SLA ID	Interface ID

Mandatory IPSEC support

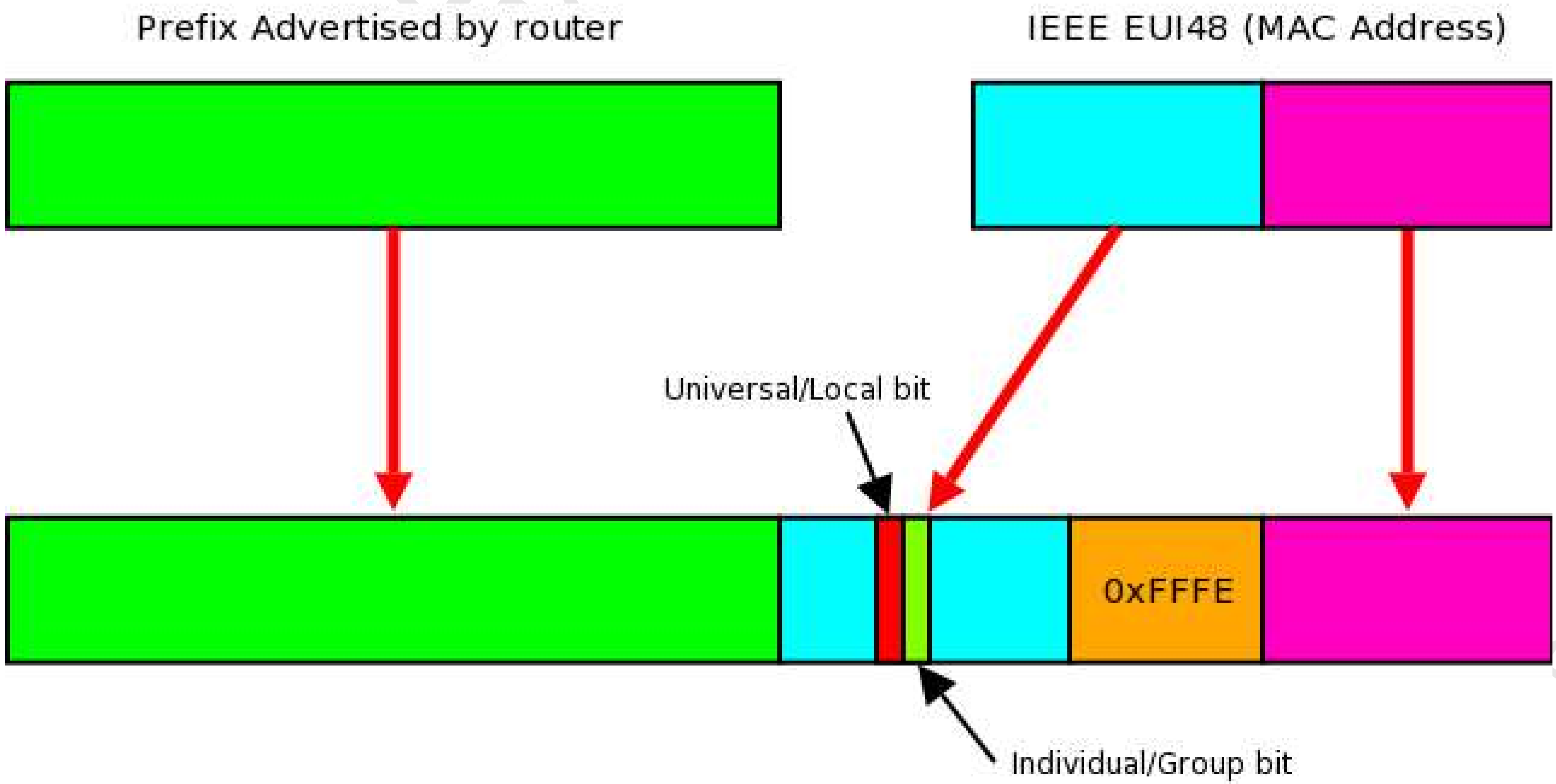
- Support for authentication and privacy
- IPv6 has native IPSEC support
  - AH (Authentication Header)
  - ESP (Encapsulated Security Payload)
- Securing data at network layer is sometimes better than doing it at transport layer (e.g. SSL)
- Allows the creation of VPNs (Virtual Private Networks)

Advanced autoconfiguration functions

- Two different kinds of address autoconfiguration for networking interfaces
  - DHCPv6
  - Stateless Address Autoconfiguration
- Support for (almost) automatic renumbering
- Minimization of human intervention costs

Stateless Address Autoconfiguration

- Prefix of link is obtained by router advertisement
- Interface ID is obtained by MAC address



Improved mobile networking - 1

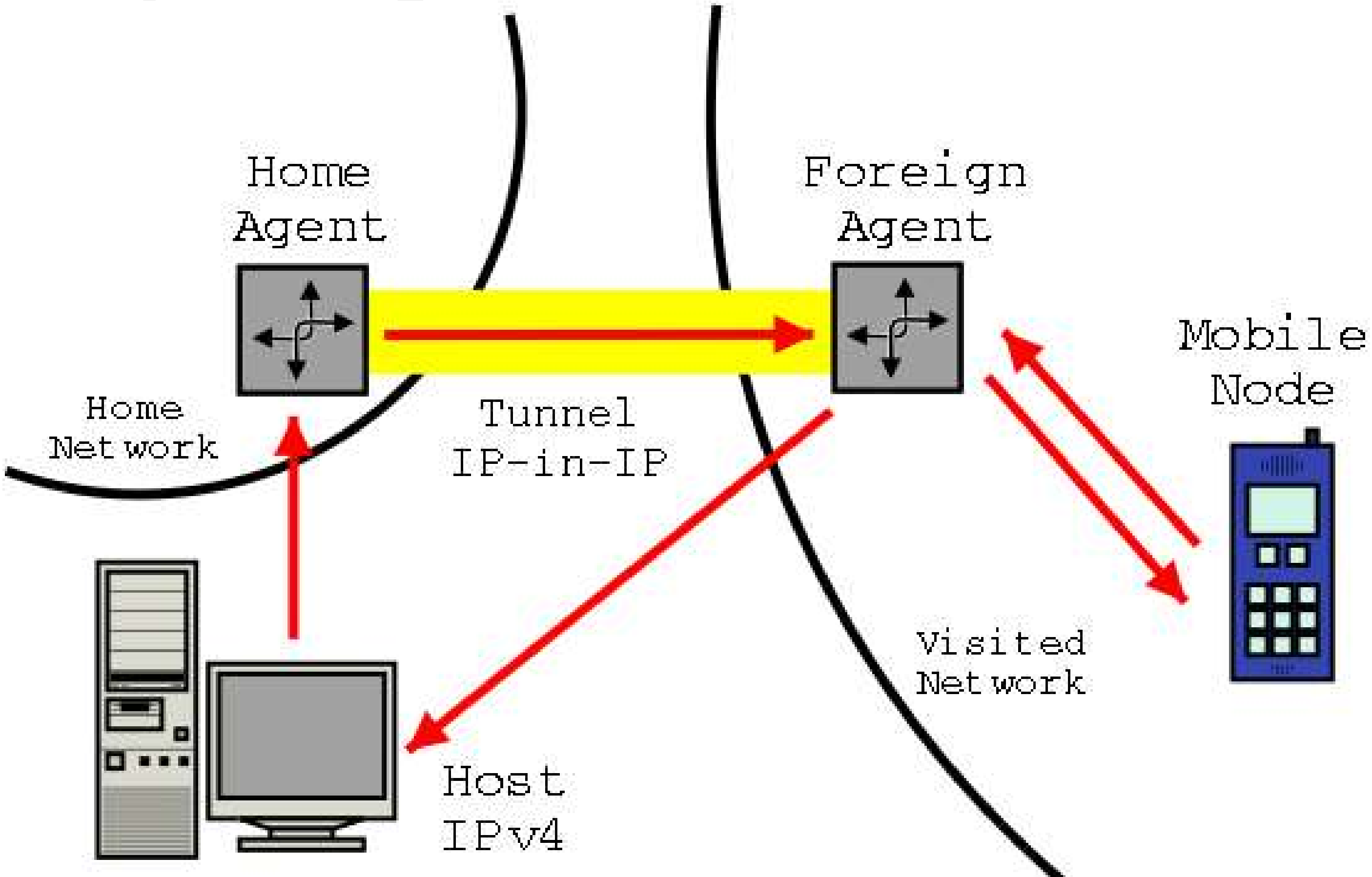
- Mobile IPv6 is more performant than Mobile IP ...
  - Route optimization included in the protocol
  - Use of Routing Header instead of encapsulation
  - Dynamic Home Address Discovery uses anycast and returns a single response to the mobile node
  - Large use of piggybacking thanks to Dest. Options
- ... more secure ...
  - Mandatory use of IPSEC
  - Packet filtering is easier to perform

Improved mobile networking - 2

- ... more robust and flexible
  - Use of Neighbor Discovery instead of ARP
  - Better support for multicast traffic
  - No more Foreign Agents
  - Bidirectional movement detection mechanism
  - New “Advertisement Interval” option on Router Advertisements

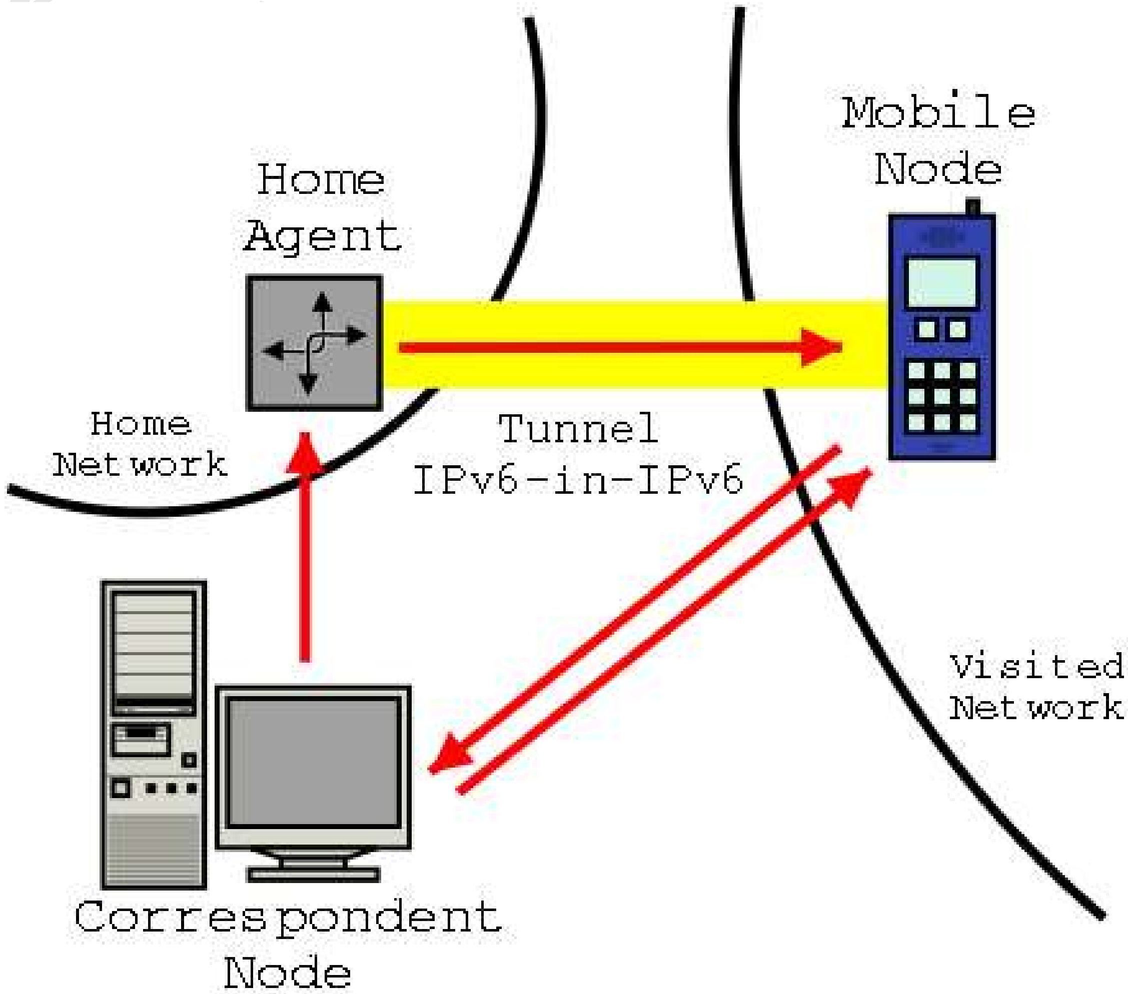
Improved mobile networking - 3

- Mobile IP operations:



Improved mobile networking - 4

- Mobile IPv6 operations:

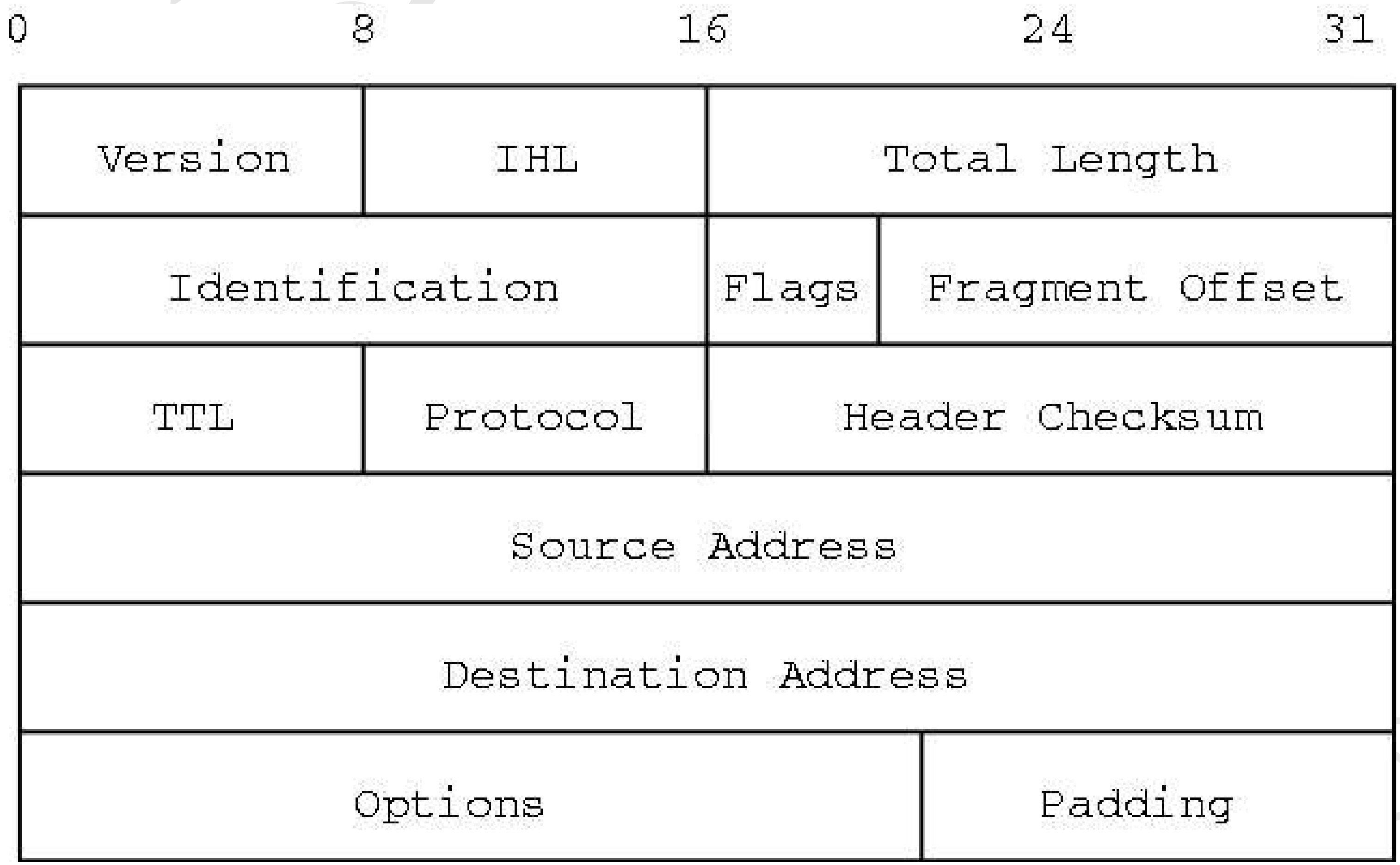


High levels of performance and scalability - 1

- IPv6 headers are only 2 times bigger than IPv4 ones, even if IPv6 addresses are 4 times longer
- No more checksum at network layer
- Handling IP options is easier
- Fragmentation is made only on source host, not from routers along the communication path
- Lower overhead than IPv4

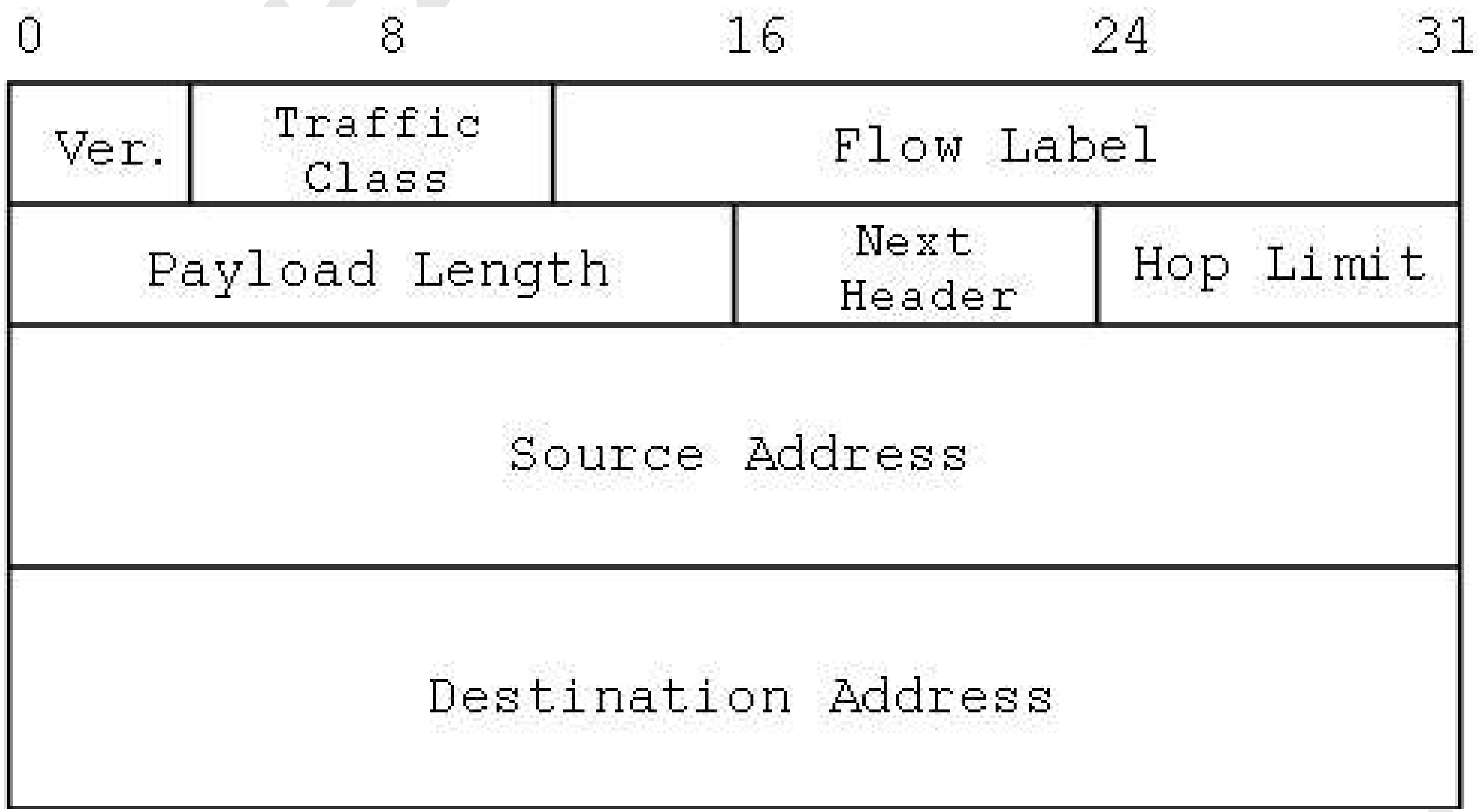
High levels of performance and scalability - 2

- IPv4 headers are complex (variable length, many fields):



High levels of performance and scalability - 3

- IPv6 headers are simpler and have 64-bit aligned fields:



High levels of performance and scalability - 4

- IPv6 outperforms IPv4
- Protocol implementation is easier and cheaper (especially on small battery-operated devices)
- IPv6 can make full use of
  - High bandwidth/performance networks
  - Computational capabilities of supercomputers in highly-distributed computation environments

The transition to IPv6 - 1

- All nodes (hosts, routers, firewalls, L3 switches, etc...) must be upgraded in order to support IPv6
- IPv6 connectivity must be provided to LANs and WANs
- All applications must be ported to IPv6
- IPv6 nodes and applications should preserve compatibility with IPv4
- Very difficult task!!!

The transition to IPv6 - 2

- The transition will be a long and delicate process
- It must be completed before the total collapse of IPv4 routing and addressing capabilities
- To have a successful (or not too painful) transition we need:
  - High interoperability between IPv4 and IPv6 host
  - Maximum flexibility in the deployment of an IPv6 node in an IPv4 network
  - Easy migration from IPv4 to IPv6 services

The transition scenario - 1

- During the transition phase we'll have mixed IPv4 and IPv6 environments
- Many networks won't have native IPv6 connectivity
- Transition tools and mechanisms will be deployed to provide IPv6 connectivity to hosts and LANs (6TO4, NAT-PT, etc...)
- The network scenarios will be very complex
- Applications must be designed to work in all possible environments

The transition scenario - 2

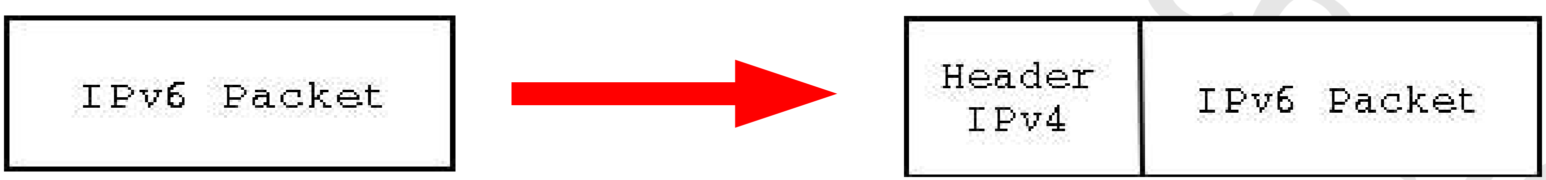
- During the transition we'll have:
  - nodes with IPv4 connectivity but no IPv6 connectivity (or support)
  - nodes with IPv6 connectivity but no IPv4 connectivity (or support)
  - nodes with both IPv4 and IPv6 connectivity
- IPv4 connectivity may be preferred to IPv6 connectivity or viceversa (cost, reliability, etc...)
- There may be problems with DNS resolution

Towards the IPv6 Internet

- So far the native IPv6 routing infrastructure is not very extended
- There may not be a native routing path between two IPv6 peers
- A possible solution is to use the existing IPv4 infrastructure to route IPv6 packets
- We can build virtual IPv6 links with IPv6-in-IPv4 tunnels

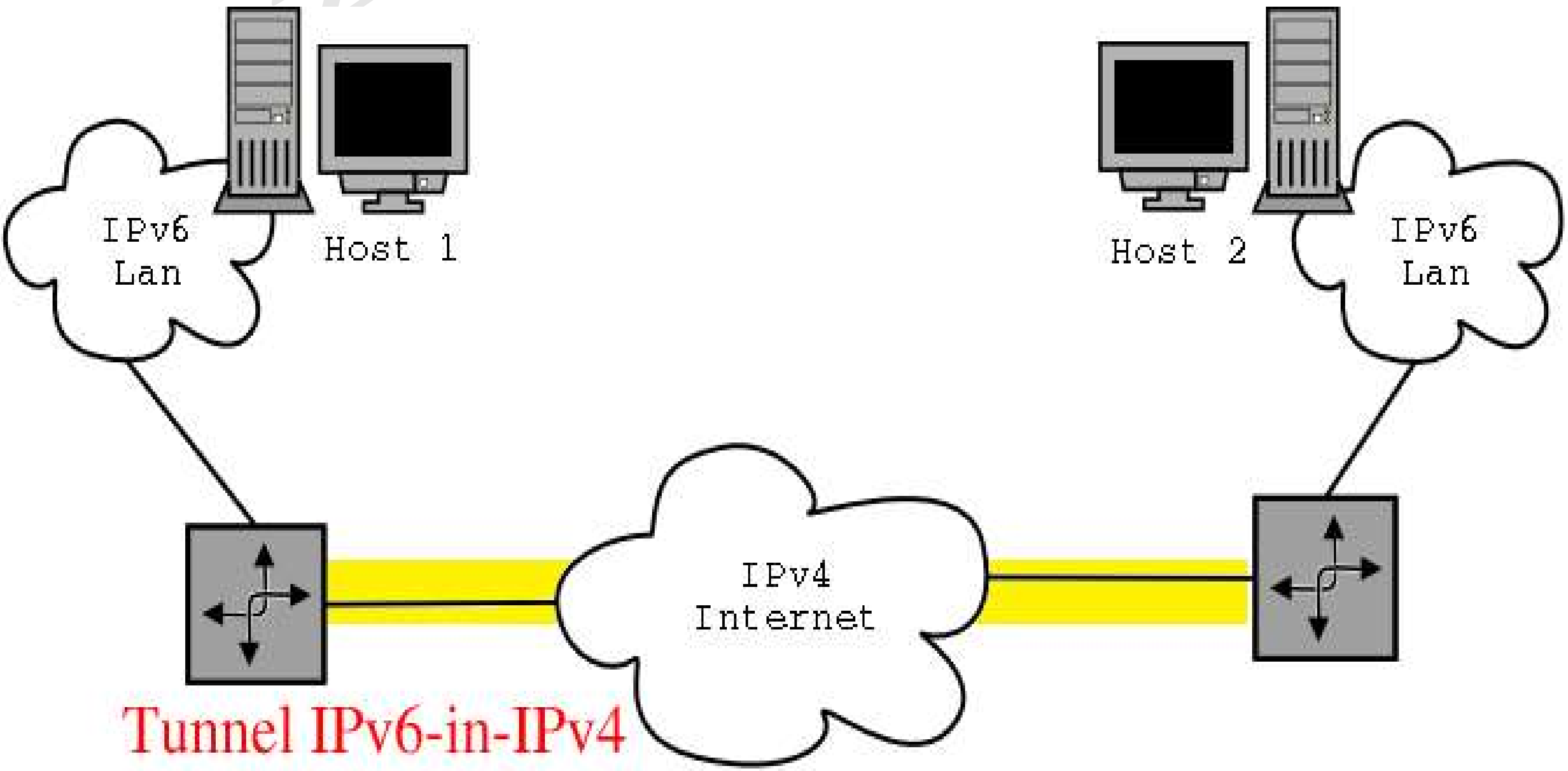
IPv6-in-IPv4 tunnels - 1

- IPv6 packets are encapsulated in IPv4 packets and then routed to destination via the IPv4 Internet inftastructure
- Once they arrive to destination, IPv6 packets are first decapsulated and then processed by IPv6 stack



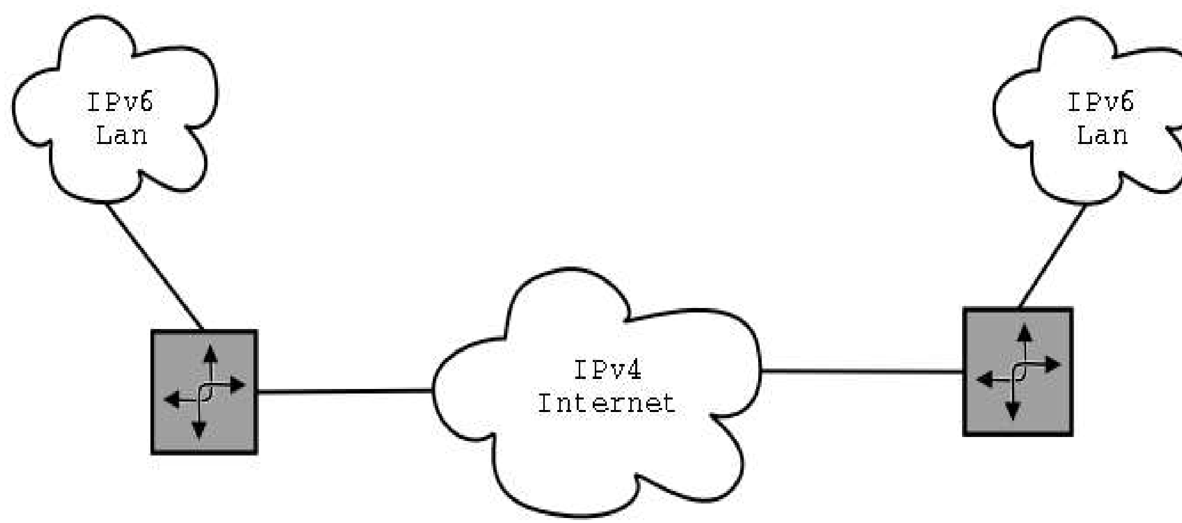
IPv6-in-IPv4 tunnels - 2

- Typical LAN-to-LAN tunnel usage:



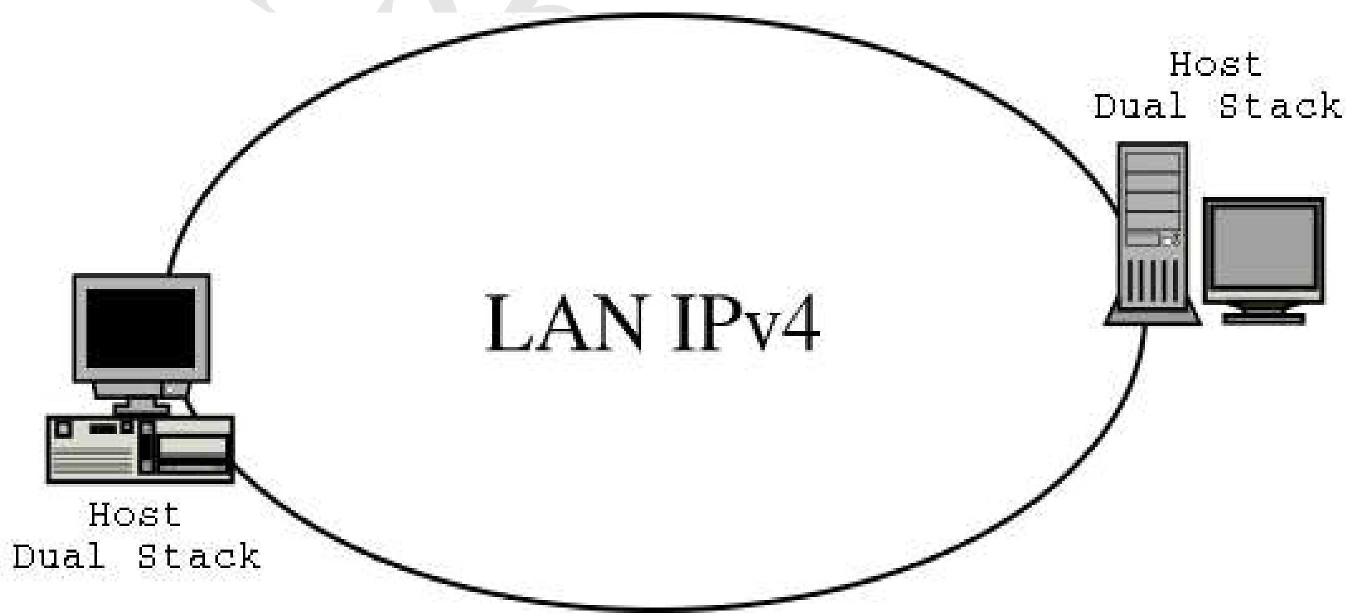
Transition techniques - 1

- Connection between IPv6 native "islands":
  - Manually configured tunnels
  - Automatic tunnels
  - Tunnel Brokers
  - BGP Tunnels
  - 6TO4



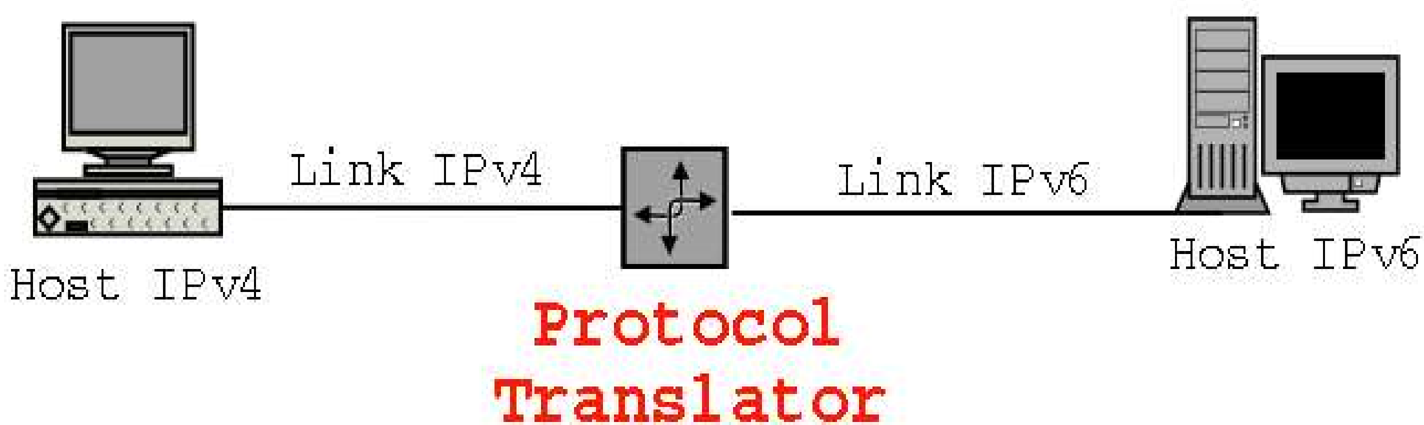
Transition techniques - 2

- Connection between IPv6 hosts inside an IPv4-only LAN:
  - ISATAP
  - 6OVER4



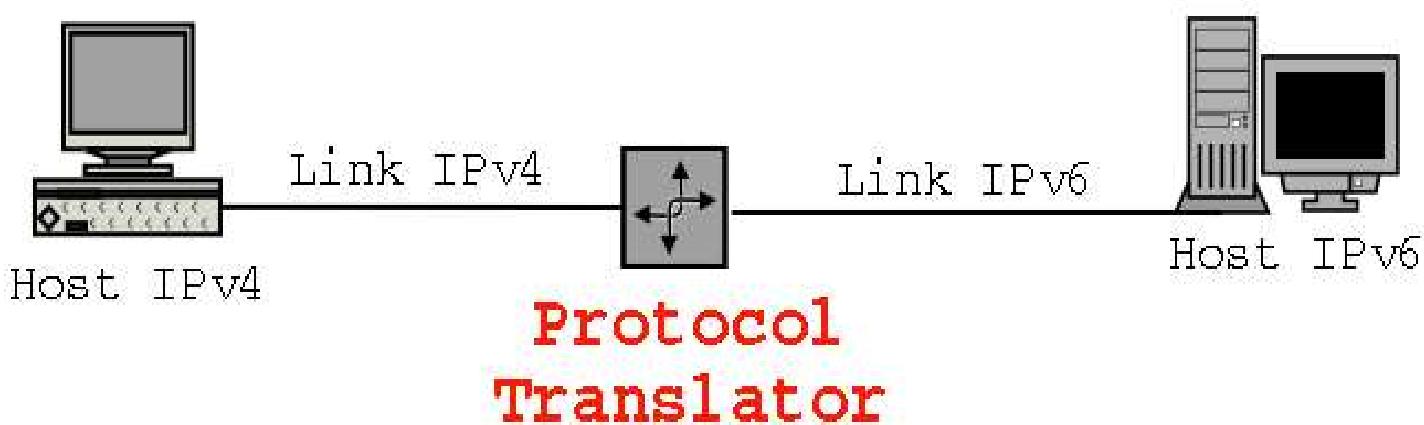
Transition techniques - 3a

- Communication between IPv4-only and IPv6-only nodes:
  - Dual stack model / Limited Dual stack model
  - SOCKS64
  - NAT-PT
  - SIIT



Transition techniques - 3b

- Communication between IPv4-only and IPv6-only nodes:
  - BIS (Bump-In-The-Stack) / BIA (Bump-In-The-API)
  - Transport Relay Translator
  - DSTM (Dual Stack Transition Mechanism)
  - Shipworm/Teredo



IPv6 and DNS

- Name to address resolution
  - AAAA records (simple extension of IPv4 A records)
  - A6 records (experimental, supports multihoming and renumbering but is a potential security threat)
- Address to name resolution
  - IN6.INT domain (deprecated but still used)
  - IN6.ARPA domain

Backward compatibility

- To support the new protocol, all existing TCP/IP software must be modified
- Many operating systems already offer a rather advanced IPv6 support
- Many applications altrady support IPv6
- Commercial routers (Cisco, Juniper) already offer (at least partial or experimental) IPv6 support
- There is still a lot of work to do

Linux & IPv6 - Status

- The Linux kernel has IPv6 support since 1998
- All the major distros have an IPv6-enabled kernel
- Glibc is IPv6-enabled (apart from RPC)
- Most of the applications are IPv6-enabled (see the IPv6 Status Page for applications at DS6)
- USAGI project (http://www.linux-ipv6.org) distributes a patch for both userspace applications and kernel

Linux & IPv6 - Basic setup - 1

- Check if your distro has IPv6 support
  - *test -f /proc/net/if_inet6 && echo "IPv6 support"*
- If IPv6 support has been compiled as a module
  - *modprobe ipv6*
  - *echo "alias net-pf-10 ipv6" >> modprobe.conf*
- You may need to recompile your kernel
  http://www.deepspace6.net/docs/best_ipv6_support.html

Linux & IPv6 - Configuration - 1

- Use ip addr for manual configuration of IPv6 addressess:
  - ip -6 addr add 2001:dead:beef::1/64 dev eth0
  - ip -6 addr del 2001:dead:beef::1/64 dev eth0
- Use ip route for manual configuration of IPv6 routes:
  - ip -6 route add 2000::/3 via 2001:dead:beef::1
  - ip -6 route del 2000::/3 via 2001:dead:beef::1
- Use ip neigh for manual configuration of neighbours (just like static ARP cache in IPv4)

Linux & IPv6 - Configuration - 2

- Use ip tunnel for manual configuration of IPv6 tunnels

ip tunnel add sit1 mode sit ttl default_ttl remote ipv4_remote_endpoint local ipv4_local_endpoint
ip link set dev sit1 up
ip -6 route add ipv6_prefix dev sit1 metric 1

- To manually delete a tunnel

ip -6 route del ipv6_prefix dev sit1
ip link set sit1 down
ip tunnel del sit1

Linux & IPv6 - Testing

- Many testing and debugging tools are available
  - netstat from net-tools
  - ping6 from iputils
  - traceroute6 and tracepath6 from iputils
  - nmap
  - tcpdump and/or ethereal
- nc6 from netcat6 can be very useful for testing applications & services
  - echo “GET http://myhost” | nc6 myhost 80

Linux & IPv6 - DNS - 1

- BIND supports IPv6

```
options {
        # to listen also on IPv6
        listen-on-v6 { none; };
};

acl example-acl {
        127.0.0.1;
        172.24.0.0/16;
        2001:dead:beef::/64;
        ::1/128;
        ::ffff:172.24.0.104/128;
};
```

Linux & IPv6 - DNS - 2

- Direct resolution

$ORIGIN mynetwork.org.
myhost IN AAAA 2001:dead:beef::1

- Inverse resolution using IP6.INT (nibble based)

$ORIGIN 0.0.0.0.f.e.e.b.d.a.e.d.1.0.0.2.ip6.int.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR myhost.mynetwork.org

- Inverse resolution using IP6.ARPA (bitstring based)

$ORIGIN \[x2001deadbeef0000/64].ip6.arpa.
\[x0000000000000001/64] IN PTR myhost.mynetwork.org

Linux & IPv6 - Radvd

- Use radvd to advertise network prefix

```
interface eth0 {
        AdvSendAdvert on;
        MinRtrAdvInterval 3;
        MaxRtrAdvInterval 10;
        prefix 2001:dead:beef::/64
                AdvOnLink on;
                AdvAutonomous on;
                AdvRouterAddr on;
        };
};
```

Linux & IPv6 - Web Servers

- Apache 2 supports IPv6
- Works with VirtualHost, too!

```
Listen [2001:dead:beef::1]:80
Listen 5.6.7.8:80
<VirtualHost [2001:dead:beef::1]:80 5.6.7.8:80>
    ServerName myhost.mynetwork.org
    ...
</VirtualHost>
```

- Many other HTTP server support IPv6 (boa, thttpd, webfs, bozohttpd, etc...)

Linux & IPv6 - Web Clients

- Many web browsers/clients support IPv6
  - Mozilla
  - Konqueror
  - Opera
  - lynx, elinks
  - wget, httrack, cURL, etc...
- IPv6 address can be used on URLs (RFC2372)
  - http://[2001:dead:beef::1]/path
- Not all web proxies are IPv6 enabled

Linux & IPv6 - FTP

- Most FTP server support IPv6
  - vsftpd
  - pure-ftpd
  - proftpd
  - oftpd, etc...
- Many FTP clients support IPv6
  - lftp
  - ncftp
  - wget, etc...

Linux & IPv6 - Email - 1

- Many SMTP servers natively support IPv6
  - Sendmail
  - Exim
  - Courier
- Patches to add IPv6 support to many SMTP servers are available
  - Postfix
  - Qmail

Linux & IPv6 - Email - 2

- Many IMAP/POP servers natively support IPv6
  - dovecot
  - courier
- Most email clients support IPv6
  - mutt
  - kmail
  - sylpheed (also sylpheed-claws)
  - mozilla-mail
  - fetchmail

Linux & IPv6 - Programming Languages

- Linux supports the Extended BSD Socket API for socket-based applications written in C
- Perl supports IPv6
  - NET::Socket6
  - IO::INET6
- Python supports IPv6
- Ruby supports IPv6
- Java supports IPv6

Linux & IPv6 - Misc services

- Other famous IPv6 enabled packages:
  - openssh
  - openldap
  - xinetd
  - netkit-inetd+tcpd
  - xntpd
  - X Windows (both XFree86 and X.org)

So, when will IPv4 die?

- Always too late ;-)
- There are areas in which the shortage of IPv4 addresses is really dramatic (especially Asia)
- However, IPv4 is not going to disappear soon:
  - http://potaroo.net/2003-08/ale.html
- NAT, private networks and Realm IP will extend lifetime of IPv4
- The transition to IPv6 will be probably very long

Conclusions

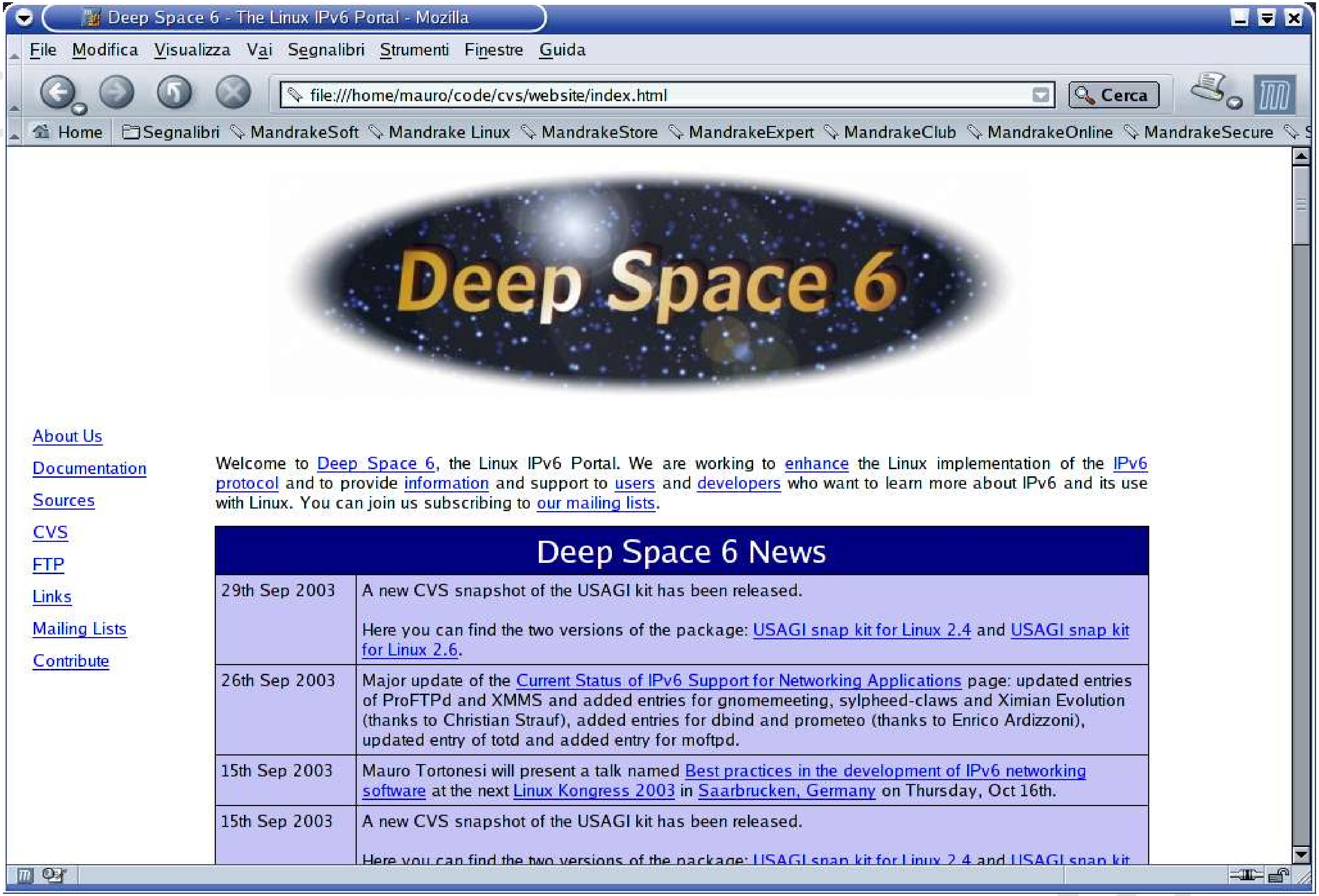
- IPv6 is going to mainstream, so consider starting the migration of your network, applications and services to IPv6 RIGHT NOW!!!

Suggestions

- Read the Linux IPv6 HOWTO
- Visit http://www.deepspace6.net
  - read the documentation
  - subscribe to the ds6 mailing list
  - maybe subscribe to the ds6-devel mailing list
- Start the migration to IPv6
  - ask your ISP for IPv6 connectivity
  - subscribe to a free tunnel broker service
  - setup 6to4 on your network

Acknowledgements

- Simone Piunno (co-founder of Deep Space 6)
- Peter Bieringer (co-founder of Deep Space 6)
- Chris Leisham (co-author of netcat6)
- Filippo Natali (co-author of netcat6)



http://www.deepspace6.net

Questions?

IPv6 FAQs

- What happened to IP version 5?
- Weren't 64 bits enough for IPv6 addresses?
- Do the USA need IPv6 too?
- Will you put this slides on the web?