IPv6 Tutorial

János Mohácsi

6DEPLOY Partners

NRENs

Renater	France
GRNET	Greece
FCCN	Portugal
NIIF/HUNGARNET	Hungary
UNINETT	Norway
BREN	Bulgaria

SMEs

Consulintel	Spain
Martel Consulting (coordinator)	Switzerland

Industry

Cisco	Netherlands

Universities

UCL	United Kingdom
Soton ECS	United Kingdom

Non-European Partners

AfriNIC	Mauritius
LACNIC	Uruguay

Associated partners: RIPE NCC, APNIC

Project Objectives

Support of EU policy

The Internet is now the main telecommunications technology that underpins all aspects of business and leisure, and as such is central to the economic growth of a country. Awareness of the evolution of the Internet, and providing support for the introduction of IPv6 is therefore crucial as ICT becomes a major theme in FP7

Specific ***technical focus*** on supporting the deployment of IPv6 in:

- research infrastructures, for supporting all fields of science and technology
- FP7 projects (especially in the areas of emergency services, healthcare, transport, gaming)
- developing countries (Africa, Latin America, Asia and E. Europe), and
- commercial organisations in Europe

History

FP4: 6INIT, 6WINIT

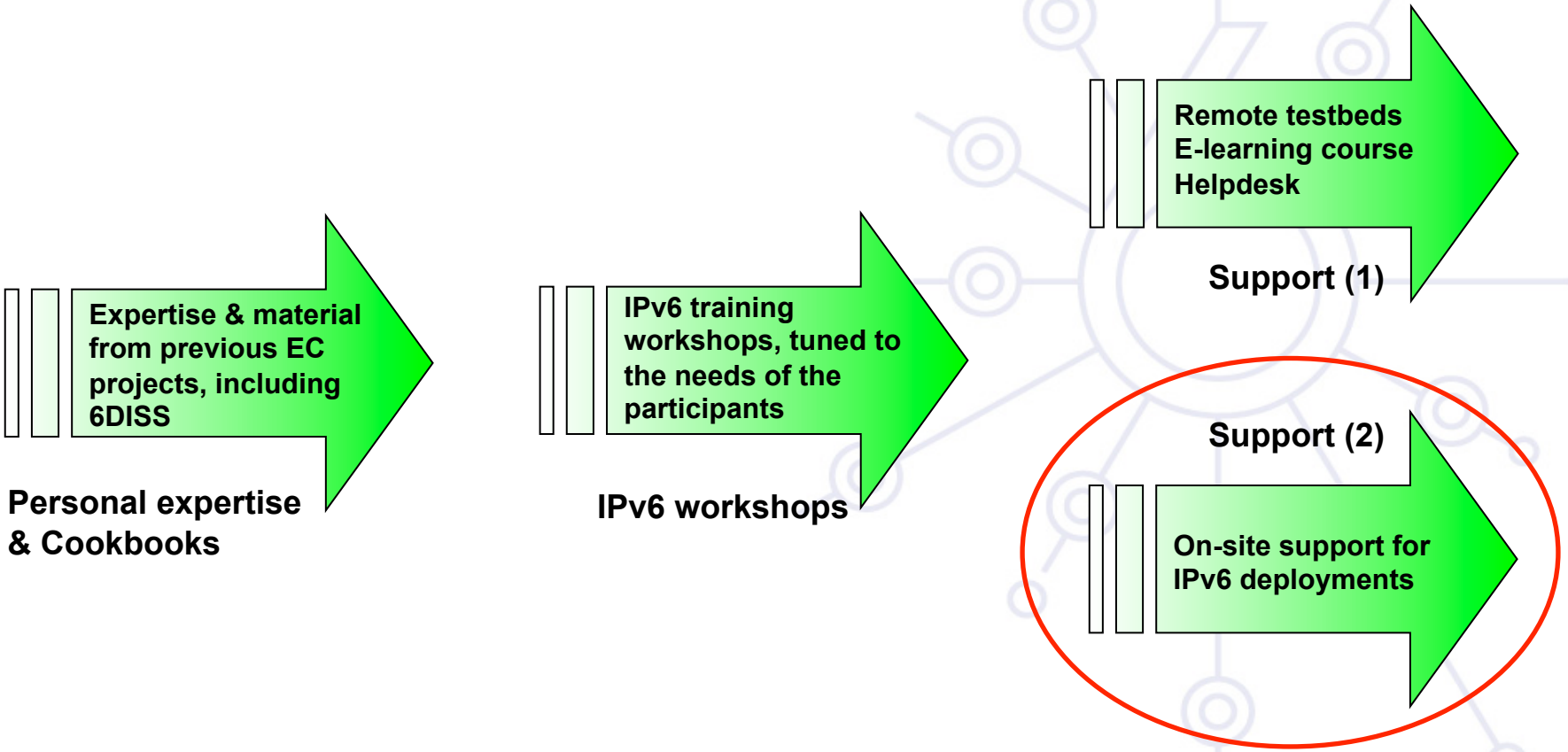
FP5: 6NET, Euro6IX, Occasion, 6Power, IPv6 TF

FP6: 6DISS, Sponge, IPv6 TF (continued)

FP7: 6DEPLOY

6DEPLOY is the one we exploit the most, in terms of partners and material

Technical Approach



The 6DEPLOY Toolkit

- Workshops for direct training, and for „training other trainers“
- Presentation material on 20 topics associated with IPv6
- Practical configuration exercises
- Professional e-learning package
- Remote testbeds in Paris and Sofia (for use in- and out- side the workshops)
- Book on deployment guidelines (from 6NET)
- Helpdesk service run by experienced persons
- Website with links to 6DEPLOY documents and external sources

http://www.6deploy.org/index.php?page=e-learning
www.6deploy.org

Copy ...Rights

This slide set is the ownership of the 6DEPLOY project via its partners

The Powerpoint version of this material may be reused and modified only with written authorization

Using part of this material must mention 6DEPLOY courtesy

PDF files are available from www.6deploy.org

Contributions

Main authors

- János Mohácsi – NIIF Insitute, Hungary
- Jérôme Durand, Simon Muyal, Bernard Tuy - Renater, France
- Tim Chown - University of Southampton, Great-Britain
- Stig Venaas – UNINETT, Norway
- Joao Nuno Ferreira, Carlos Friacas - FCCN, Portugal

Why a new version for IP ?

Agenda

Historical facts

IPv4 address space status

From Emergency measures ...

... to IPv6

Historical facts

1983 : Research network for ~ 100 computers

1992 : Internet is open to the commercial sector :

- Exponential growth
- IETF urged to work on a IP next generation protocol

1993 : - Exhaustion of the class B address space

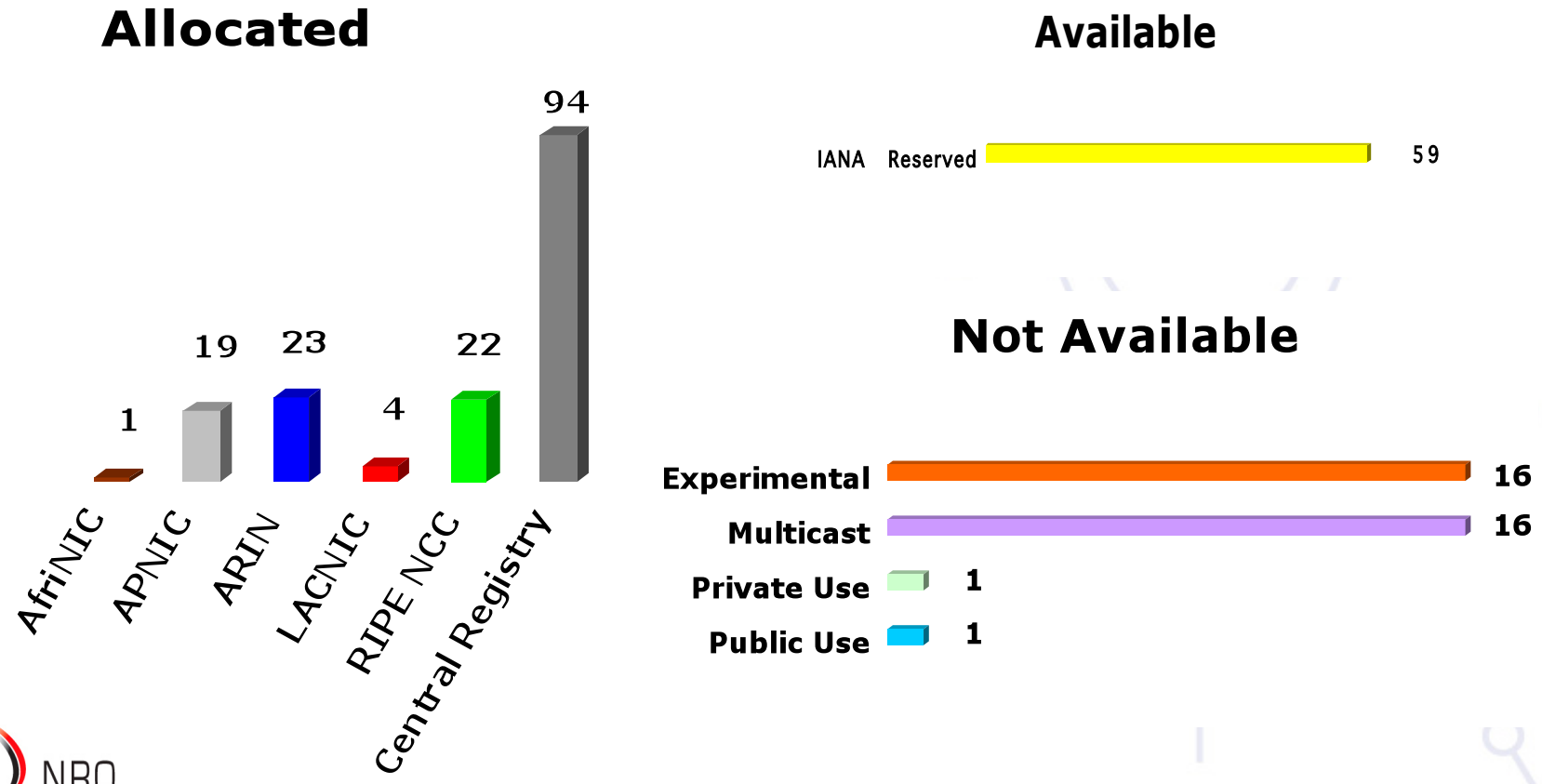
Forecast of network collapse for 1994 !

- RFC 1519 (CIDR) published

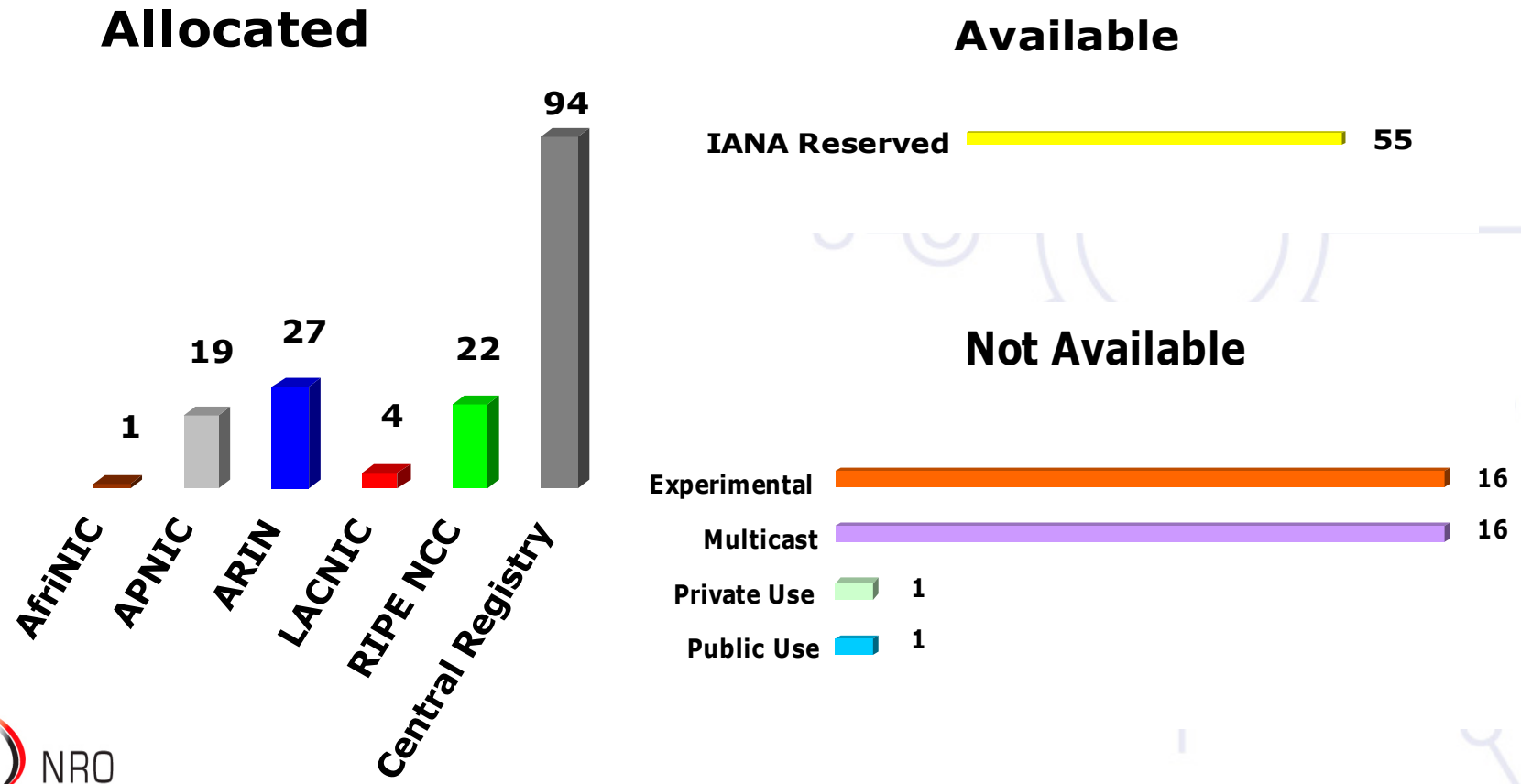
1995 : RFC 1883 (IPv6 specs) published

- First RFC about IPv6

IPv4 Address Space Status (sep. 2006)



IPv4 Address Space Status (dec.2006)



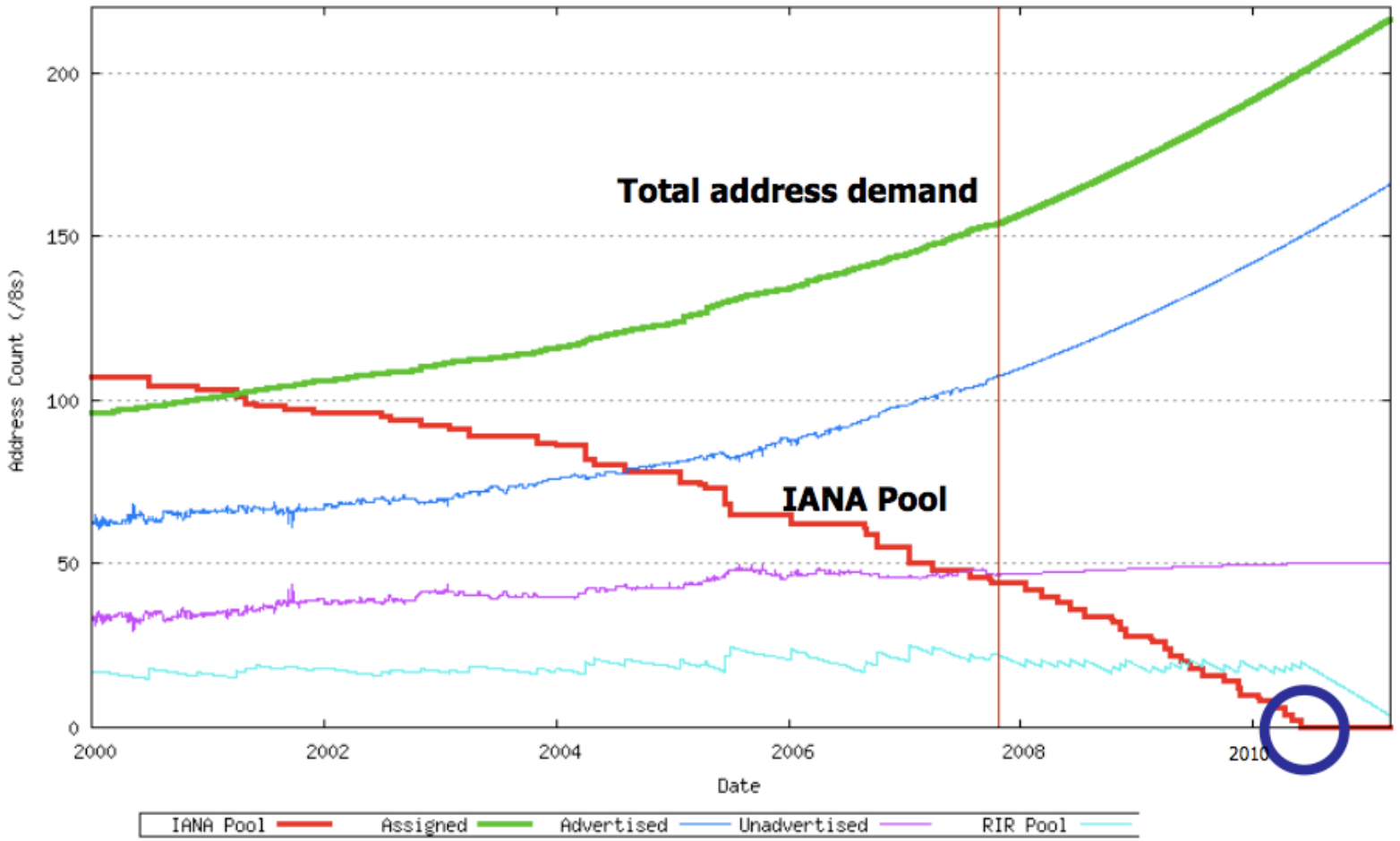
IPv4 prefixes consumption pace

200

200

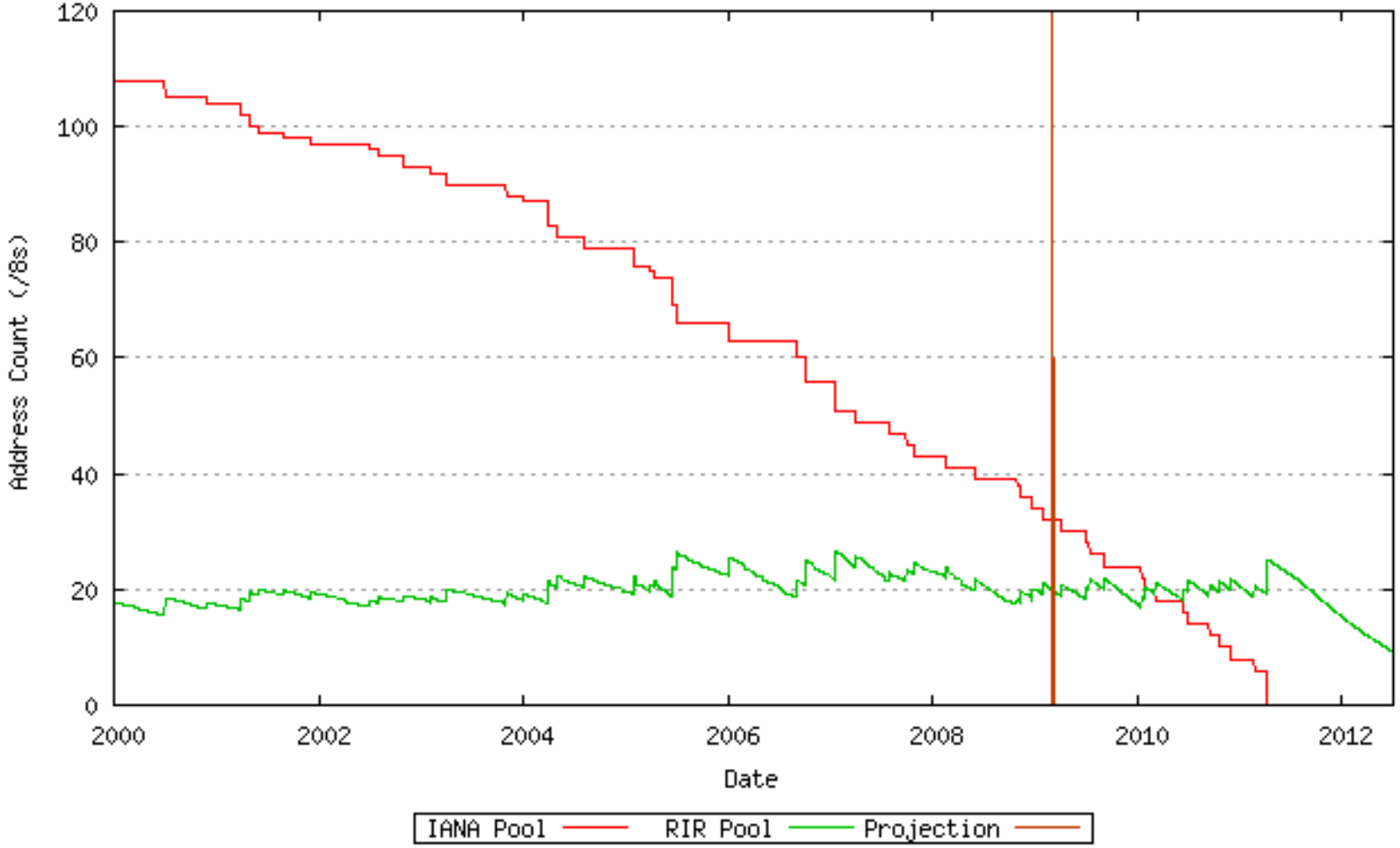
Year	Month	available /8s (IANA)	Yearly consumption
2006	September	59	
	December	55	16
2007	September	44	15
	December	42	14
2008	June	39 (-2) ?	11 (13) ?
	December	34	11
2009	March	32	11

IPv4 address space depletion



Geoff Huston
APNIC
Sept. 2007

IPv4 address space depletion



Emergency measures ...

Summary

CIDR

Private addresses

NAT

CIDR ...

Allocate former "class B" addresses exceptionally

- known as /16 prefixes since then

Re-use "class C" address space

- Without any more address classes

CIDR (*Classless Internet Domain Routing*)

- RFC 1519 (PS)
- network address = {prefix/prefix length}
- Classes abandon = less address waste
- allows aggregation => reduces routing table size

Private addresses
(RFC 1918)

Allow private addressing plans

Addresses are used internally

Similar to security architecture with firewall

Use of proxies or NAT to go outside

- RFC 1631, 2663 and 2993

NAT-PT

- the most commonly used of NAT variations in the IPv6 world

NAT (continued)

Advantages:

- Reduce the need of official addresses
- Ease the internal addressing plan
- Transparent to some applications
- "Security" vs obscurity
- Netadmins/sysadmin

Disadvantages:

- Translation sometime complex (e.g. FTP)
- Apps using dynamic ports
- Does not scale
- Introduce states inside the network:
  - Multihomed networks
- Breaks the end-to-end paradigm
- Security with IPsec

=> Should be reserved for small sites in Client/Server mode

Emergency Measures

- These emergency measures gave time to develop a new version of IP, named IPv6
- IPv6 keeps principles that have made the success of IP
- Corrects what was wrong with the current version (v4)
- BUT are emergency measures enough?

From emergency to IPv6

IPv6 is already there …

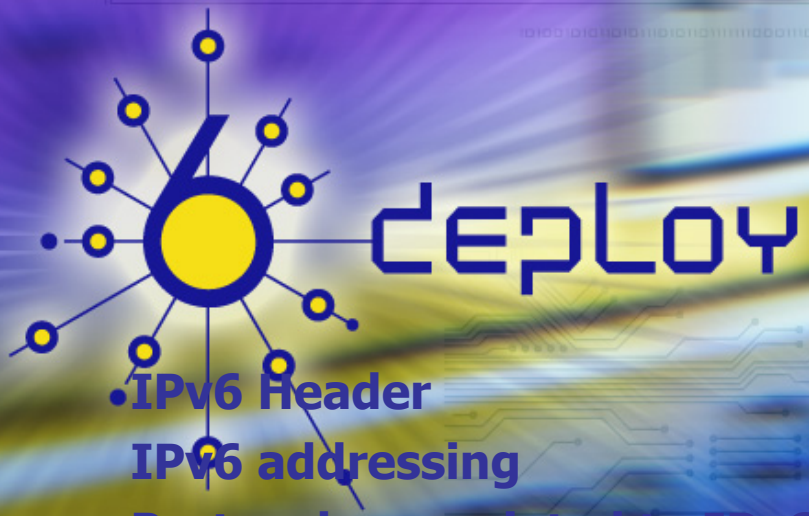
- Internet v6 is there today :
- NRENs in EU, North America, Asia … are interconnected in IPv6
- Lots of IXP are offering IPv6 connectivity
- ISPs and Telcos exchange IPv6 routes
- Vista and Windows 2008 (servers) are IPv6 enabled by default

Then the question is not "if" but "when ?" and "how ?"

By Apr. 10th 2009 resources exhaustion are projected

- IANA pool : Jun. 2011
- RIRs pool : Oct. 2012
- Data from : http://www.potaroo.net/tools/ipv4/index.html

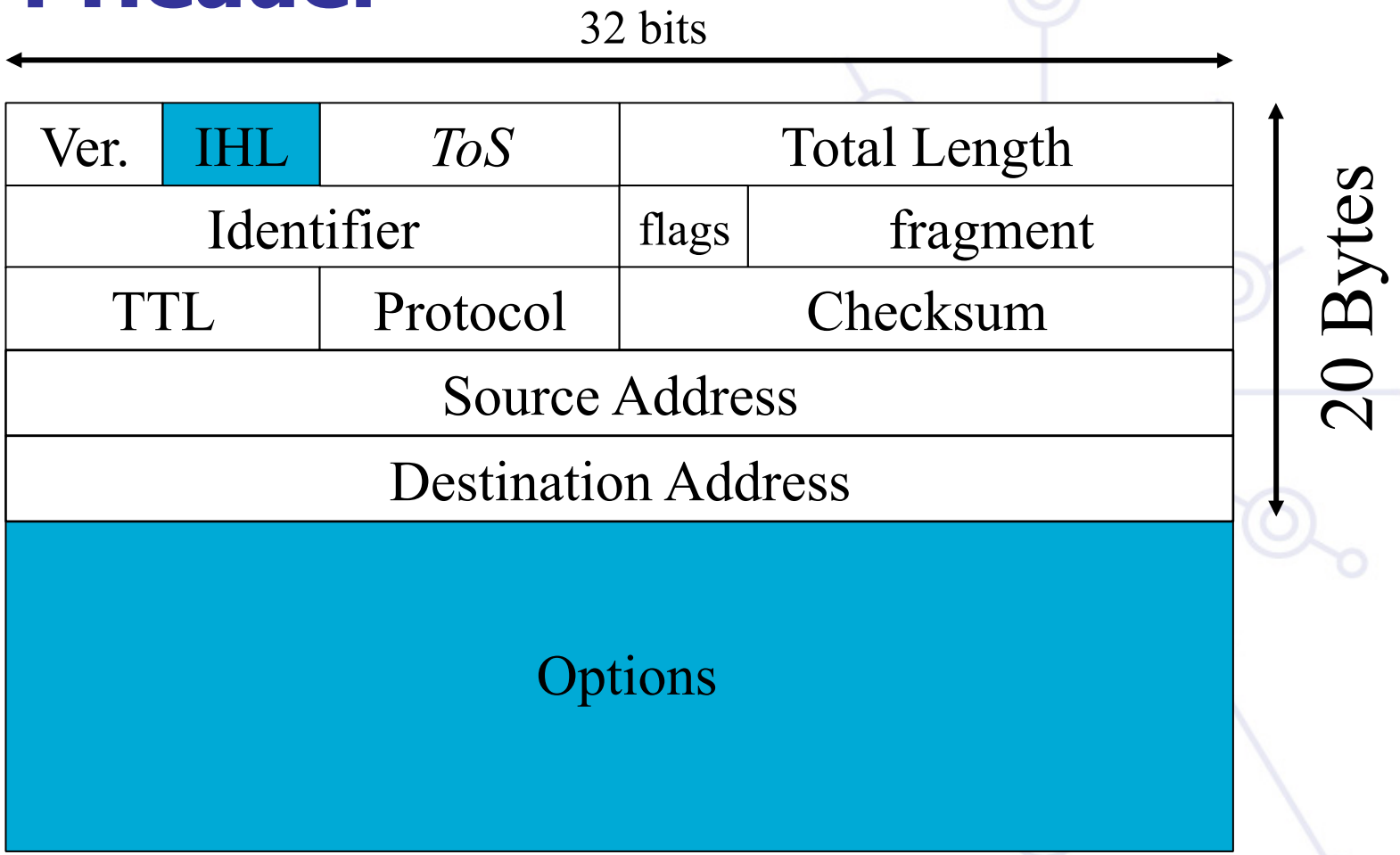
IPv6 Protocol (RFC 2460 DS)



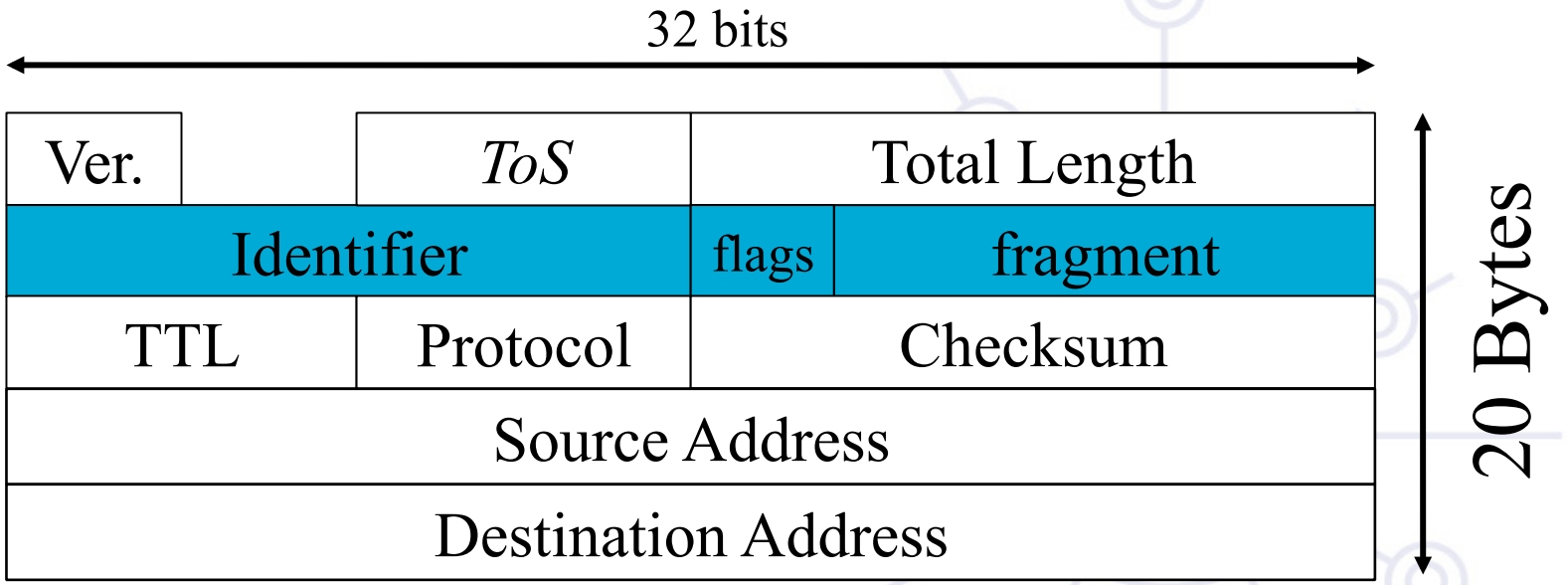
- IPv6 Header
- IPv6 addressing
- Protocols associated to IPv6

IPv6 Header

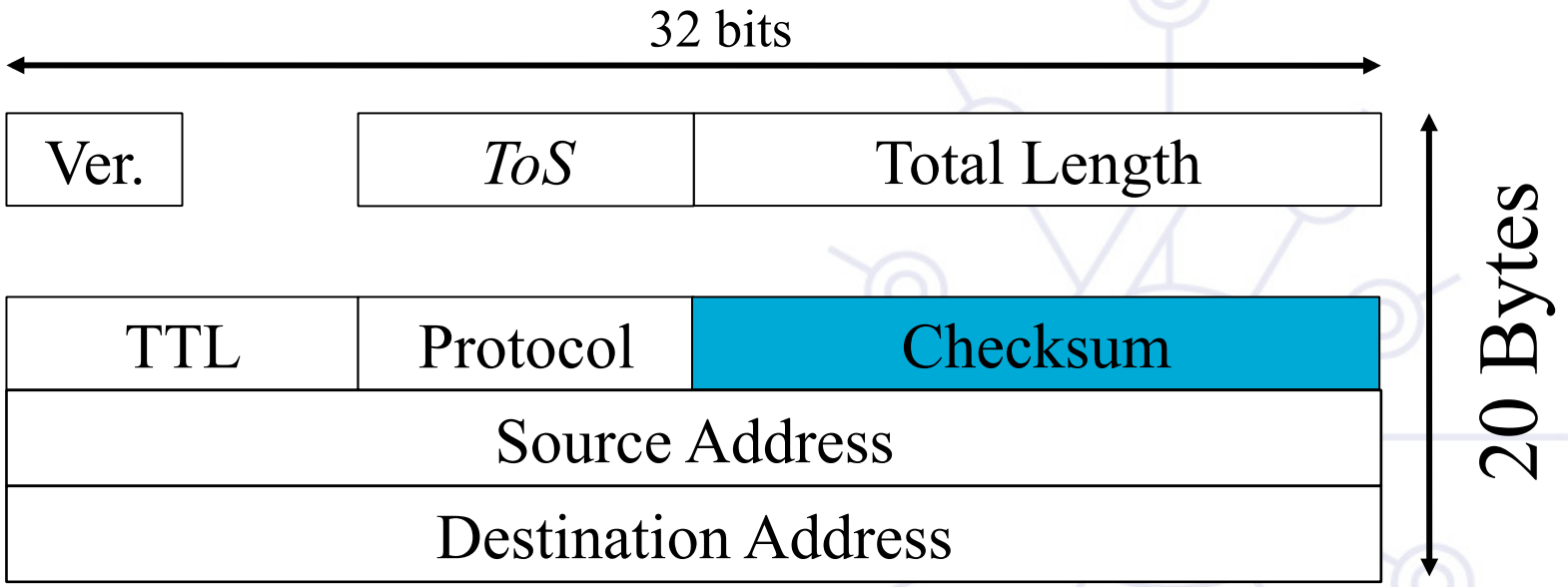
IPv4 Header


32 bits
Ver.
IHL
ToS
Total Length
Identifier
flags
fragment
TTL
Protocol
Checksum
Source Address
Destination Address
Options
20 Bytes

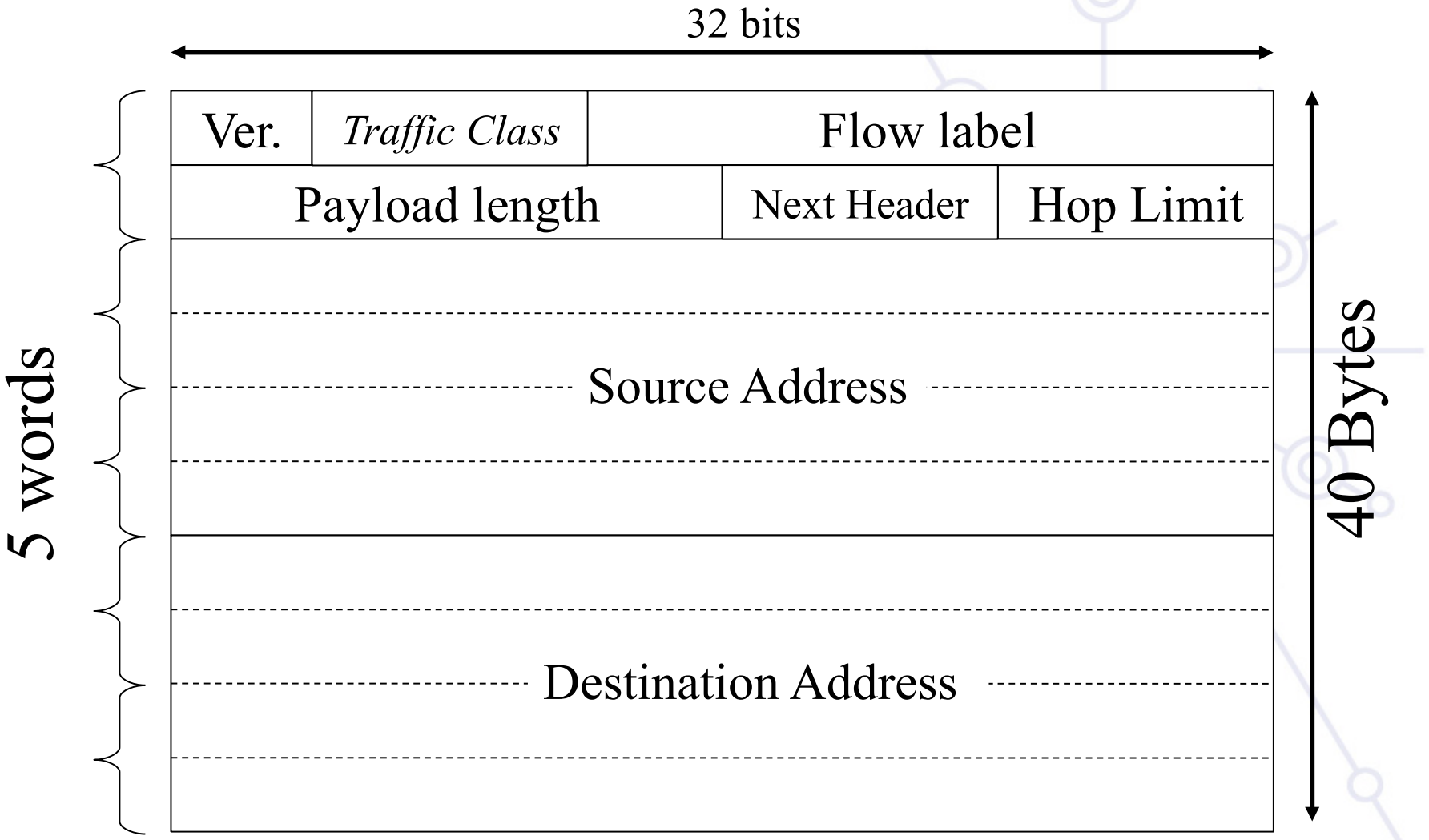
IPv4 Header


32 bits
Ver.
ToS
Total Length
Identifier
flags
fragment
TTL
Protocol
Checksum
Source Address
Destination Address
20 Bytes

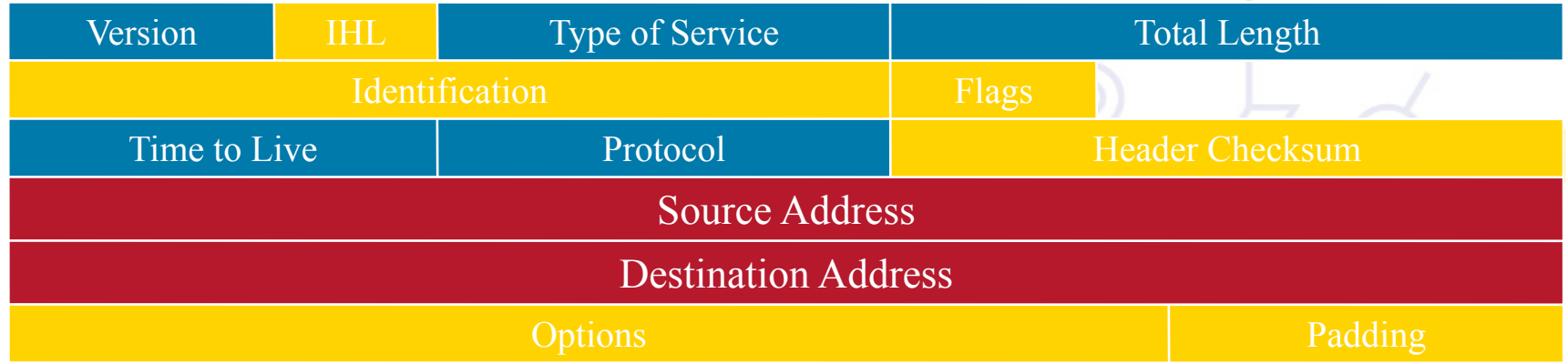
IPv4 Header

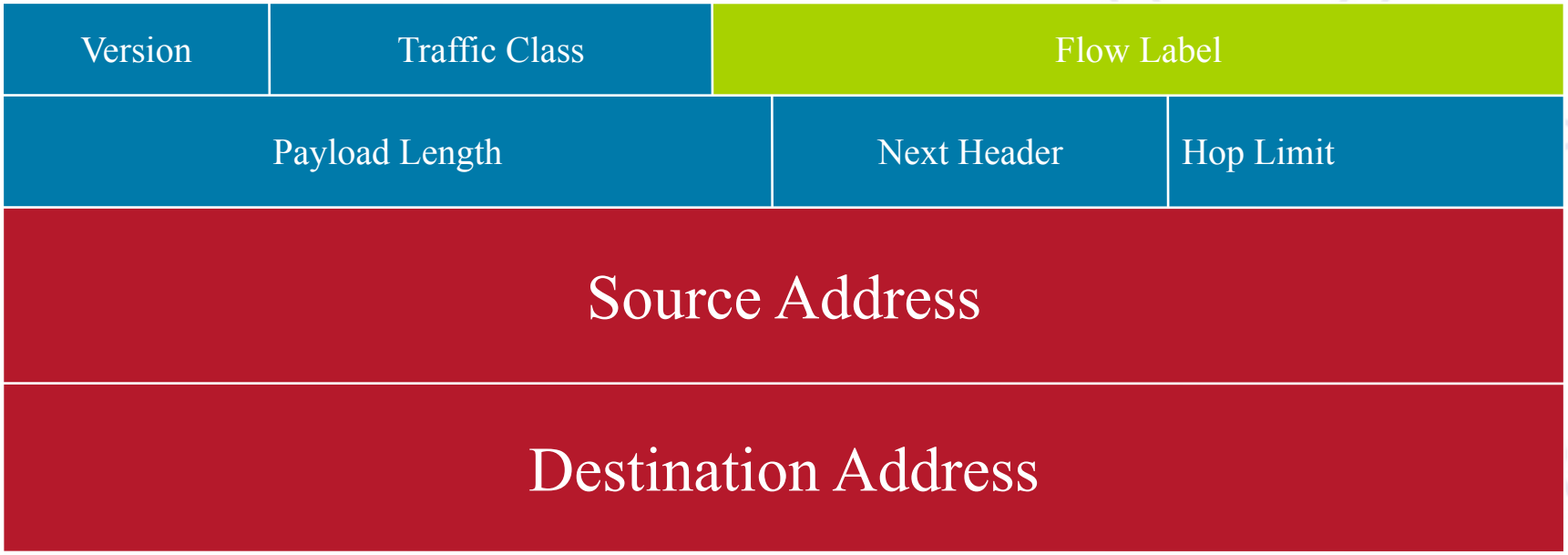

32 bits
Ver.
ToS
Total Length
TTL
Protocol
Checksum
Source Address
Destination Address
20 Bytes

IPv6: Header simplification



IPv4 & IPv6 Fejléc összehasonlítás





Is it enough for the future ?

Address length

- Between 1 564 and 3 911 873 538 269 506 102 addresses by m^2
- Justification of a fix address length

Hop Limit

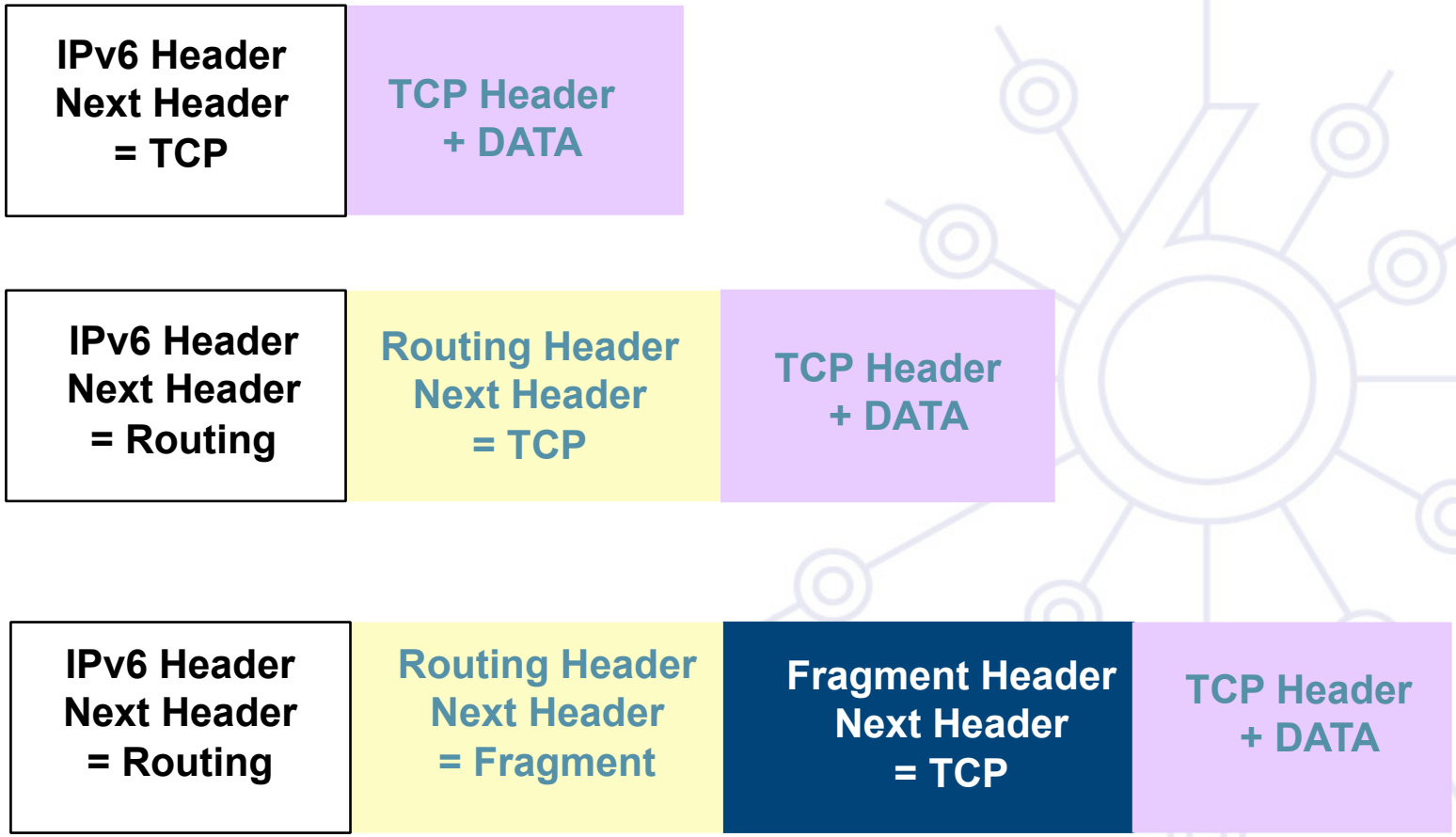
- Should not be a problem

Payload Length

- Use Jumbogram for specific cases

IPv6 extensions

IPv6: Optional headers



IPv6: Optional extensions

Hop-by-hop (jumbogram, router alert)

- Always the first extension
- Replace IPv4 options,
- Analyzed by every router.

Destination

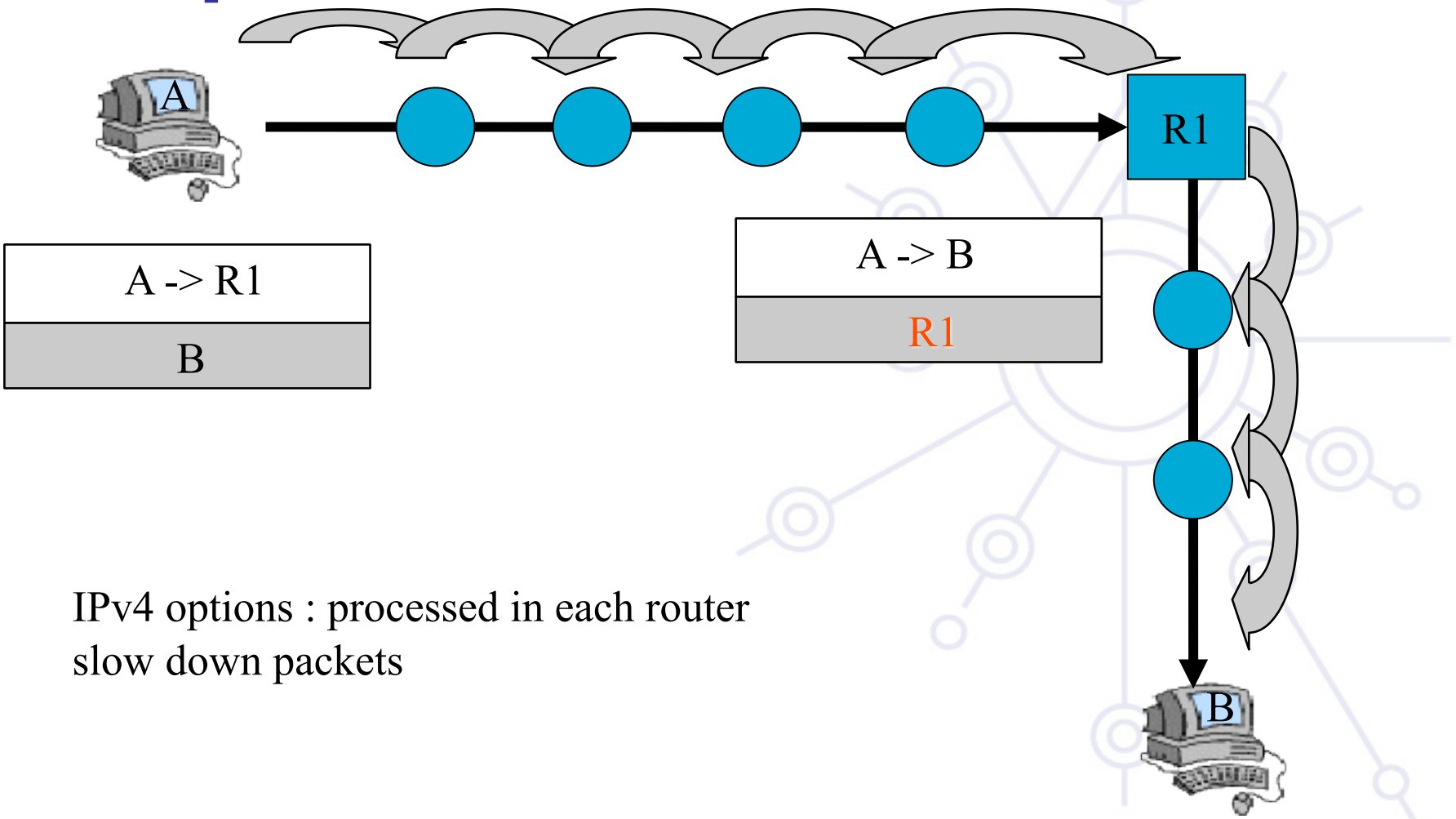
Routing (loose source routing)

Fragmentation

Authentication

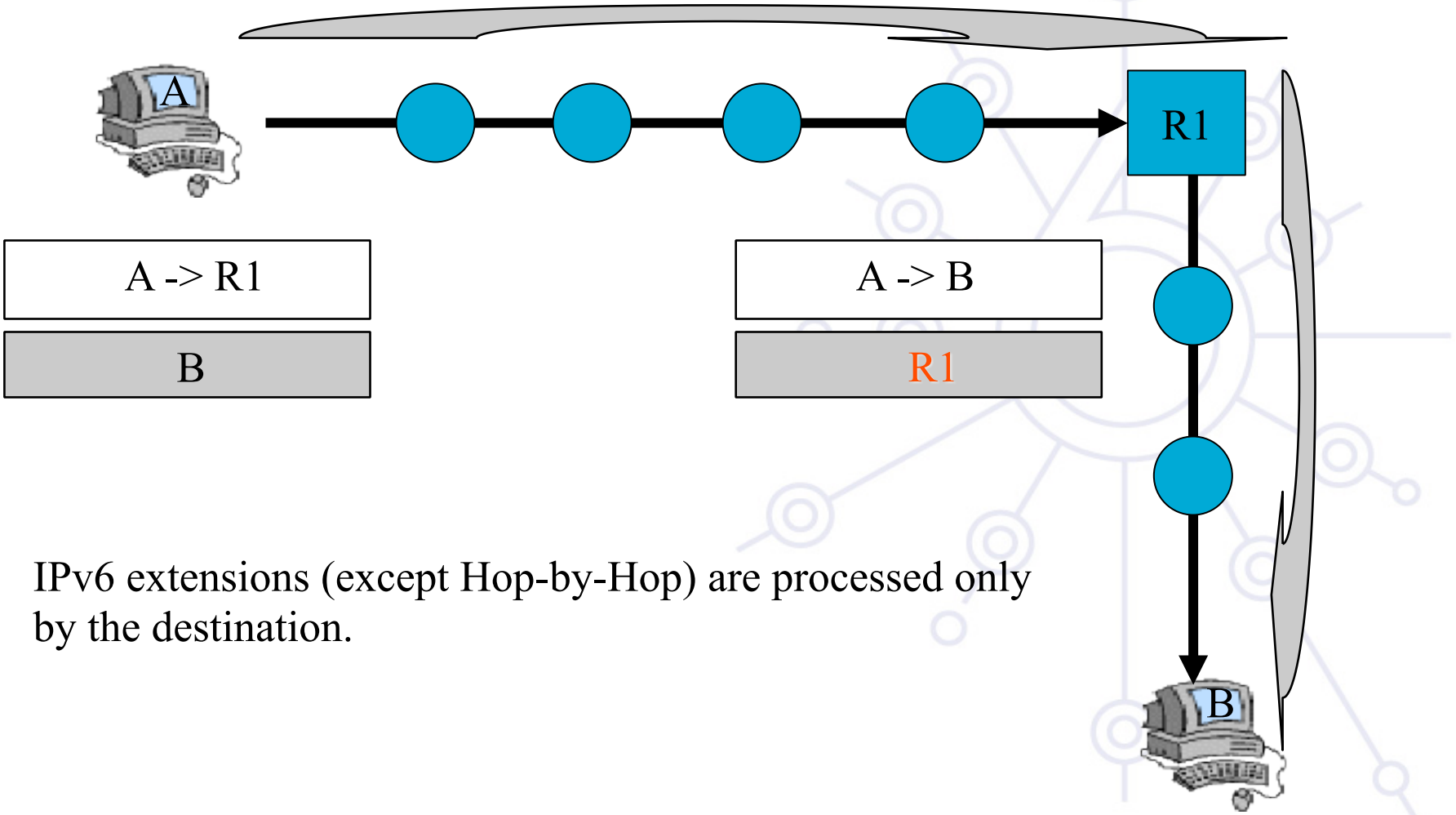
Security

v4 options vs. v6 extensions



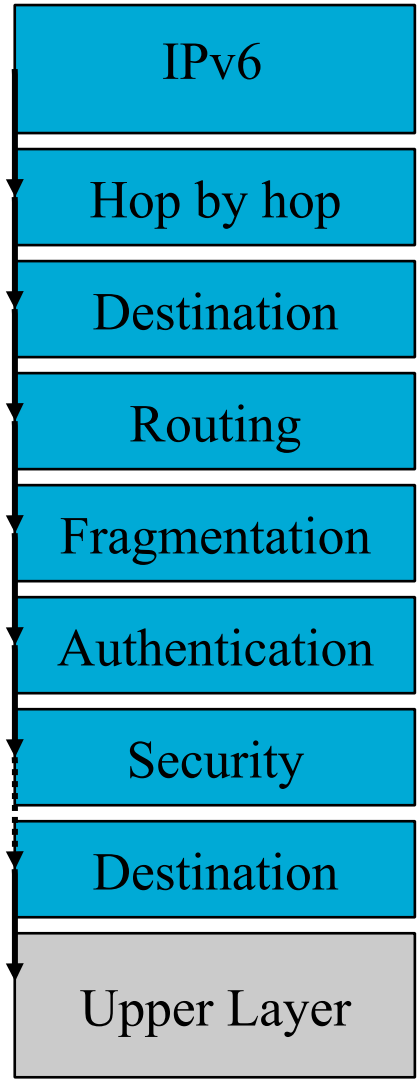
IPv4 options : processed in each router
slow down packets

v4 options vs. v6 extensions



IPv6 extensions (except Hop-by-Hop) are processed only by the destination.

Order is important



- Hop by hop — Processed by every router
- Destination — Processed by routers listed in Routing extension
- Routing — List of routers to cross
- Fragmentation — Processed by the destination
- Authentication — After reassembling the packet
- Security — Cipher the content of the remaining information
- Destination — Processed only by the destination

IPv6 Addressing

6DEPLOY. IPv6 Deployment and Support

Alcatel-Lucent Szeminárium 2009 - IPv6 tutorial

IPv6 Addressing Scheme

RFC4291 defines IPv6 addressing scheme

RFC3587 defines IPv6 global unicast address format

128 bit long addresses

- Allow hierarchy
- Flexibility for network evolutions

Use CIDR principles:

- Prefix / prefix length
  - 2001:660:3003::/48
  - 2001:660:3003:2:a00:20ff:fe18:964c/64
- Aggregation reduces routing table size

Hexadecimal representation

Interfaces have several IPv6 addresses

IPv6 Address Types

Unicast (one-to-one)

- global
- link-local
- site-local (deprecated)
- Unique Local (ULA)
- IPv4-compatible (deprecated)
- IPv6-mapped

Multicast (one-to-many)

Anycast (one-to-nearest)

Reserved

Textual Address Format

Preferred Form (a 16-byte Global IPv6 Address):

2001:0DB8:3003:0001:0000:0000:6543:210F

Compact Format:

2001:DB8:3003:1::6543:210F

IPv4-mapped: ::FFFF:134.1.68.3

Literal representation

- [2001:DB8:3003:2:a00:20ff:fe18:964c]
- http://[2001:DB8::43]:80/index.html

IPv6 Address Type Prefixes

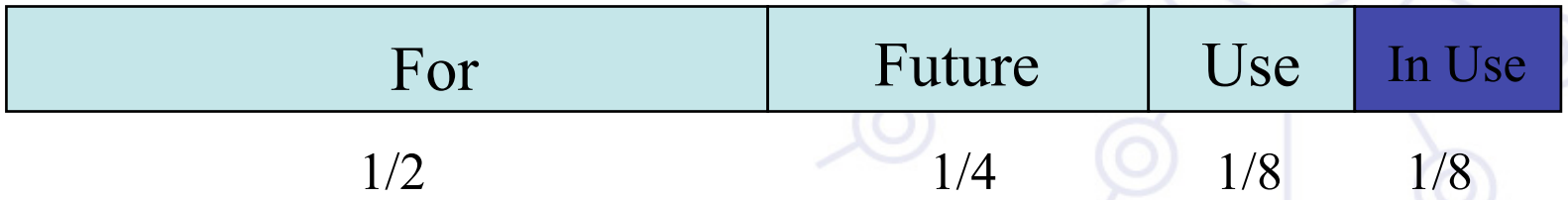
Address Type	Binary Prefix	IPv6 Notation
Unspecified	00…0 (128 bits)	::/128
Loopback	00…1 (128 bits)	::1/128
Multicast	1111 1111	FF00::/8
Link-Local Unicast	1111 1110 10	FE80::/10
ULA	1111 110	FC00::/7
Global Unicast	(everything else)	
IPv4-mapped	00…0:1111 1111:IPv4	::FFFF:IPv4/128
Site-Local Unicast (deprecated)	1111 1110 11	FEC0::/10
IPv4-compatible (deprecated)	00…0 (96 bits)	::IPv4/128

Global Unicast assigments actually use 2000::/3 (001 prefix)

Anycast addresses allocated from unicast prefixes

IPv6 Address Space

Aggregatable Global Unicast Addresses (001): 1/8
Unique Local Unicast addresses (1111 1110 00): 1/128
Link-Local Unicast Addresses (1111 1110 10): 1/1024
Multicast Addresses (1111 1111): 1/256



More info:
http://www.iana.org/assignments/ipv6-address-space

Some Special-Purpose Unicast Addresses

Listed in RFC5156

The **unspecified address,** used as a placeholder when no address is available:

0:0:0:0:0:0:0:0 (::/128)

The **loopback address,** for sending packets to itself:

0:0:0:0:0:0:0:1 (::1/128)

The **documentation prefix [RFC3849]:**
2001:db8::/32

Link-Local & Site-Local Unicast Addresses

Link-local addresses for use during auto-configuration and when no routers are present **(FE80::/10):**

10 bits	*54 bits*	*64 bits*
1111111010	0 0	Interface ID
FE80		

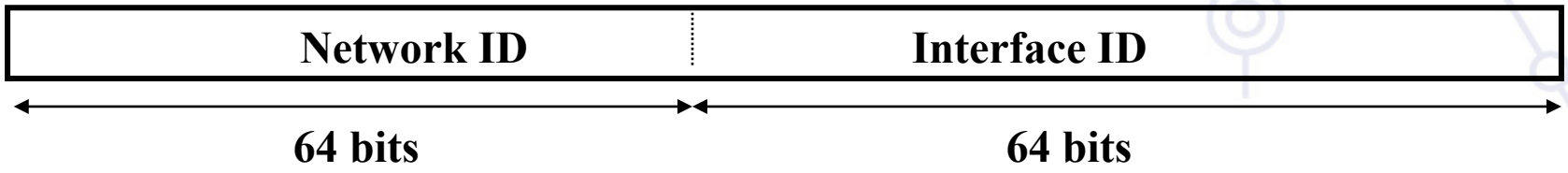
Site-local addresses for independence from changes of TLA / NLA* **(FEC0::/10):** (deprecated by RFC3879)

10 bits	*54 bits*	*64 bits*
1111111011	Subnet ID	Interface ID

Interface IDs

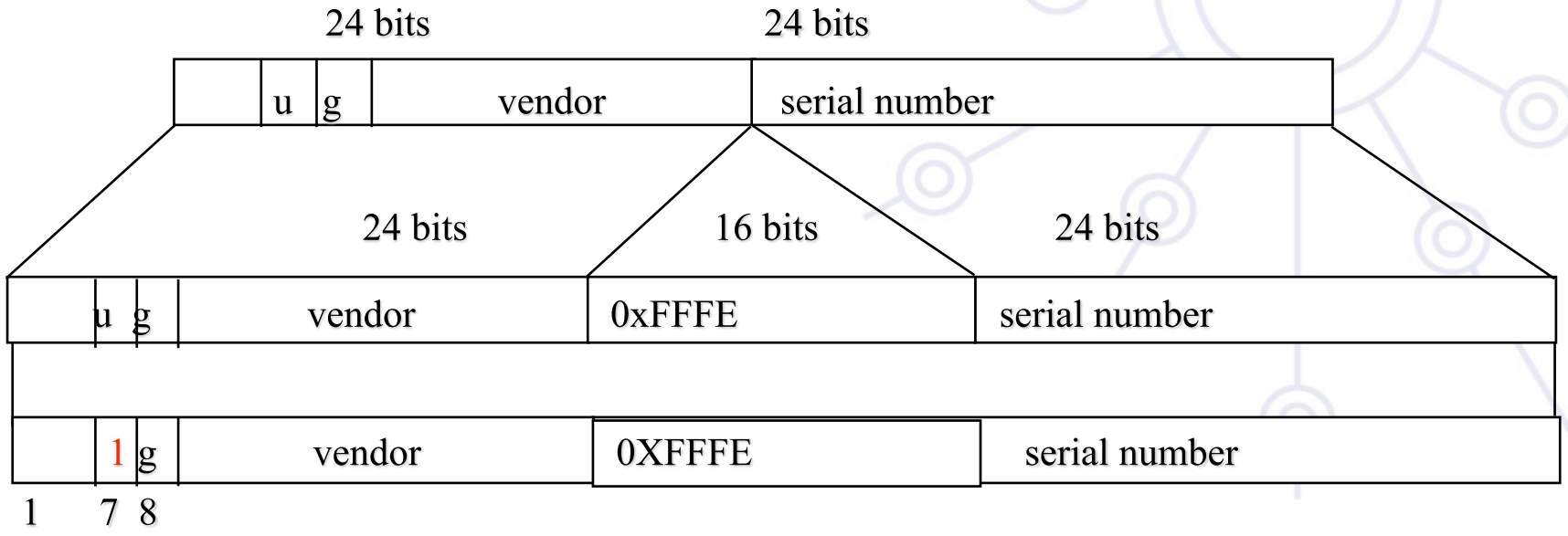
The lowest-order 64-bit field of unicast addresses may be assigned in several different ways:

- auto-configured from a 64-bit MAC address
- auto-configured from a 48-bit MAC address (e.g., Ethernet) expanded into a 64-bit EUI-64 format
- assigned via DHCP
- manually configured
- auto-generated pseudo-random number (to counter some privacy concerns)
- CGA (Cryptographically Generated Address)
- possibly other methods in the future



Autoconfigured Interface IDs (1)

- 64 bits to be compatible with IEEE 1394 (FireWire)
- Eases auto-configuration
- IEEE defines the mechanism to create an EUI-64 from IEEE 802 MAC addresses (Ethernet, FDDI)



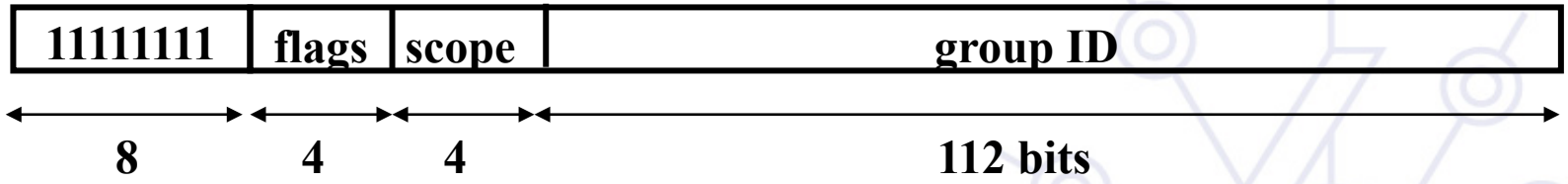
Autoconfigured Interface IDs (2)

Links with non global identifier (e.g., the Localtalk 8 bit node identifier) → fill first left bits with 0

For links without identifiers, there are different ways to proceed (e.g., tunnels, PPP) to have a subnet-prefix-unique identifier:

- Choose the universal identifier of another interface
- Manual configuration
- Node Serial Number
- Other Node-Specific Token

Multicast Addresses



Flags: 0RPT: The high-order flag is reserved, and must be initialized to 0.

- **T**: Transient, or not, assignment
- **P**: Assigned, or not, based on network prefix
- **R**: Rendezvous Point Address embedded, or not

Scope field:
1 - Interface-Local
2 - link-local
4 - admin-local
5 - site-local
8 - organization-local
E - global

(3,F reserved)(6,7,9,A,B,C,D unassigned)

Unique Local IPv6 Unicast Addresses (1)

ULAs are defined in **RFC4193:**

- Globally unique prefix with high probability of uniqueness
- Intended for local communications, usually inside a site
- They are not expected to be routable on the Global Internet
- They are routable inside of a more limited area such as a site
- They may also be routed between a limited set of sites
- Locally-Assigned Local addresses vs. Centrally-Assigned Local addresses

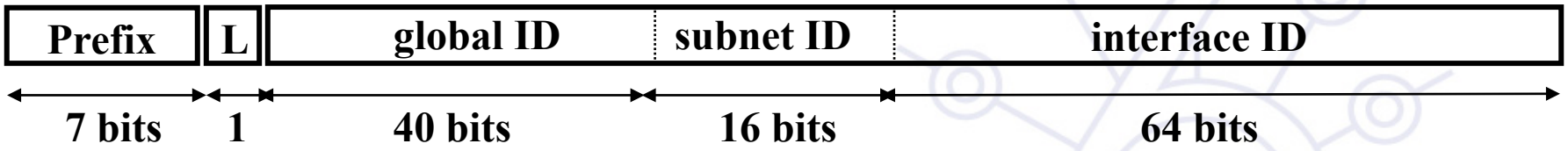
Unique Local IPv6 Unicast Addresses (2)

ULA characteristics:

- Well-known prefix to allow for easy filtering at site boundaries
- ISP independent and can be used for communications inside of a site without having any permanent or intermittent Internet connectivity
- If accidentally leaked outside of a site via routing or DNS, there is no conflict with any other addresses
- In practice, applications may treat these addresses like global scoped addresses

Unique Local IPv6 Unicast Addresses (3)

Format:



FC00::/7 Prefix identifies the Local IPv6 unicast addresses

L = 1 if the prefix is **locally assigned**

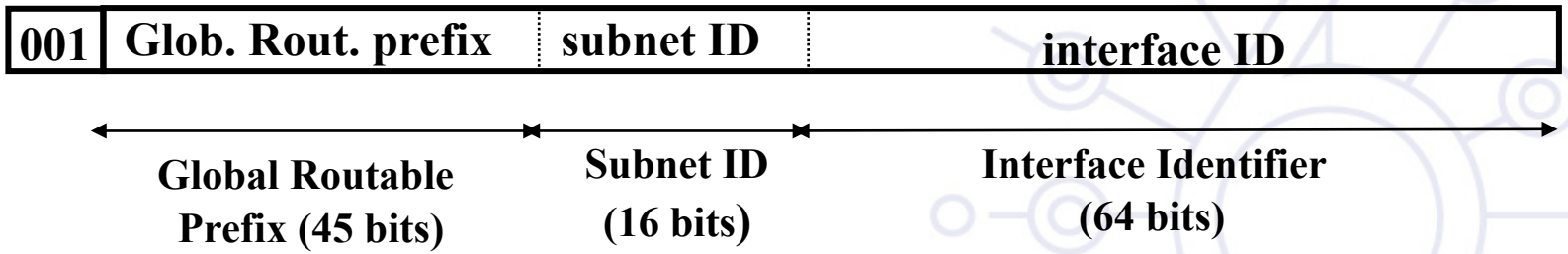
L = 0 may be defined in the future (in practice used for **centrally assigned** prefixes)

ULA are created using a pseudo-randomly allocated global ID

- This ensures that there is not any relationship between allocations and clarifies that these prefixes are not intended to be routed globally

Global Unicast Addresses

Defined in RFC3587



The global routing prefix is a value assigned to a zone (site, a set of subnetworks/links)

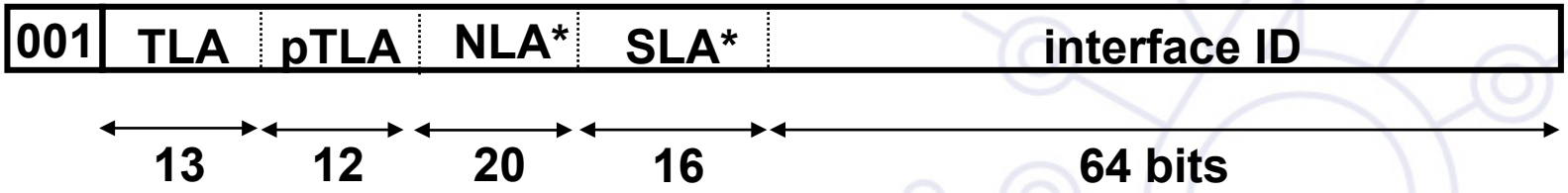
- It has been designed as an hierarchical structure from the Global Routing perspective

The subnetwork ID, identifies a subnetwork within a site

- Has been designed to be an hierarchical structure from the site administrator perspective

6Bone Global Unicast Addresses

Obsoleted 6/6/6 – RFC3701



6Bone: experimental IPv6 network used for testing only

TLA 1FFE (hex) assigned to the 6Bone

- 6Bone addresses start with 3FFE (binary 001 + 1 1111 1111 1110)

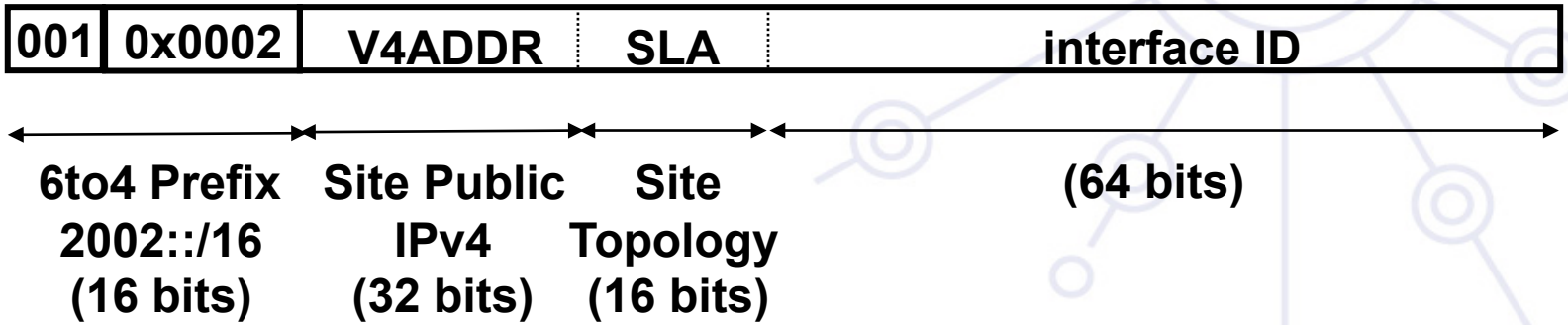
Not to be used for production IPv6 service

6to4 Addresses

Defined in **RFC3056: Connection of IPv6 Domains via IPv4 Clouds**

Assigned Prefix: 2002::/16

To assign to sites 2002:V4ADDR::/48



Miért fix hosszakat használunk?

A fix méret csökkenti a felhasználó problémáit, ha szolgáltatót kíván váltani.

A szabványos méret nem igényli azt, hogy a felhasználó indokolja az igényeit.

16 bites site méret elegendő a legtöbb felhasználónak a legnagyobbakat kivéve

Anycast Addresses

Identifier for a set of interfaces (typically in different nodes). A packet sent to an anycast address is delivered to the "nearest" interface (routing protocols' distance)

Taken from the unicast address space (of any scope).**Not syntactically distinguishable from unicast addresses**

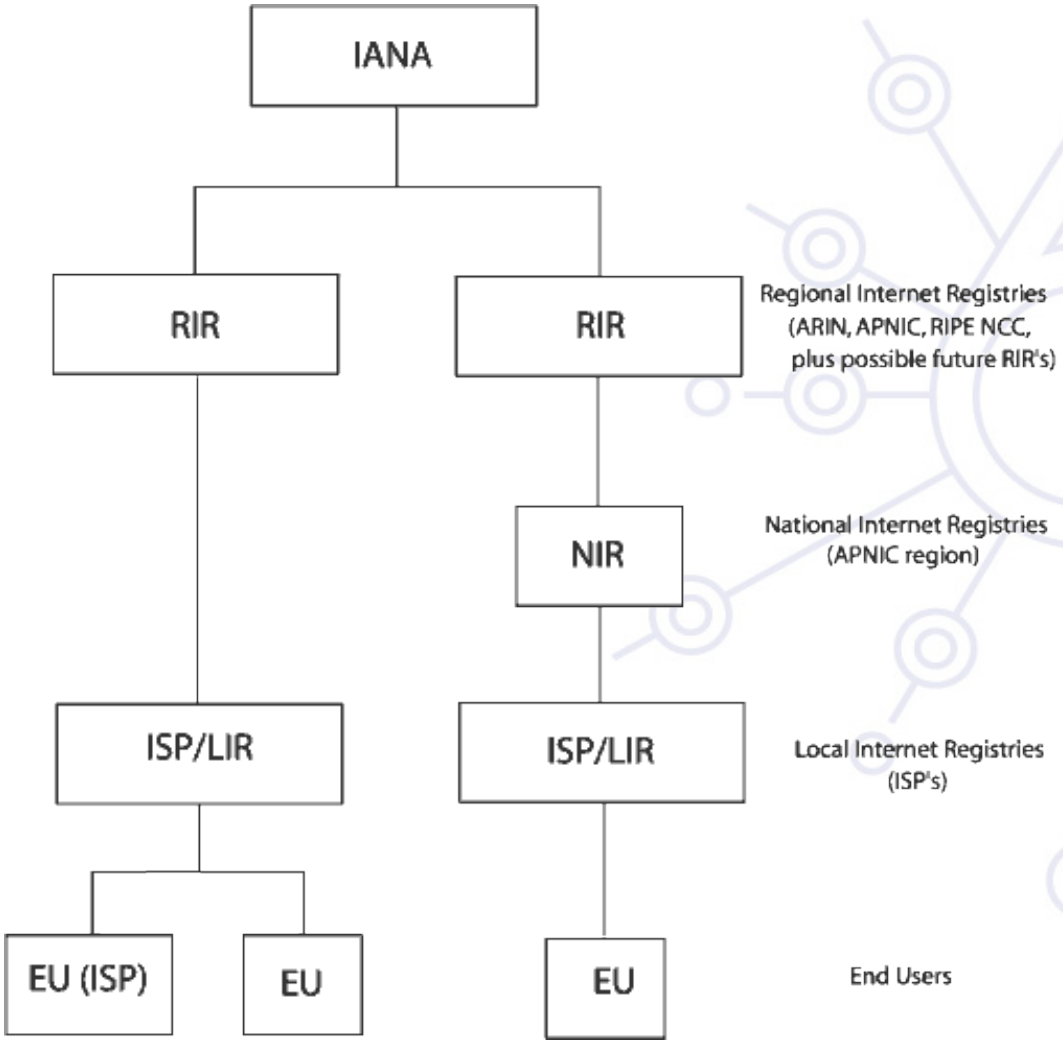
A unicast address assigned to more than one interface, turning it into an anycast address, the nodes the address is assigned must be explicitly configured to know that it is an anycast address

Anycast address cannot be source address of a packet

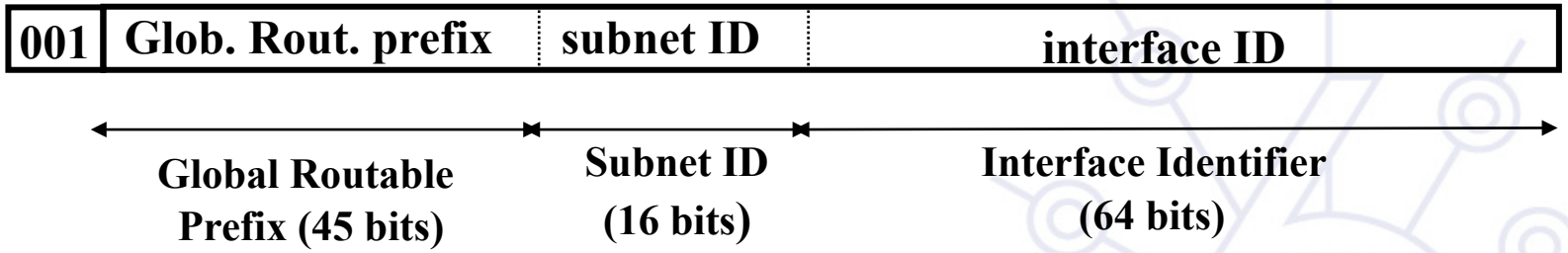
Reserved anycast addresses are defined in **RFC2526**

The Subnet-Router anycast address is predefined (mandatory on all routers):

Production Addressing Scheme (1)



Production Addressing Scheme (2)



LIRs receive by default /32

- Production addresses today are from prefixes 2001, 2003, 2400, etc.
- Can request for more if justified

/48 used only within the LIR network, with some exceptions for critical infrastructures

/48 to /128 is delegated to end users

- Recommendations following RFC3177 and current policies
- /48 general case, /47 if justified for bigger networks
- Small networks somewhere between /48-/60
- /64 if one and only one network is required
- /128 if it is sure that one and only one device is going to be connected

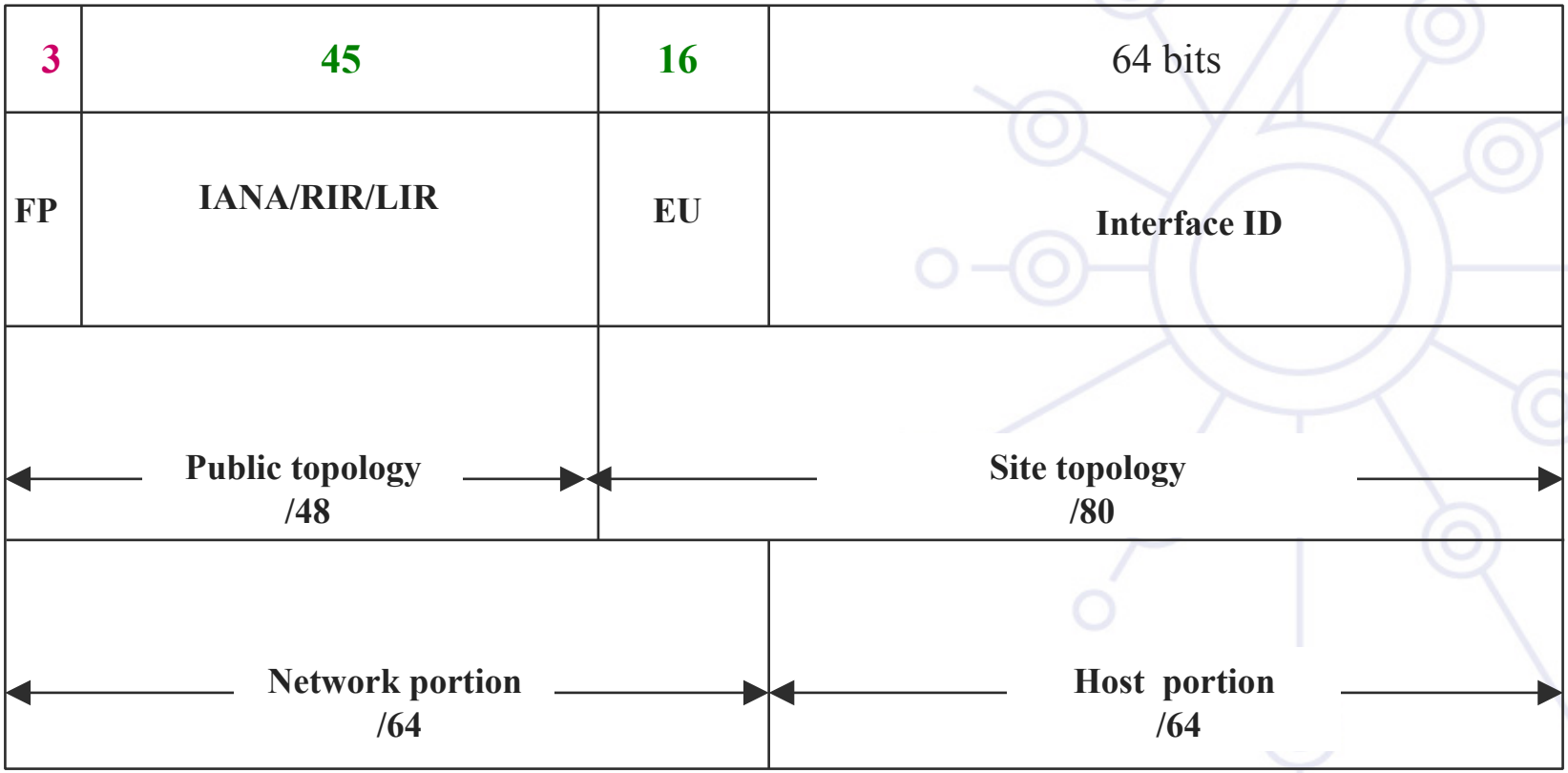
Production Addressing Scheme (3)

Source:
http://www.iana.org/assignments/ipv6-unicast-address-assignments

IPv6 Global Unicast Address Assignments [0]
[last updated 2008-05-13]

Global Unicast Prefix	Assignment	Date	Note
2001:0000::/23	IANA	01 Jul 99	[1]
2001:0200::/23	APNIC	01 Jul 99	
2001:0400::/23	ARIN	01 Jul 99	
2001:0600::/23	RIPE NCC	01 Jul 99	
2001:0800::/23	RIPE NCC	01 May 02	
2001:0A00::/23	RIPE NCC	02 Nov 02	
2001:0C00::/23	APNIC	01 May 02	[2]
2001:0E00::/23	APNIC	01 Jan 03	
2001:1200::/23	LACNIC	01 Nov 02	

Production Addressing Scheme (4)



RIR Allocation Policies

AfriNIC:
http://www.afrinic.net/IPv6/index.htm
http://www.afrinic.net/docs/policies/afpol-v6200407-000.htm *
APNIC:
http://www.apnic.org/docs/index.html
http://www.apnic.org/policy/ipv6-address-policy.html *
ARIN:
http://www.arin.net/policy/index.html
http://www.arin.net/policy/nrpm.html#ipv6 *
LACNIC:
http://lacnic.net/sp/politicas/
http://lacnic.net/sp/politicas/ipv6.html *
RIPE-NCC:
http://www.ripe.net/ripe/docs/ipv6.html
http://www.ripe.net/ripe/docs/ipv6policy.html *

- *describes policies for the allocation and assignment of globally unique IPv6 address space

RIR Allocation Statistics

AfriNIC:

- http://www.afrinic.net/statistics/index.htm

APNIC:

- http://www.apnic.org/info/reports/index.html

ARIN:

- http://www.arin.net/statistics/index.html

LACNIC:

- http://lacnic.org/sp/est.html

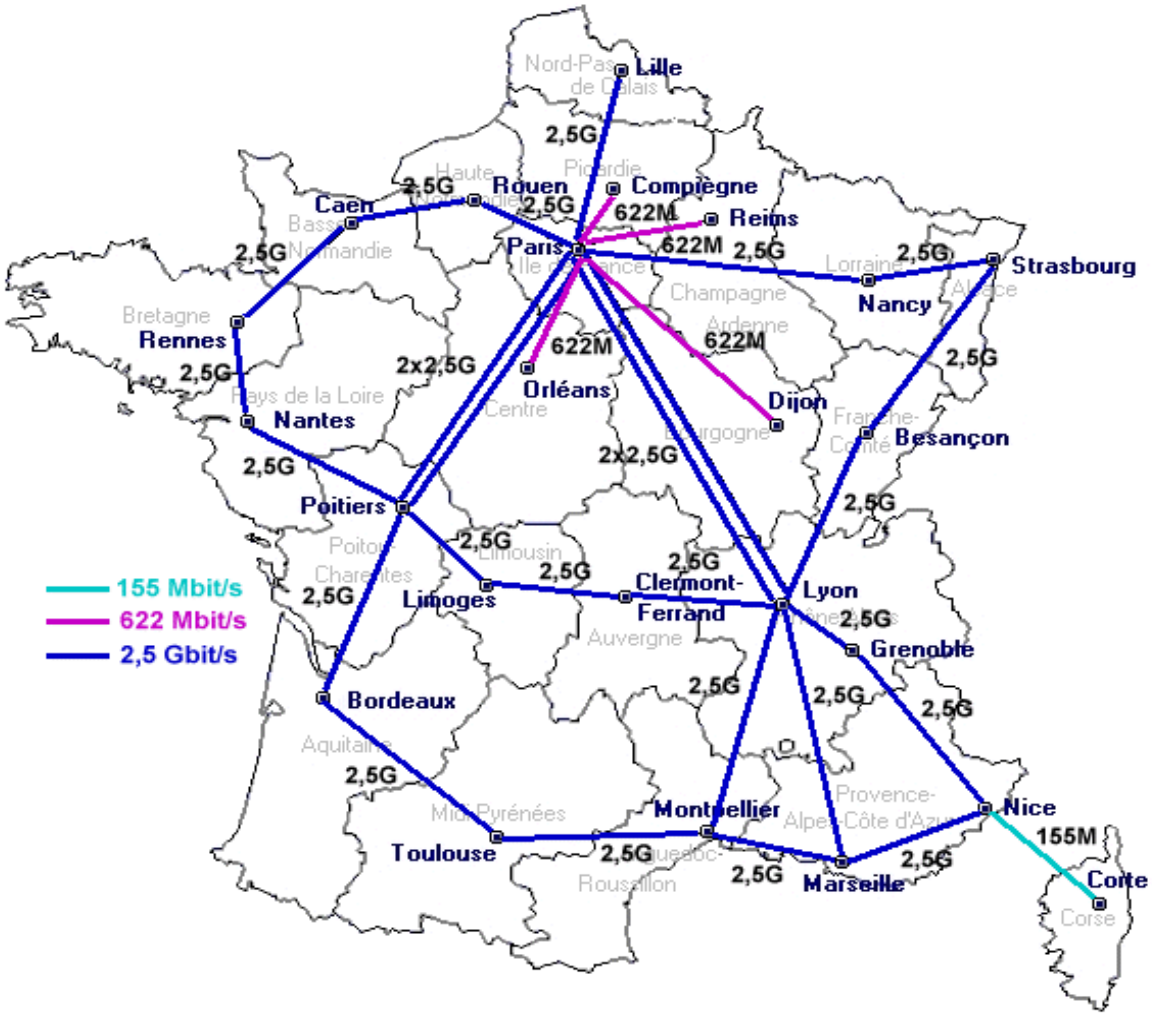
RIPE-NCC:

- http://www.ripe.net/info/stats/index.html

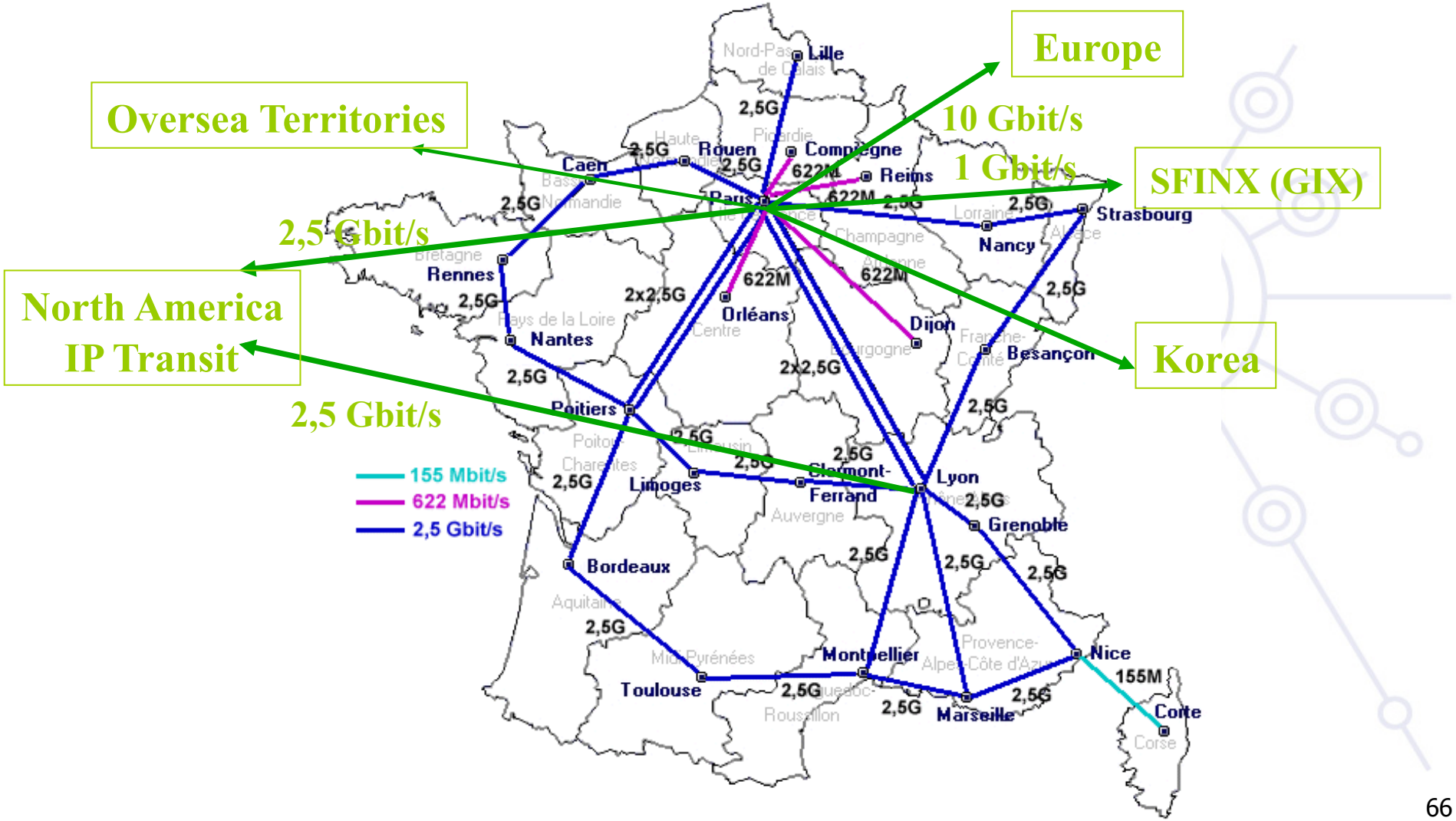
See http://www.ripe.net/rs/ipv6/stats/


deploy
RENATER IPv6 numbering

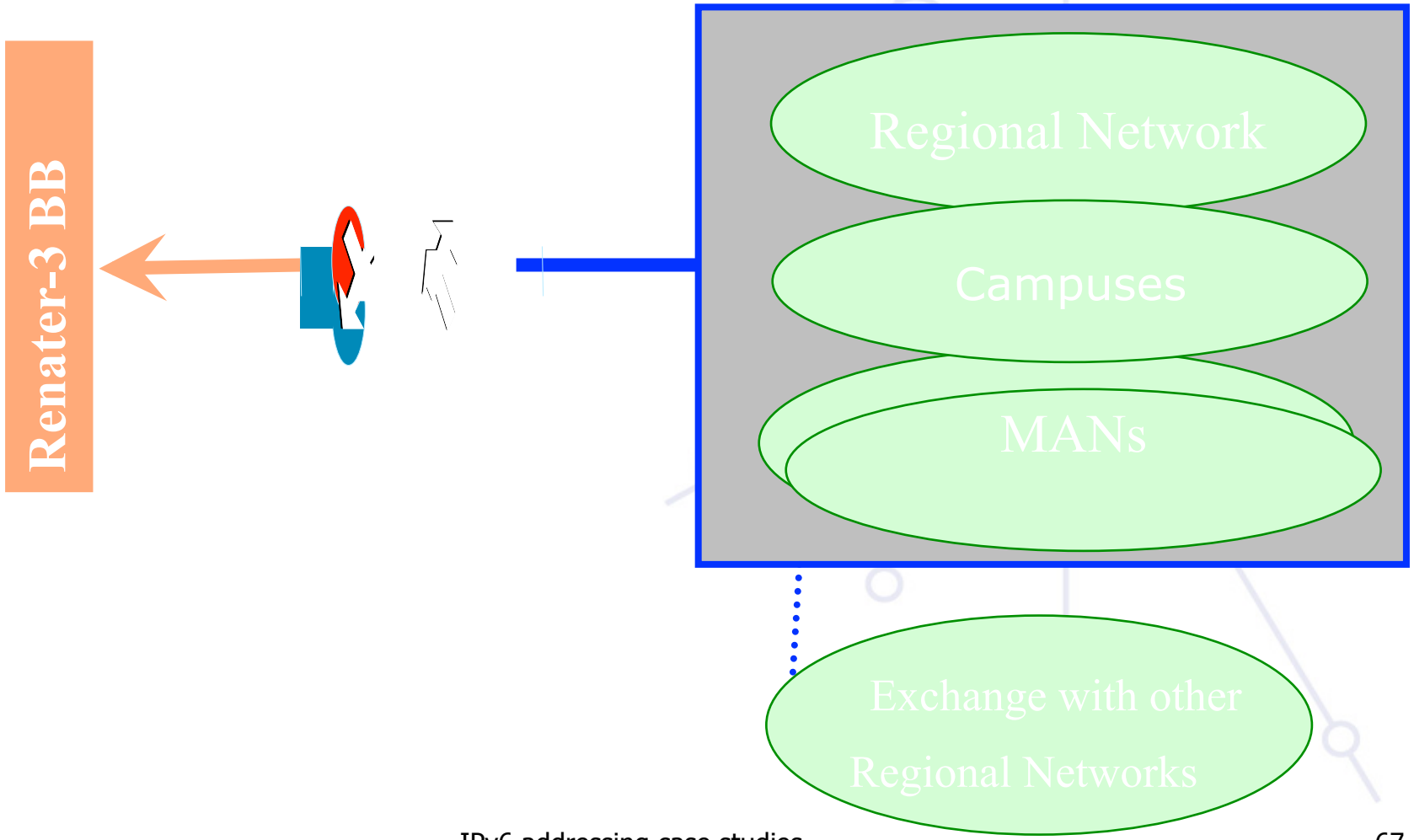
Renater-3: national backbone



Renater-3: international links



Renater-3 architecture



RENATER's Production IPv6 service

Why a production-like IPv6 service ?

ATM removed …

- Move all network services on a unique topology
- Do we want to forget about IPv6, IPv4 multicast … ?

Needs for an IPv6 transport

- Research projects using IPv6
- Sites with native IPv6 network
- → install a native IPv6 core
- → run both versions of IP on the same equipments

Monitor the IPv6 service in the same operational way than IPv4

Renater 3 : IPv6 Native support

2.5 Gbits/s backbone

30 Regional Nodes (NR)

Native IPv6 on all regional nodes

- Dual stack backbone → IPv4 and IPv6

Global IP Service

- IPv4 unicast and multicast
- IPv6 unicast
- IPv6 and IPv4 carried without any distinction

Goal : achieve for both versions of IP an equal level of

- Performance
- Availability
- Management
- Support

Addressing

Hierarchical addressing

Renater

- Prefix = 2001:0660::/32
- Allocated by the RIR (RIPE NCC)

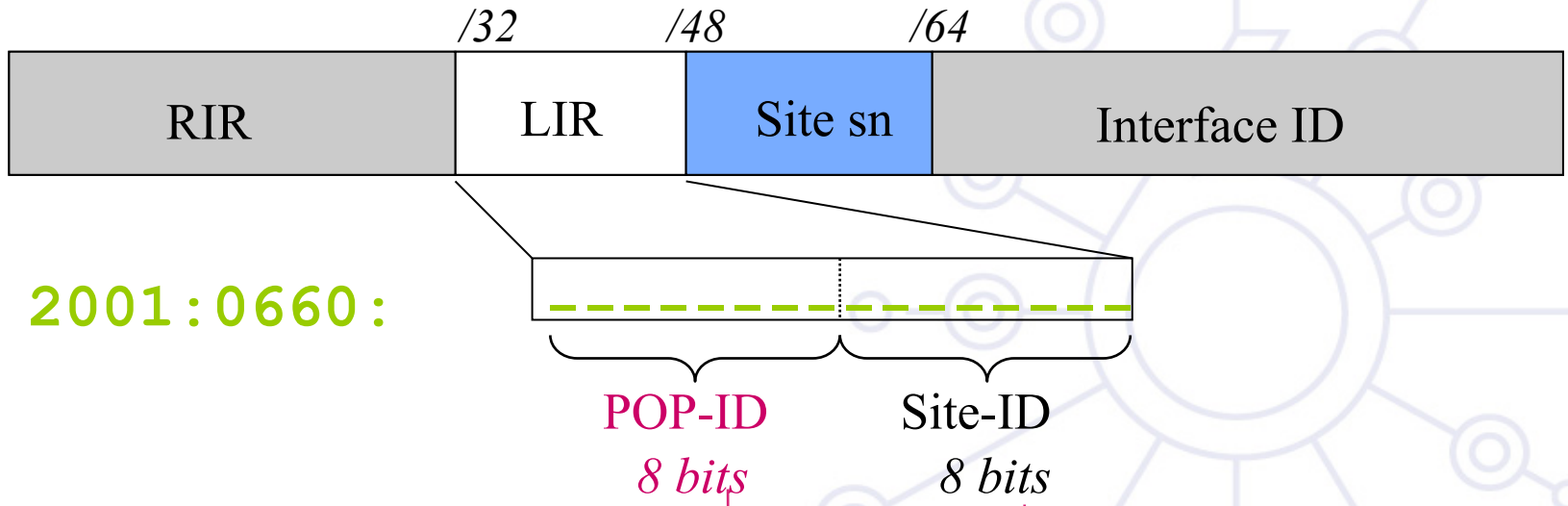
Regional Nodes

- POP-ID =2001:0660:xy::/40

Site

- Site-ID : a /48
  - from RN's prefix (/40) it's connected to
- Site-IDs allocated by Renater (LIR)
- 16 bits are reserved for the site topology

Addressing




```
2001:0660:3000:/40 Paris NRI
2001:0660:3300:/40 Paris Jussieu RI
2001:0660:4400:/40 Lille RI
2001:0660:5400:/40 Marseille RI
(…)
```

```
2001:0660:300x:/48
```

Example

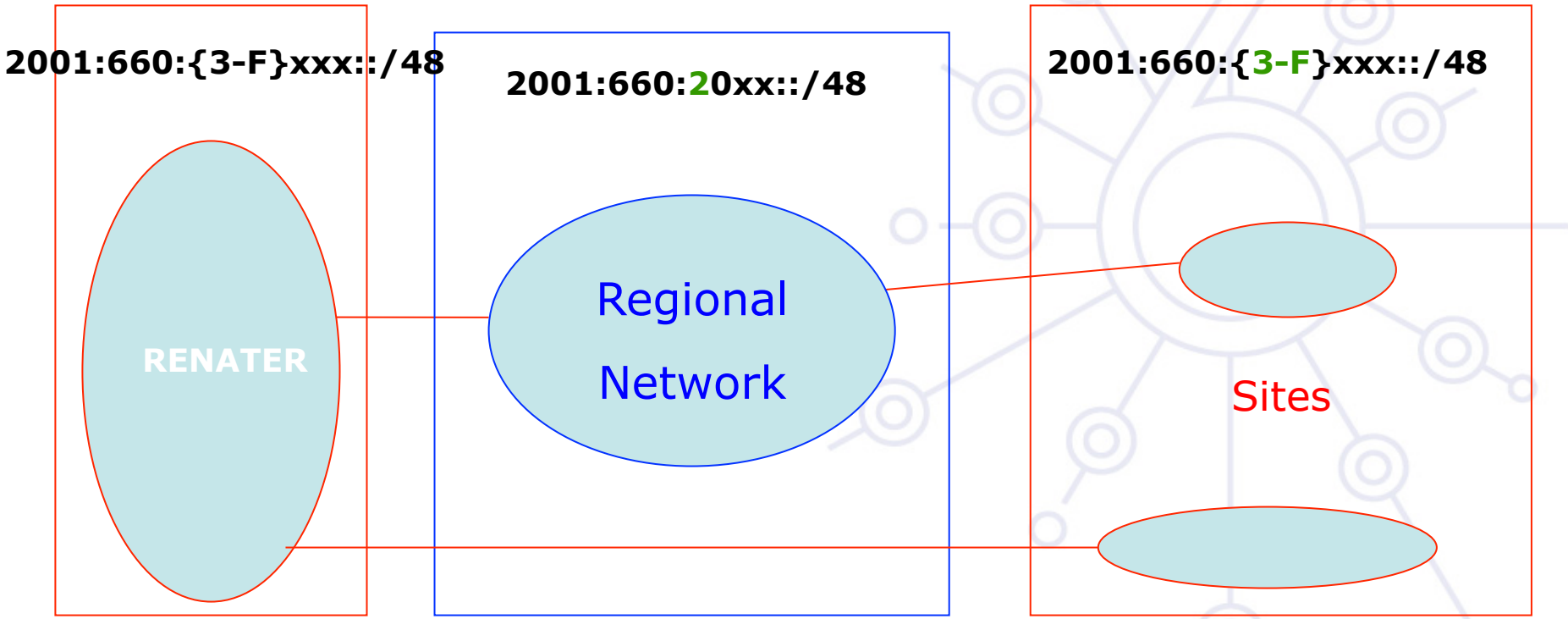
Renater's prefix	2001:0660::/32
POP-ID Strasbourg	2001:0660:4700::/40
Sites connected to Strasbourg's RI	2001:0660:4701::/48 2001:0660:4702::/48 …

Regional Nets Addressing

Two possibilities

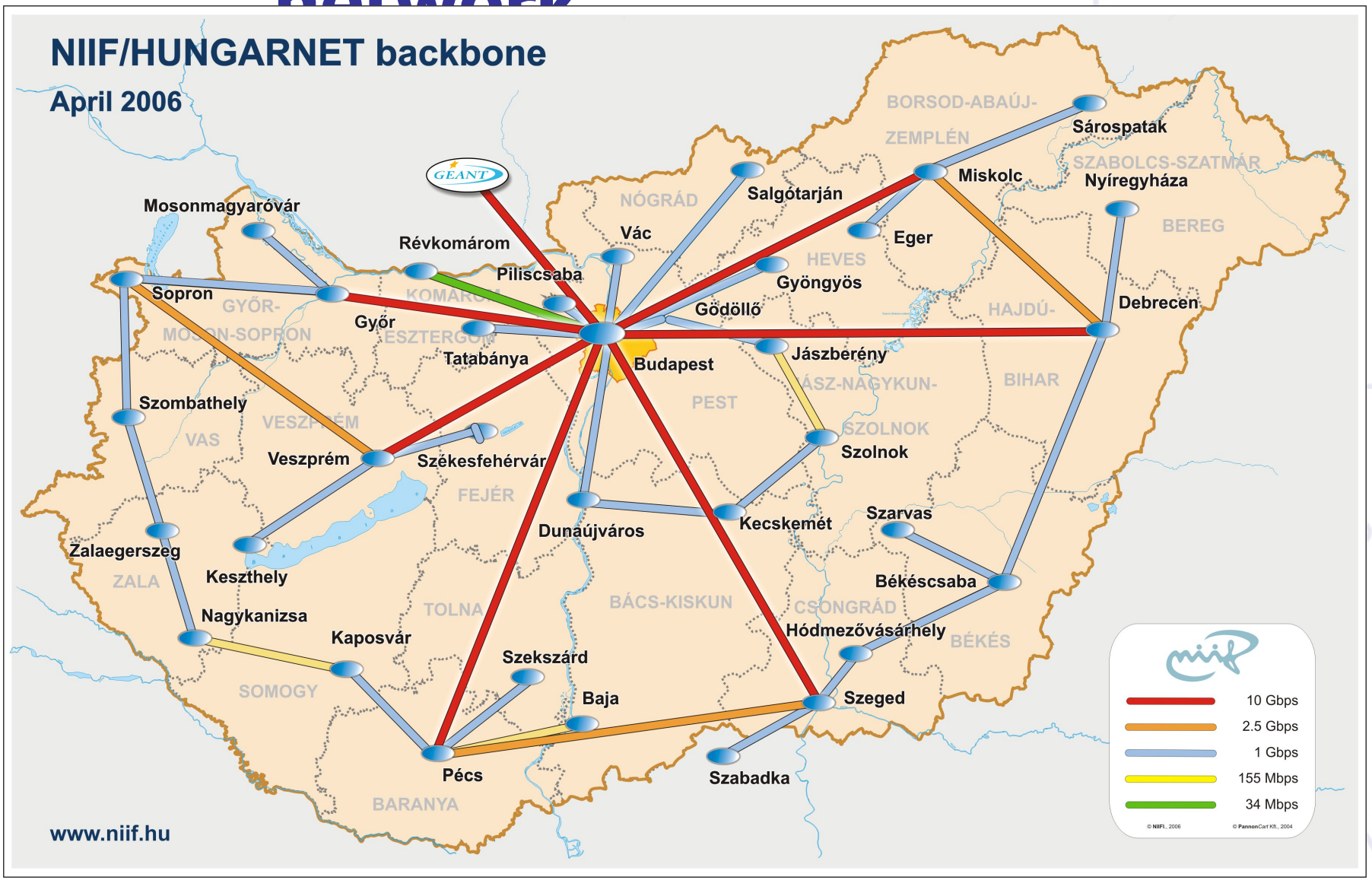
- Uses its own prefix (Commercial ISP)
- Uses Renater's address space
  - 2001:0660:2---::/48
- In both cases
  - Sites are addressed in Renater's sTLA
  - 2001:0660:{3-F}---::/48
  - Interco Network (site – Regional / MAN)
    - First /64 from the NLA-ID

Addressing scheme




deploy
NIFF/HUNGARNET IPv6
numbering

NIIF/HUNGARNET network



IPv6 deployment at NIIF/ Hungarnet

Initial IPv6 deployment:

- MPLS based backbone: 6PE with additional dual stack routers + sometimes tunnels at connected institutions

Second phase (2004):

- Router upgrade for HW based IPv6 forwarding
- Used features
  - Routing: IPv4 (unicast, multicast), IPv6 (unicast only), OSPFv2, OSPFv3, BGP, MPLS VPNs
  - Netflow, minimal QoS
  - IPv6 multicast with additional dual stack routers with tunnels

Third phase (2008):

- Software upgrade for IPv6 multicast support
- Netflow v9 support

IPv6 address space – based on flexible address allocation RFC3531

Location	IPv6 POP addressing:
CNTRL (Central)	2001:0738:0::/36
Gödöllő (Szent István University)	2001:0738:58::/44
BME (Budapest University of Technology and Economics)	2001:0738:2000::/44
KFKI (Research Institute on Physics	2001:0738:5000::/44
SZEGED (University of Szeged	2001:0738:7000::/44
MISKOLC (University of Miskolc)	2001:0738:6000::/44
PECS (University of Pécs)	2001:0738:7800::/44

Site addressing

Each site (including site infrastructure) gets /48:

- each NIIF managed site the 16 bit SLA is allocated based on the following convention: <SLA> = Address segmentation within the POP
- Where for <SLA>:
  - Range: 0000 till 00FF: Loopback addresses
  - Range: 0100 till 01FF: Intra-pop point-to-points (if it necessary to number it)
  - Range: 0200 till 02FF: connections to HUNGARNET member of institution
  - Range: 0300 till 03FF: external IPv6 connectivity (e.g. local IPv6 peering)
  - Range: 0400 till 04FF: POP Local Ethernets

IPv6 loopback addresses

- loopback address will also be used for operational and management actions on the equipment, and for routing protocols like iBGP, which will use these addresses for terminating the peering-sessions.
- Loopback addresses have typically a prefix mask of /128. This will avoid unnecessary unused addresses although address conservation is not really an issue in IPv6.

Link IPv6 addresses?

Not necessary!

- OSPFv3 is working with link-local
- IS-IS not necessary

IGP table can be quite small!

- Reduces the convergence time

Customer network is propagated into BGP (even if static routes are used)

- not with redistribute
- with network statement

Drawback:

- Traceroute can pick up arbitrary IPv6 address as a reply source -
- Avoid - configure on each point-to-point links:
  - `ipv6 unnumbered loopback0`

Link IPv6 addresses -other options

/127: not a good idea

- the all-zeros address is supposed to be the any router anycast address although this is not widely implemented today - see more RFC 3627

/126: works

- although the top 128 addresses are reserved for anycast stuff

/120: no clashes with top 128 anycast addresses

/112: alignment is on a nice colon boundary

/64: based on RFC 3513

- Allows to use EUI-64 addressing
- advisable for point-multipoint and broadcast link scenarios

Customers' Nets Addressing

Two possibilities

- Uses its own prefix (Commercial ISP)
- Uses NIIF/Hungarnet's address space
  - 2001:0738:<cutomer id>::/48 (/44 pre-allocated, /48 assigned)

Conclusion

Preparing an IPv6 addressing plan is a bit complex

Plan it in advance …

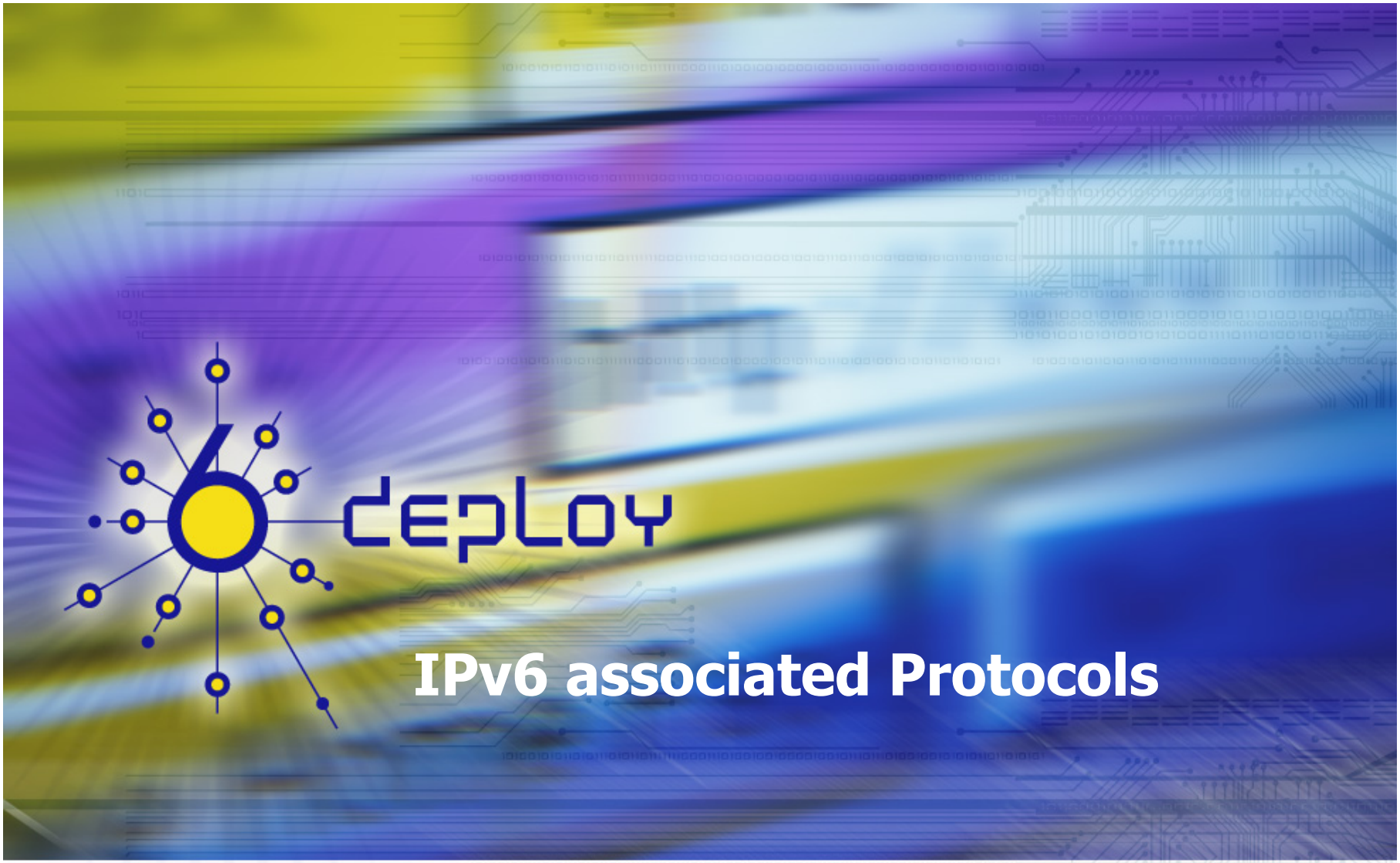
- Not forgetting your PoPs equipment (loopbacks, admin LANs, interconnects …)

Draw benefit from aggregation

- Smaller routing tables to manage (even in the core)
- Less prefixes to advertise to BGP peers

Lot of people have an experience yet …

- Not necessary to reinvent the wheel ;)


deploy
IPv6 associated Protocols

New Protocols

New features specified in IPv6 Protocol (RFC 2460 DS)

Neighbor Discovery (ND) (RFC 2461 DS)

Auto-configuration :

- Stateless Address Auto-configuration (RFC 2462 DS)
- DHCPv6: Dynamic Host Configuration Protocol for IPv6 (RFC 3315 PS)
- Path MTU discovery (pMTU) (RFC 1981 PS)

New Protocols (2)

MLD (Multicast Listener Discovery) (RFC 2710 PS)

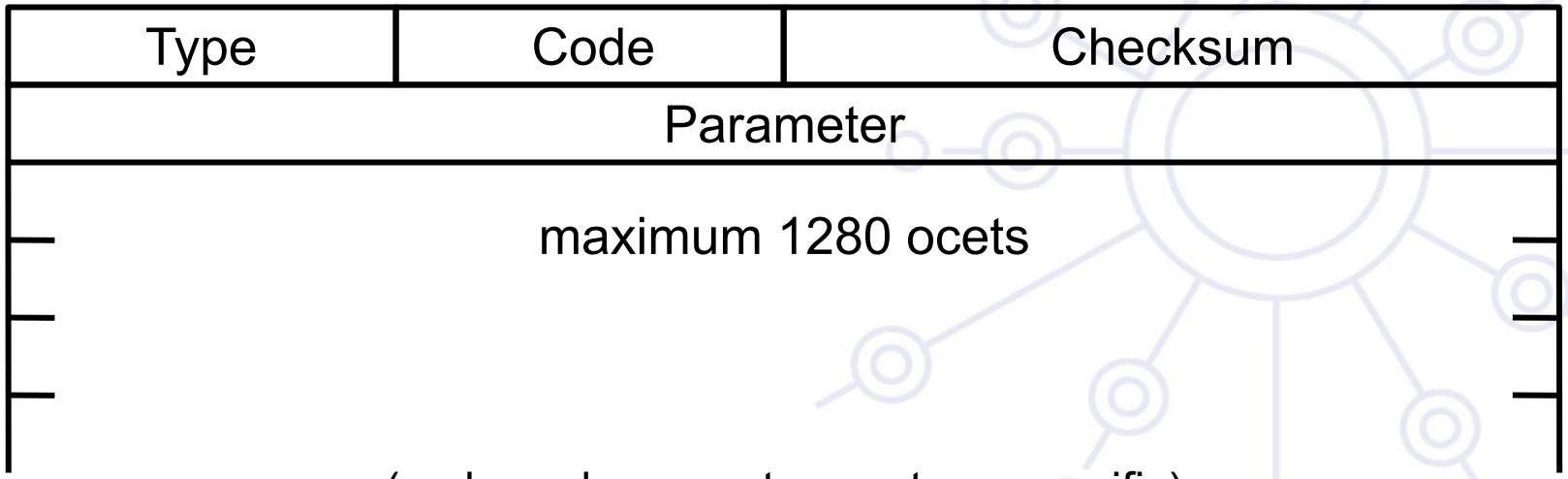
- Multicast group management over an IPv6 link
- Based on IGMPv2
- MLDv2 (equivalent to IGMPv3 in IPv4)

ICMPv6 (RFC 2463 DS) "Super" Protocol that :

- Covers ICMP (v4) features (Error control, Administration, …)
- Transports ND messages
- Transports MLD messages (Queries, Reports, …)

ICMP Hiba üzenetek

Közös formátum (mint IPv4):



(code and parameter are type-specific)

ICMP Hibaüzenet típusok

destination unreachable
- no route
- administratively prohibited
- address unreachable
- port unreachable

packet too big

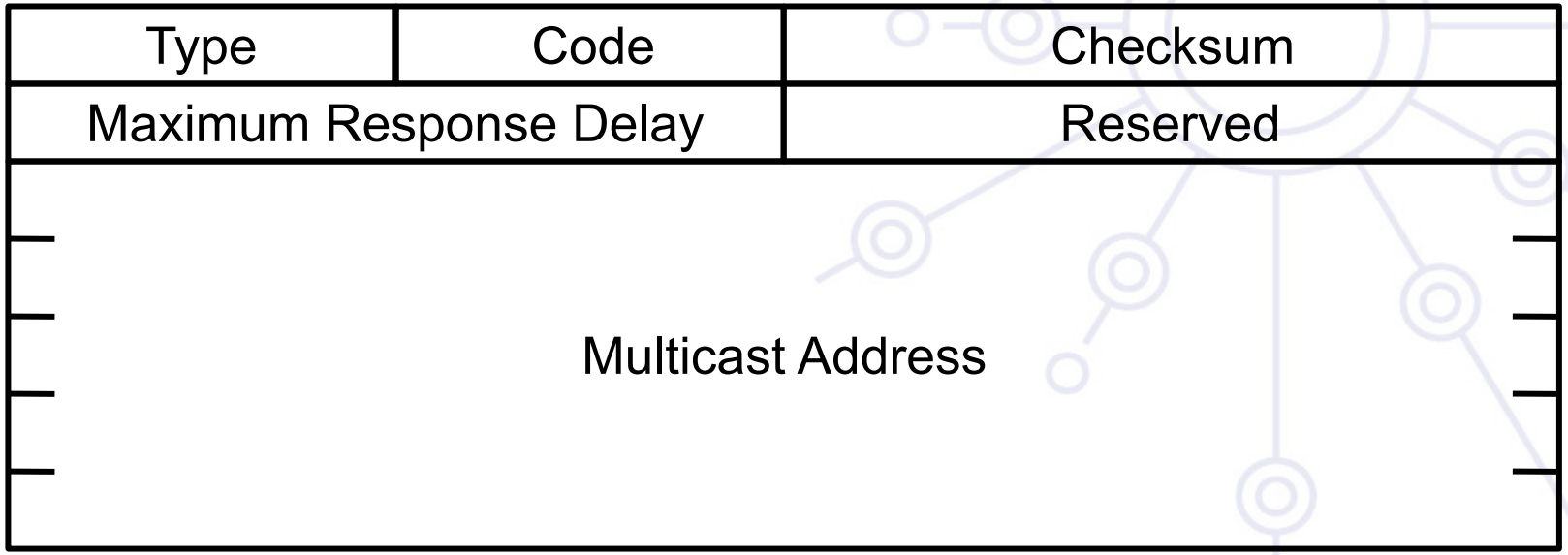
time exceeded

parameter problem
- erroneous header field
- unrecognized next header type
- unrecognized option

ICMP(v6) packets

Echo request & reply (like IPv4)

Multicast Listener Discovery packets:
query, report, done (like IGMP IPv4):



Neighbor Discovery

- **IPv6 nodes which share the same physical medium (link) use Neighbor Discovery (NDP) to:**

  discover their mutual presence
  determine link-layer addresses of their neighbors
  find routers
  maintain neighbors' reachability information (NUD)
  not directly applicable to NBMA (Non Broadcast Multi Access) networks ND uses multicast for certain services.

Neighbor Discovery (2)

Protocol features:

- Router discovery
- Prefix(es) discovery
- Parameters discovery (link MTU, Max Hop Limit, ...)
- Address auto-configuration
- Address resolution
- Next Hop determination
- Neighbor Unreachability Detection
- Duplicate Address Detection
- Redirect

Neighbor Discovery (3): Comparison with IPv4

It is the synthesis of:

- ARP
- R-Disc
- ICMP redirect
- ...

Neighbor Discovery (4)

ND specifies 5 types of ICMP packets :

- Router Advertisement (RA) :
  - periodic advertisement (of the availability of a router) which contains:
    - » list of prefixes used on the link (autoconf)
    - » a possible value for Max Hop Limit (TTL of IPv4)
    - » value of MTU
- Router Solicitation (RS) :
  - the host needs RA immediately (at boot time)

Neighbor Discovery (5)

- Neighbor Solicitation (NS):
  - to determine the link-layer @ of a neighbor
  - or to check its reachability
  - also used to detect duplicate addresses (DAD)
- Neighbor Advertisement (NA):
  - answer to a NS packet
  - to advertise the change of physical address
- Redirect :
  - Used by a router to inform a host of a better route to a given destination

Address Resolution

Find the mapping:

Destination IP @ ➔ Link-Layer (MAC) @

Recalling IPv4 & ARP

- ARP Request is broadcasted
  - e.g. `ethernet @: FF-FF-FF-FF-FF-FF`
  - Btw, it contains the Source's Link-local @
- ARP Reply is sent in unicast to the Source
  - It contains the Destination's Link-local @

Address Resolution (2) IPv6 with Neighbor Discovery

Every IPv6 node MUST join 2 special multicast groups for each network interface:

- All-nodes multicast group: `ff02::1`
- Solicited-node multicast group

Concatenation **of the prefix** `FF02::1:FF00:0/104` **with the last 24 bits of the IPv6 address**

Dst IPv6 **@:** 2001:0660:010a:4002:4421:21FF:FE24:87c1

Sol. Mcast **@:** FF02:0000:0000:0000:0000:0001:FF24:87c1

ethernet: 33-33-FF-24-87-c1

Address Resolution (3)
IPv6 with Neighbor Discovery

- H1 knows IP address of H2 (IP2) and wants to know its MAC address (MAC2)
- H1 builds the sollicited multicast address of IP2 : Multi (IP2)
- H1 sends « Neighbor sollicitation » message to this sollicited multicast IPv6 address
- At **link level**, NS packet is sent to a **multicast address** instead of broacast

Address Resolution (4)
IPv6 with Neighbor Discovery

- Ethernet manages multicast – Not always implemented
- Ethernet frame is often broadcasted on the link
- Only H2 is destination of the ethernet frame and sends the « Neighbor Sollicitation » packet to its IPv6 stack
- H2 replies sending a unicast « Neighbor Advertisement » message to H1. This message contains the link layer address of H2.

Path MTU discovery (RFC 1981)

Derived from RFC 1191, (IPv4 version of the protocol)

Path **: set of links followed by an IPv6 packet between source and destination**

link MTU **: maximum packet length (bytes) that can be transmitted on a given link without fragmentation**

Path MTU **(or pMTU) = min { link MTUs } for a given path**

Path MTU Discovery = automatic pMTU discovery for a given path

Path MTU discovery (2)

Protocol operation

- makes assumption that pMTU = link MTU to reach a neighbor (first hop)
- if there is an intermediate router such that link MTU < pMTU ➔ it sends an ICMPv6 message: "Packet size Too Large"
- source reduces pMTU by using information found in the ICMPv6 message

=> Intermediate equipments aren't allowed to perform packet fragmentation

Auto-configuration

Hosts should be plug & play

Use ICMPv6 messages (Neighbor Discovery)

When booting, the host asks for network parameters:

- IPv6 prefix(es)
- default router address(es)
- hop limit
- (link local) MTU
- ...

Auto-configuration (continued)

Only routers have to be manually configured

- but work on **prefix delegation** is in progress
  (draft-ietf-ipv6-prefix-delegation-requirement-01.txt)

Hosts can get automatically an IPv6 address

- BUT it is not automatically registered in the DNS
- If the address is always the same: may be manually registered

NEED for DNS Dynamic Update

(RFC 2136 PS and RFC 3007 PS) for IPv6

- Security issues …

Stateless auto-configuration

IPv6 Stateless Address Auto-configuration

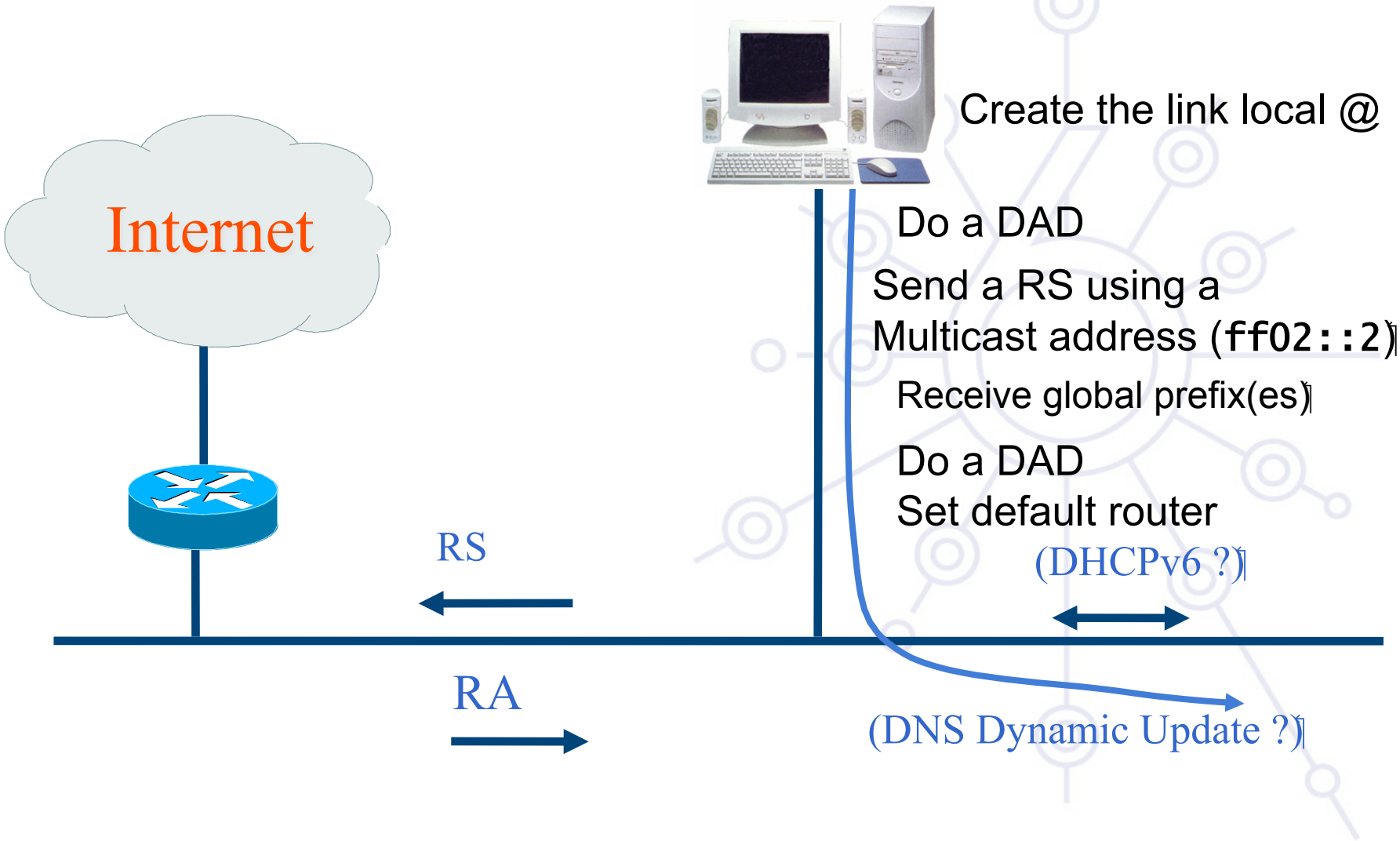
- RFC 2462 DS
- Does not apply to routers

Allows a host to create a global IPv6 @ from:

- Its interface identifier = EUI64(MAC @)
- router advertisements coming from router(s) on the link

=> GA = concat (RA, EUI64)

Auto-configuration example



Interface Identifier: probléma

IEEE 24 bit OUI azonosítja a hardwaret

(http://standards.ieee.org/regauth/oui/oui.txt)

Interface ID alkalmazható a felhasználó követésére:

- A prefix megváltozik, de az interface ID ugyanaz marad!

Privacy extensions (RFC 3041)

- Interfész ID megváltoztatható
- MD5 algorithmus - véletlenszám/tároló
- Biztonsági probléma?

Privacy extension (RFC 4941)

- A privacy extension nincsen default bekapcsolva
- DAD minden később generált címre
- Per prefix engedélyezhető a privacy extension
- Nemcsak MD5 hash algoritmus használható

Stateless Autoconfiguration: javaslatok

Csak routereket kell manuálisan konfigurálni

- prefix delegáció – hogy ezt is meglehessen spórolni

  (http://www.ietf.org/rfc/rfc3633.txt)

Hosztok automatikusan kapnak IPv6 címet

- DE nincsenek automatikusan DNS-be regisztrálva
  - Kivéve Windows esetén

Szervereket célszerű manuálisan konfigurálni

Statefull Autoconfiguration DHCPv6

Dynamic Host Configuration Protocol for IPv6

- Defined in RFC 3315
- Stateful counterpart to IPv6 Stateless Address Autoconfiguration.

According to RFC 3315 DHCPv6 is used when:

- No router is found
- Or if Router Advertisement message enables use of DHCP
  - Using ManagedFlag and OtherConfigFlag

There is also 'stateless DHCPv6' (RFC3736)

- Used by clients that already have an address
- Based upon standard DHCPv6

Statefull Autoconfiguration DHCPv6 /2

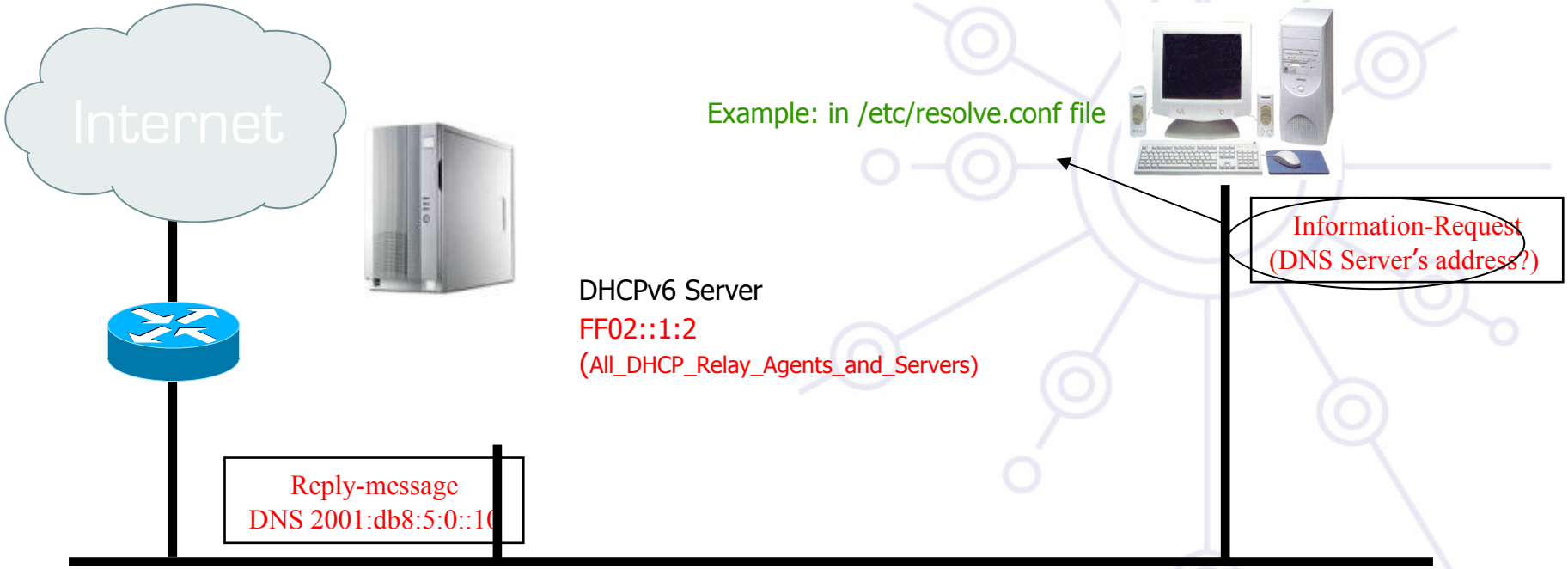
DHCPv6 works in a client / server model

- Server
  - Responds to requests from clients
  - Optionally provides the client with:
    - IPv6 addresses
    - Other configuration parameters (DNS servers…)
  - Listens on the following multicast addresses:
    - All_DHCP_Relay_Agents_and_Servers (FF02::1:2)
    - All_DHCP_Servers (FF05::1:3)
  - Provides means for securing access control to network resources
  - Usually storing client's state, though 'stateless operation' is also possible (the usual method used for IPv4 today)

Statefull Autoconfiguration DHCPv6 /3

- Client
  - Initiates requests on a link to obtain configuration parameters
  - Uses its link local address to connect the server
  - Sends requests to FF02::1:2 multicast address (All_DHCP_Relay_Agents_and_Servers)
- Relay agent
  - A node that acts as an intermediary to deliver DHCP messages between clients and servers
  - On the same link as the client
  - Listens on multicast address:
    - All_DHCP_Relay_Agents_and_Servers (FF02::1:2)

Statefull Autoconfiguration DHCPv6 /4



IPv6 support in the DNS

DNS Extensions for IPv6

- RFC 1886 (PS) → RFC 3596 (DS) (upon successful interoperability tests)
- **AAAA** (RFC 3596): forward lookup ('Name → IPv6 Address'):
  - Equivalent to '**A**' record
  - Example:

```
                ns3.nic.fr.      IN      A       192.134.0.49
                                 IN      AAAA
      2001:660:3006:1::1:1
```

- **PTR** : reverse lookup ('IPv6 Address → Name'):
  - Reverse tree equivalent to `in-addr.arpa`
    - Nibble (4 bits) boundary
    - New tree: `ip6.arpa` (RFC 3596), used
    - Former tree: `ip6.int` (RFC 1886), obsolete
  - Example:

```
$ORIGIN 1.0.0.0.6.0.0.3.0.6.6.0.1.0.0.2.ip6.{int,arpa}.
1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0   PTR   ns3.nic.fr.
```

IPv6 DNS and root servers

DNS root servers are critical resources!

13 roots « around » the world (#10 in the US)

As of 04/02/2008, 6 of 13 root servers are IPv6 enabled and reachable via IPv6 networks (A, F, H, J, K, M).

IPv6 DNS operation

Recent BIND, NSD, PowerDNS - OK

Microsoft Windows XP default resolver only queries over IPv4 transport:

- Install BIND 9 for Windows XP and use BINDs resolver

The target today IS NOT the transition from an IPv4-only to an IPv6-only environment

- Start by testing DNSv6 on a small network and get your own conclusion that DNSv6 is harmless
- Deploy DNSv6 in an incremental fashion on existing networks
- DO NOT BREAK something that works fine (production IPv4 DNS)!

Internal and External Routing

Agenda

Internal Routing

- RIPng
- IS-IS
- OSPFv3

External Routing

- Multiprotocol BGP

RIPng

Same as IPv4

- Based on RIPv2
- Distance vector, max. 15 hop, split-horizon, …

It's an IPv6 only protocol

- In a dual-stack environment, running RIP, you'll need RIP (IPv4) and RIPng (IPv6)

IPv6 related functionality

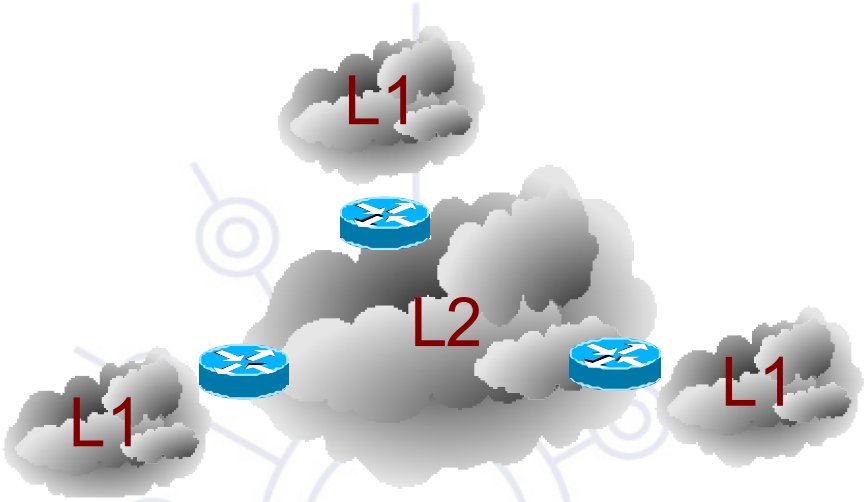
- Uses IPv6 for transport
- IPv6 prefix, next-hop IPv6 address
- For RIP updates, uses multicast address FF02::9

ISISv6

OSI Protocol

Based on two levels

- L2 = Backbone
- L1 = Stub
- L2L1= interconnect L2 and L1



Runs on top of CNLS

- Each IS device still sends out LSP (Link State Packets)
- Send information via TLV's (Tag/Length/ values)
- Neighborship process is unchanged

Major operation remains unchanged

ISISv6 #2

Updated features:

- Two new Tag/Length/Values (TLV) for IPv6
  - IPv6 Reachability
  - IPv6 Interface Address
- New network Layer Identifier
  - IPv6 NLPID

OSPFv3

OSPFv3 = OSPF for IPv6

Based on OSPFv2

Topology of an area is invisible from outside the area

- LSA flooding is bounded by area
- SPF calculation is performed separately for each area

All areas must have a connection to the backbone

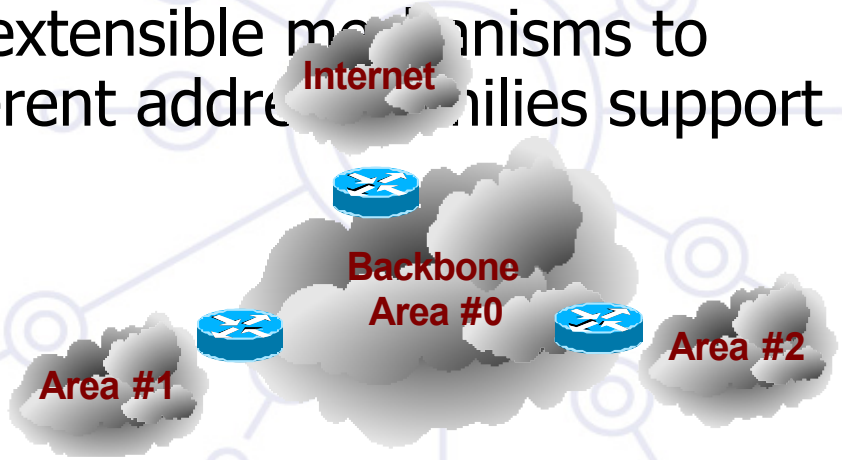
OSPFv3

OSPFv3 is an IPv6-only protocol

- In a dual-stack environment, running OSPF, you'll need OSPFv2 (IPv4) and OSPFv3 (IPv6)
- Work-in-progress about extensible mechanisms to enable OSPFv3 with different address families support

Updated Features

- Runs directly over IPv6
- Distributes IPv6 prefixes
- New LSA types
- Uses Multicast addresses
  - ALLSPFRouters (FF02::5)
  - ALLDRouters (FF02::6)



Multiprotocol BGP

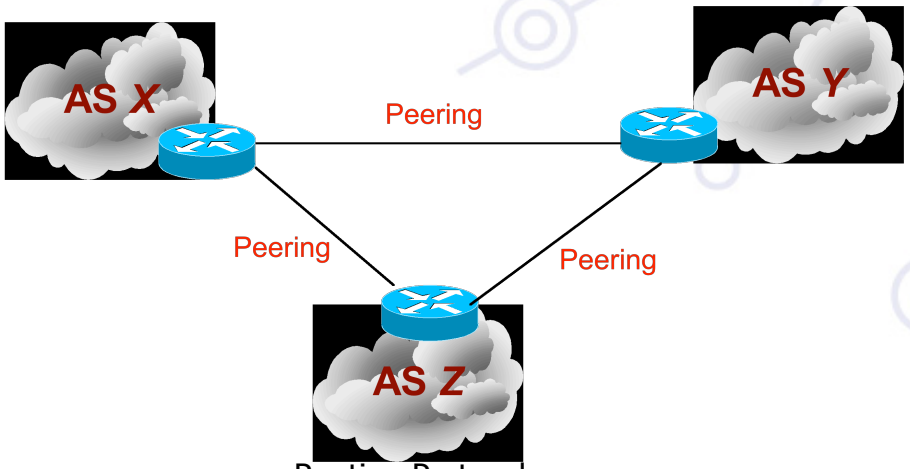
Exterior Gateway Protocol

Connect separate routing domains that contain independent routing policies (and AS numbers)

Carries sequences of AS numbers, indicating path (for each route)

Supports the same features and functionality as IPv4 BGP

Multiple addresses families: IPv4, IPv6, unicast, multicast



Multiprotocol BGP

BGP4 carries only 3 types of information wich is truly IPv4 specific:

- NLRI in the UPDATE message contains an IPv4 prefix
- NEXT_HOP attribute in the UPDATE message contains an IPv4 address
- BGP ID in AGGREGATOR attribute

Multiprotocol BGP

RFC 4760 defines multi-protocols extensions for BGP4

- this makes BGP4 available for other network layer protocols (IPv6, MPLS…)
- New BGP4 attributes:
  - MP_REACH_NLRI
  - MP_UNREACH_NLRI
- Protocol Independent NEXT_HOP attribute
- Protocol Independent NLRI attribute

Conclusions

All major routing protocols have stable IPv6 Support, and no major differences with IPv4

In a dual-stack environment, running OSPF, you'll need OSPFv2 (IPv4) and OSPFv3 (IPv6). It may change in a near future.

In a dual-stack environment, running RIP, you'll need RIPv1/RIPv2 (IPv4) and RIPng (IPv6)

Routing Stats (IPv6 vs. IPv4, globally)

(11/09/2008)	IPv6	IPv4
ROUTES	**1505**	**281136**
AGGREGATED ROUTES	**1400 (93,02%)**	**170595 (60,68%)**
AUTONOMOUS SYSTEMS	**1131**	**29345**

source: www.cidr-report.org


deploy
IPv6 deployment
In campus and
Service Provider environment

Warning …

This module is under work (it's still evolving)

- ***here are ideas drawn from experienced people***
- ***it's out of scope to recommend every one to do the same***
- ***Every campus is specific and thinking what to do and how to do it beforehand is a must***

Outline

Campus deployment strategy

Campus IPv6 address allocation

Campus deployment topology - options

Campus services

Service provider deployment considerations

Outline

Campus deployment strategy

Campus IPv6 address allocation

Campus deployment topology - options

Campus services

Service provider deployment considerations

Questions to solve

Will a coexistence mechanism be needed ?

If yes,

Which mechanism to use / to forbid

reasons of this (these) choice(s) ?

What policy to specify ?

How to implement it ?

Various Campus transition approaches

IPv4 will be used for years after IPv6 has been deployed

Then both versions of the IP protocol will have to coexist

Dual Stack

- Servers/clients speaking both protocols
- Application/service can select either protocol to use

Tunneling ("connecting IPv6 clouds")

- IPv6 packet is data payload of IPv4 packet/or MPLS frames

Translation methods ("IPv4<->IPv6 services")

- Layer 3: Rewriting IP header information (NAT-PT)
- Layer 4: Rewriting TCP headers
- Layer 7: Application layer gateways (ALGs)

Benefits of dual-stack deployment

By deploying dual-stack, you can test IPv6-only devices/ services without disrupting IPv4 connectivity

Dual stack IPv6 + IPv4 NAT: legacy IPv4 applications (email, www) can be used next to new IPv6 applications (p2p, home networking, …)

- IPv6 offers the next generation of applications

Campus deployment plan /1

1. **Obtain global IPv6 address space from your ISP**
   - NRENs usually have a /32 prefix from RIPE NCC/RIRs
   - Universities/customers will get a /48 prefix from NRENs/LIRs
2. **Obtain external connectivity**
   - You can do dual-stack connectivity
   - Many universities will use a tunnel to get IPv6 service
     - in this case be sure that nobody can abuse your tunnel – use filtering

Campus deployment plan /2

3. **Internal deployment**
   - Determine an IPv6 firewall/security policy
     - The IPv4 firewall/security policy is a good start
   - Develop an IPv6 address plan for your site
   - Determine an address management policy (RA/DHCPv6?)
   - Migrate to dual-stack infrastructure on the wire
     - Network links become IPv6 enabled
   - Enable IPv6 services and applications
     - Starting with DNS
   - Enable IPv6 on host systems (Linux, WinXP, Vista, Mac OS X…)
   - Enable management and monitoring tools

Outline

Campus deployment strategy

Campus IPv6 address allocation

Campus deployment topology - options

Campus services

Service provider deployment considerations

Campus Addressing

Most sites will receive /48 assignments:

16 bits left for subnetting - what to do with them?

Two main questions to answer:

⇒ **How many topologically different "zones" can be identified ?**

- Existing ones or new ones to be created for whatever (good) reason

⇒ **How many networks (subnets) are needed within these zones ?**

Example network « zones »

Zone description	Nb of subnets
Upstream interco and infrast	16
Administration services	4
Medical Sciences dept	32
Dept A	16
Dept B	16
…	

Campus Addressing - site level subnetting - methods -1

1. Sequentially, e.g.

- 0000
- 0001
- …
- FFFF

- 16 bits = 65535 subnets

0020/60

0030/60

Subnet ID	Zone description
0000 / 60	BB Infrastructure
0010 / 60	Administration
0020 / 59	Medical Sciences dept
0040 / 60	Dept A
0050 / 60	Dept B
…	…

⇒ *Reserve prefixes for further allocations*

Campus Addressing - site level subnetting - methods 2

2. Following existing IPv4:

- Subnets or combinations of nets & subnets, or VLANs, etc., e.g.
- IPv4 subnets:
  - `152.66.`**`60`**`.0/24` `003c`
  - `152.66.`**`91`**`.0/24` `005b`
  - `152.66.`**`156`**`.0/24` `009c`
- VLANs:
  - `VLAN id 100` `0100 (w/o decimal/hex conversion)`

    `or 0064 (w dec/hex conversion)`

⇒ *Best to start thinking about it*

Campus Addressing - site level subnetting - methods 3

3. Topological/aggregating

reflecting wiring plants, supernets, large broadcast domains, etc.

- Main library = 0010/60
  - Floor in library = 001a/64
- Computing center = 0200/56
  - Student servers = 02c0/64
- Medical school = c000/52
- and so on. . .

Example network - topological aggregation + sequential allocation

Zone description	Nb of subnets
Upstream interco and infrast	16
Administration services	4
Medical Sciences dept	32
Dept A	16
Dept B	16
…	

IPv6 subnet prefix allocations (ex.)

Subnet ID	Subnet prefix allocation	Description
0000 / 60		BB Infrastructure
	0000/64	Upstream interconnection
	0001/64	Campus architecture (DMZ)
	…	
	000B/64	Campus architecture
	…	
	000F	…
0010 / 60		Administration
	0010/64	Campus interco
	0011/64	Registration
	0012/64	Finance dept
	…	…

IPv6 subnet prefix allocations ex. /2

Subnet ID	Subnet prefix allocation	Description
0020 / 60		Medical Sciences dept
	0020/64	Upstream interconnection
	0021/64	Nobel group
	…	
0030 / 60	***Reserved***	***Medical Sciences dept***
0040 / 60		Dept A
…		…

New Things to Think About

You can use "all 0s" and "all 1s"! (0000, ffff)

You're not limited to the usual 254 hosts per subnet!

- LANs with lots of L2 switch allow for larger broadcast domains (with tiny collision domains), perhaps thousands of hosts/LAN…

No "secondary address" (though >1 address/ interface)

No tiny subnets either (no /30, /31, /32)

- plan for what you need for backbone blocks, loopbacks, etc.

You should use /64 per links

- Especially if you plan to use autoconfiguration!
- If you allocate global addressess interconnection links - not necessary in every case

New Things to Think About /2

Every /64 subnet has far more than enough addresses to contain all of the computers on the planet,

and with a /48 you have 65536 of those subnets

- use this power wisely!

With so many subnets your IGP may end up carrying thousands of routes

- consider internal topology and aggregation to avoid future problems.

New Things to Think About /3

Renumbering will likely be a fact of life.

Although v6 does make it easier, it still isn't pretty. . .

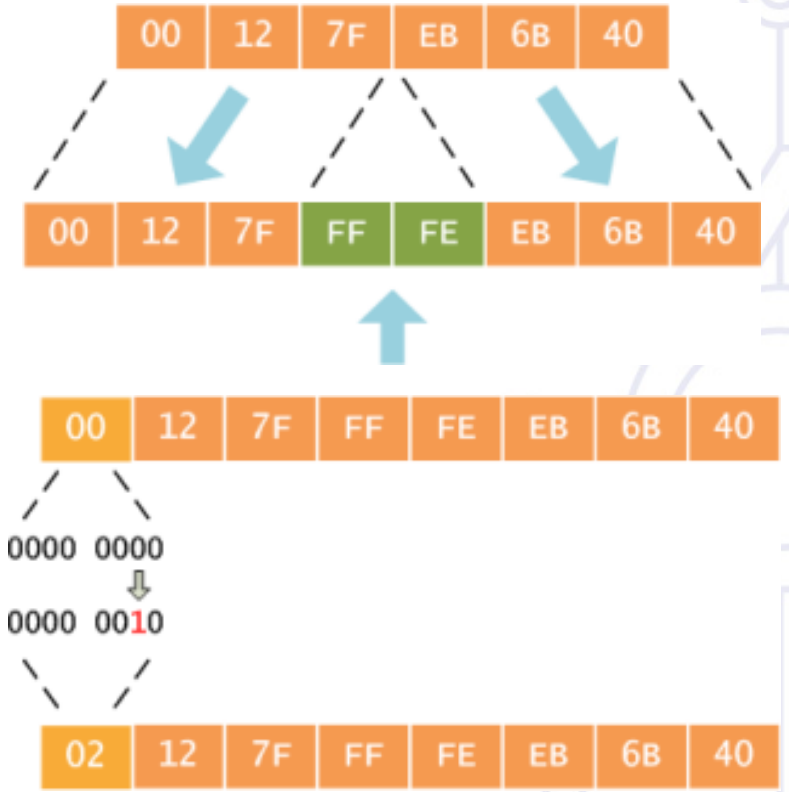
- Avoid using numeric addresses at all costs
- Avoid hard-configured addresses on hosts except for servers (this is very important for DNS servers) – use the feature that you can assign more than one IPv6 address to an interface (IPv6 alias address for servers)
- Anticipate that changing ISPs will mean renumbering
- An ISP change will impact the first 48 bits, you can keep the last 80 unchanged in every host/server's address.

Address conservation usually not an issue

DHCPv6 might help

New Things to Think About /4

Recap from EUI-64:



- The motivation for inverting the **'u'** bit when forming the interface identifier is to make it easy for system administrators to hand configure local scope identifiers. This is expected to be case for serial links, tunnel end-points and servers, etc. simply ::1, ::2, etc

Campus Addressing - address assignment

- Which address assignment to use?
  - Autoconfiguration - IEEE provides uniqueness
  - DHCPv6 - central management provides uniqueness
  - Manual - 7th bit of IID should be 0

Methods to manually assign addresses:

IID part	Description
0000::<smallnumber>	Easy to remember allocations
0080:vvww:yyzz:XXXX/112	Automaticaly assigned to vv.ww.yy.zz IPv4 address: /112 belongs to a IPv4 host - good for service virtualisation

Stateless address autoconfiguration

Discussion of SLAAC advantages and disadvantages

DHCP (1)

IPv6 has stateless address autoconfiguration but DHCPv6 (RFC 3315) is available too

DHCPv6 can be used both for assigning addresses and providing other information like nameserver, ntpserver etc

If DHCPv6 is not used for address allocation, no state is required on server side and only part of the protocol is needed.

This is called *Stateless DHCPv6* (RFC 3736)

Some server and client implementations only do Stateless DHCPv6 while others do the full DHCP protocol

- Some vendors don't implement yet a DHCPv6 client (MacOS X, …)

The two main approaches are

- Stateless address autoconfiguration with stateless DHCPv6 for other information
- Using DHCPv6 for both addresses and other information to obtain better control of address assignment

DHCP (2)

One possible problem for DHCP is that DHCPv4 only provides IPv4 information (addresses for servers etc) while DHCPv6 only provides IPv6 information. Should a dual-stack host run both or only one (which one)?

Several vendors working on DHCP integrations - several implementations available at the moment

- DHCPv6 http://dhcpv6.sourceforge.net/
- dibbler http://klub.com.pl/dhcpv6/
- NEC, Lucent etc. are working on their own implementations
- KAME-WIDE DHCPv6 http://sourceforge.net/projects/wide-dhcpv6/
- ISC DHCPv6 https://www.isc.org/software/dhcp

Cisco routers have a built-in stateless server that provides basic things like nameserver and domain name (also SIP server options).

DHCP can also be used between routers for prefix delegation (RFC 3633). There are several implementations. E.g. Cisco routers can act as both client and server

Outline

Campus deployment strategy

Campus IPv6 address allocation

Campus deployment topology - options

Campus services

Service provider deployment considerations

IPv6 deployment options

The simplest

- deploy dual stack network environment

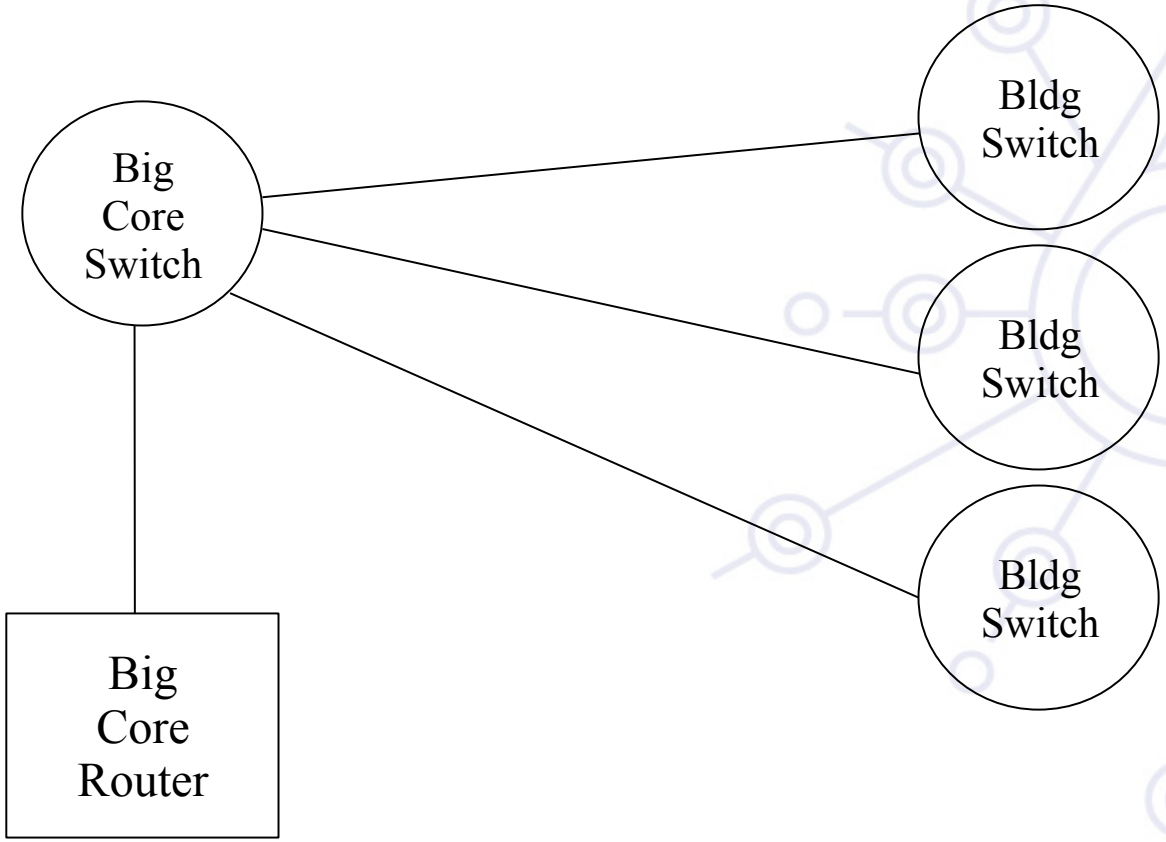
If the hosts/services are not dual stack enabled

- It does not break anything
- this tends to be a false assumption (Windows Vista, Mac OS X shipped with IPv6 enabled)

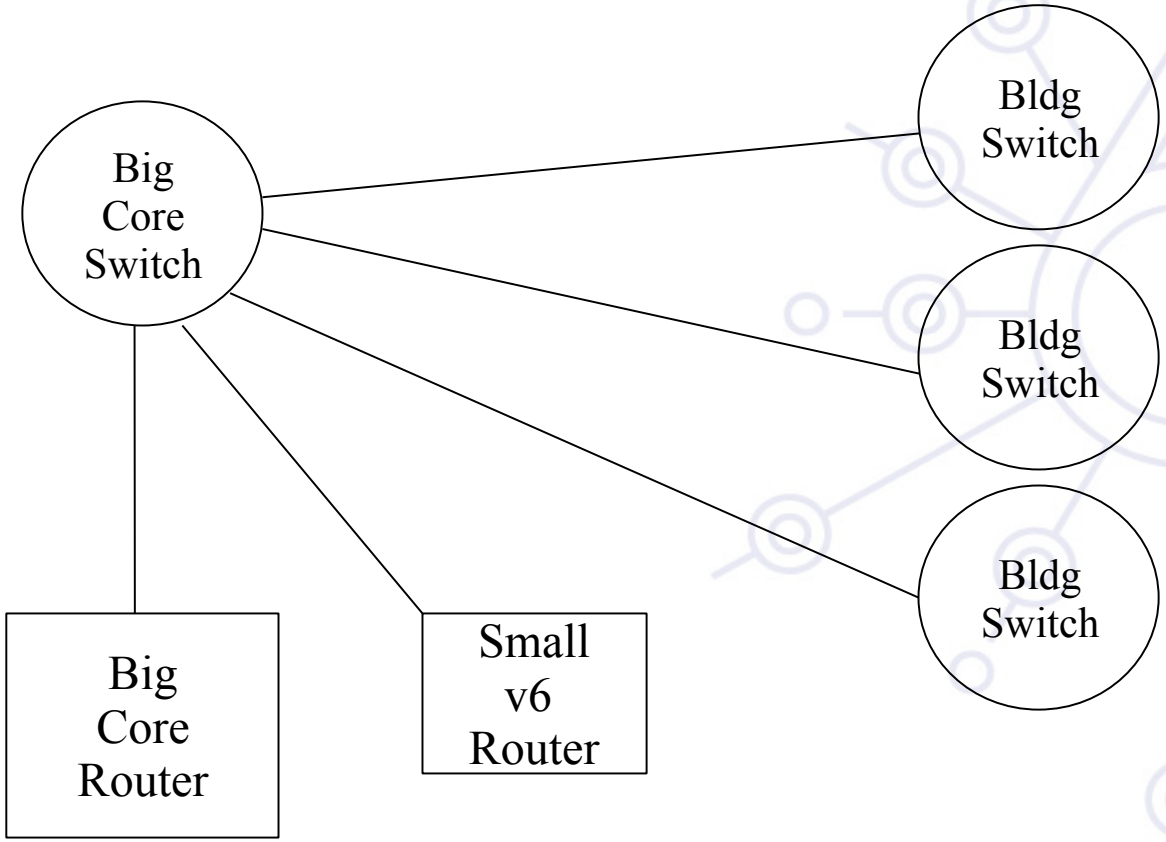
If the L3 devices cannot cope with IPv6 or administrators are not in favor of upgrading the router

- Add additional IPv6 capable L3 device(s)
- Investment money is usually a problem, but you can do some engineering with simple (low cost) PCs.

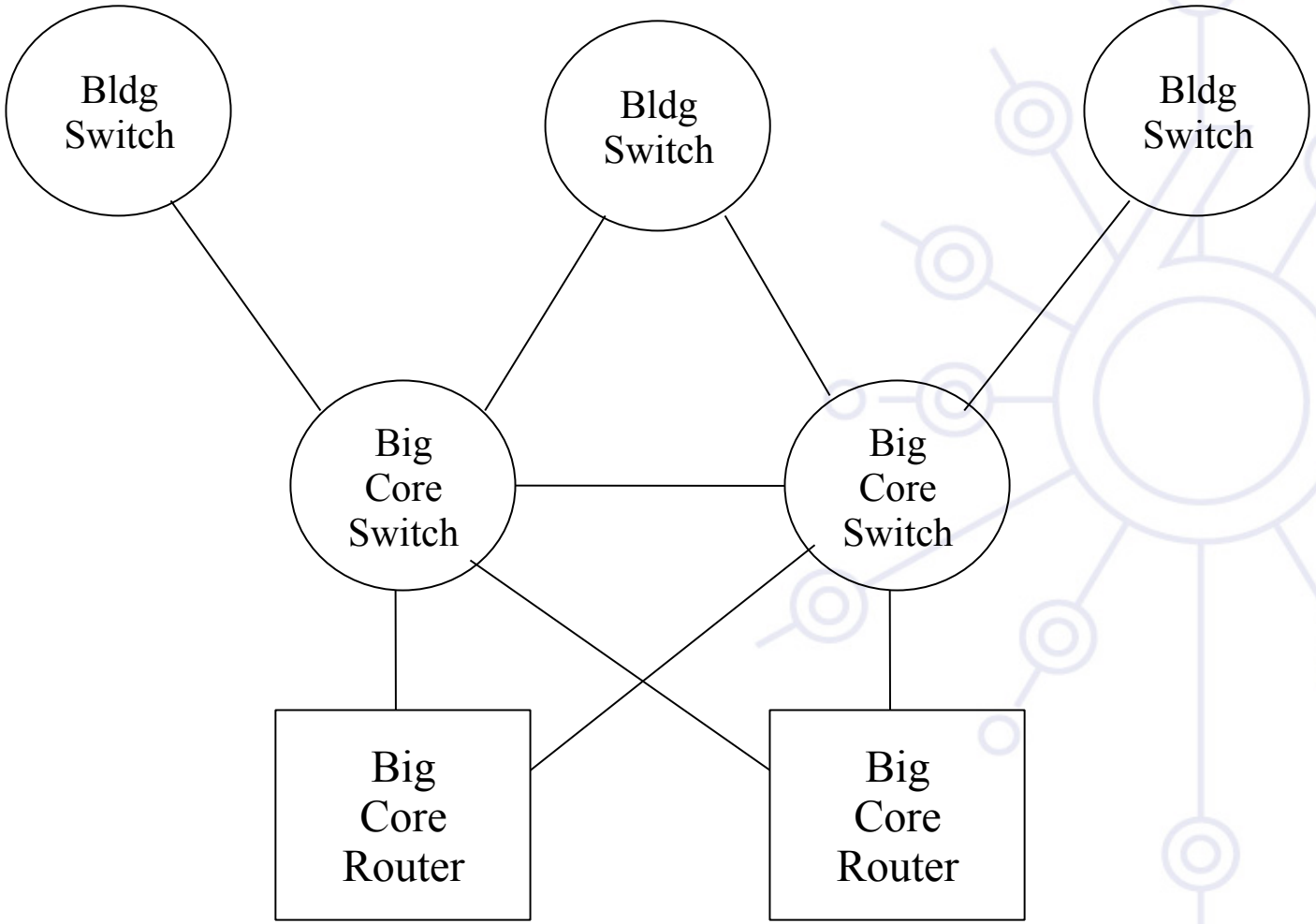
Layer-2 Campus - 1 Switch



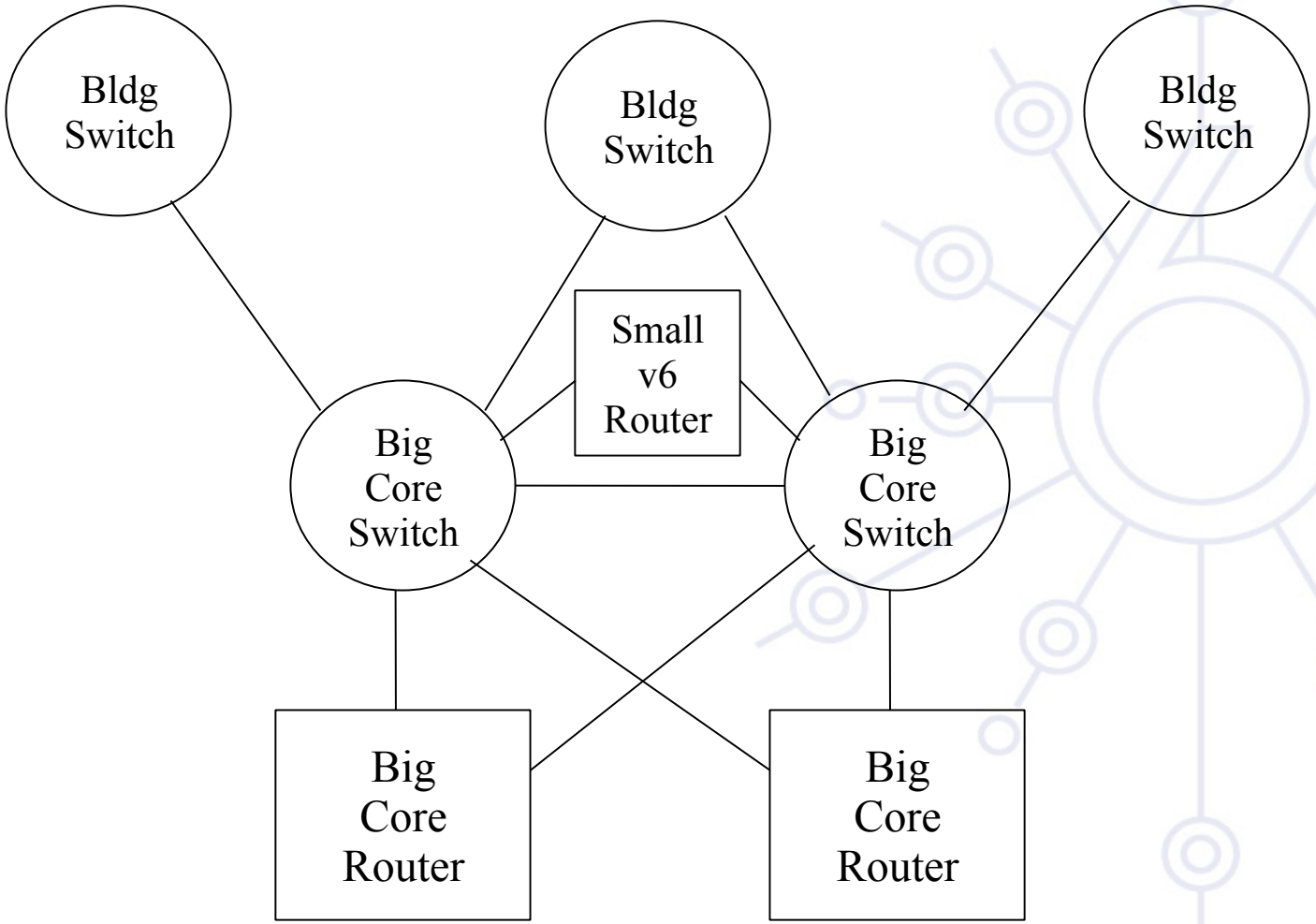
Layer-2 Campus - 1 Switch



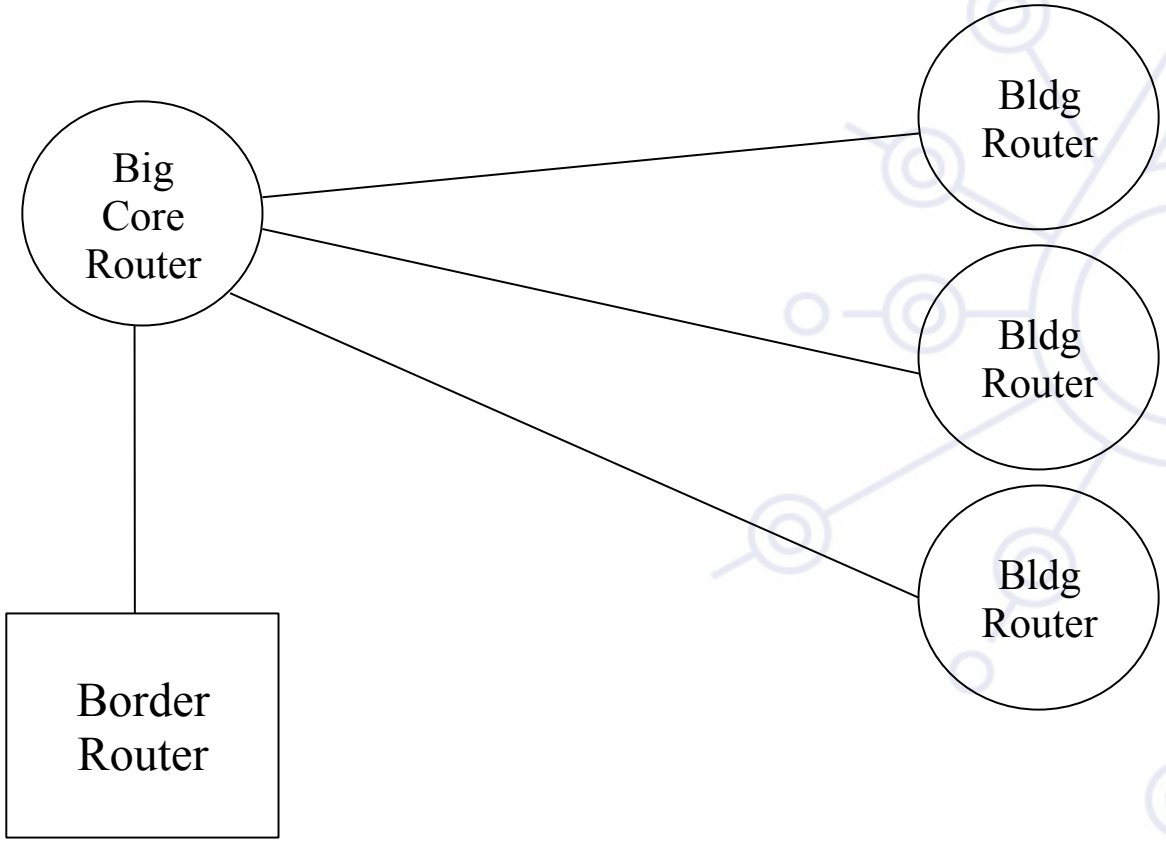
Layer-2 Campus - Redundant Switches



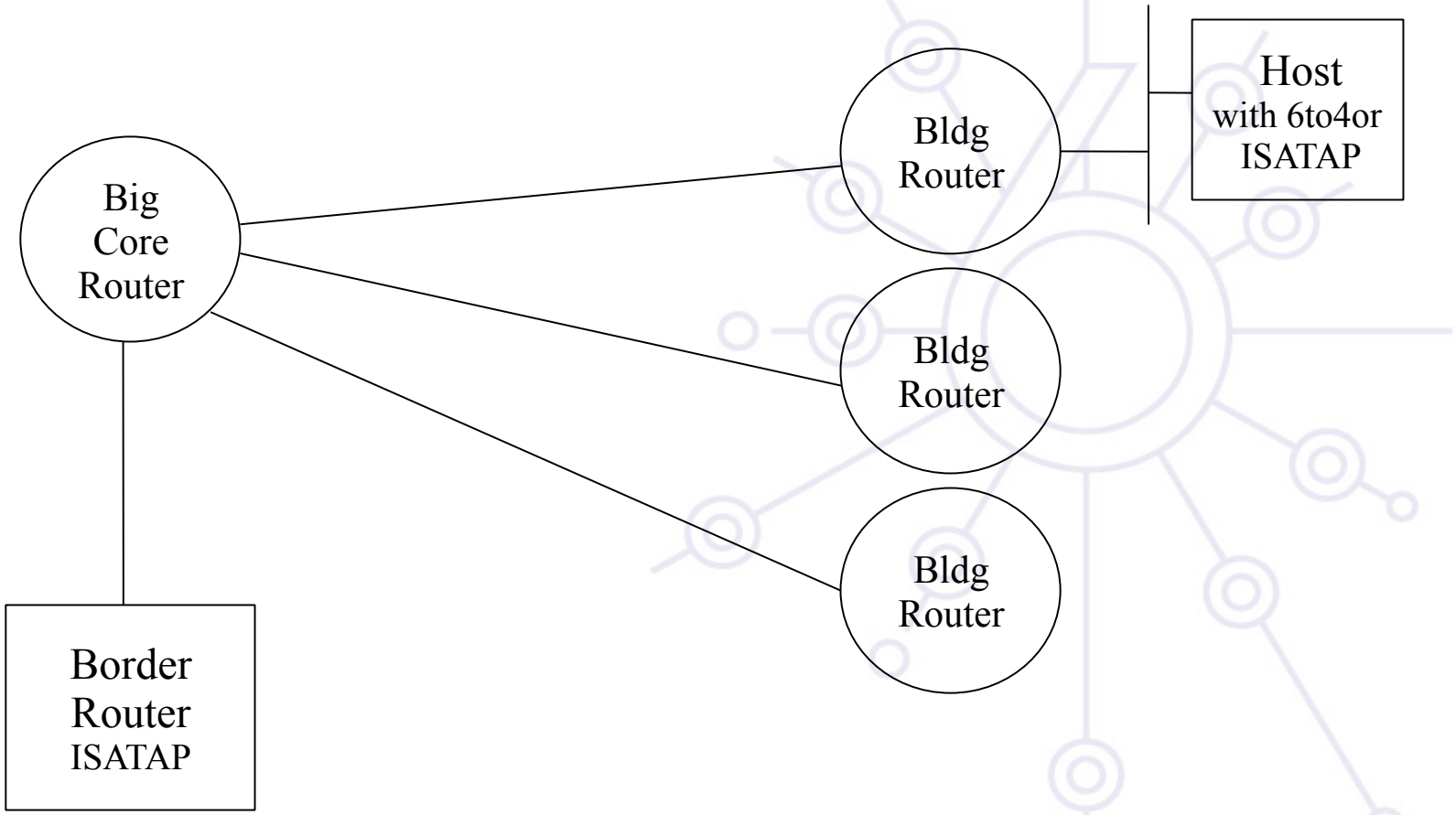
Layer-2 Campus Redundant Switches



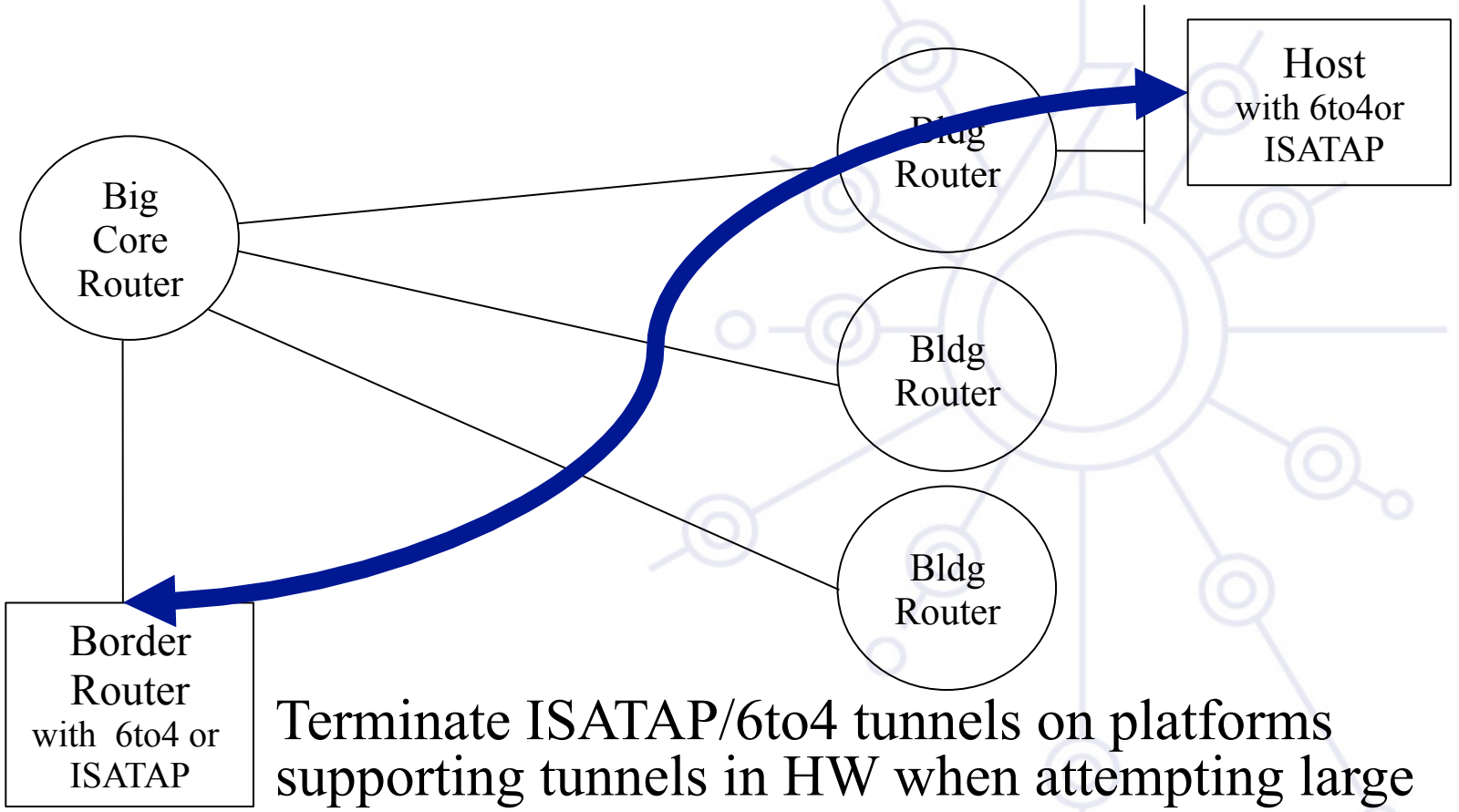
Layer-3 Campus



Layer-3 Campus

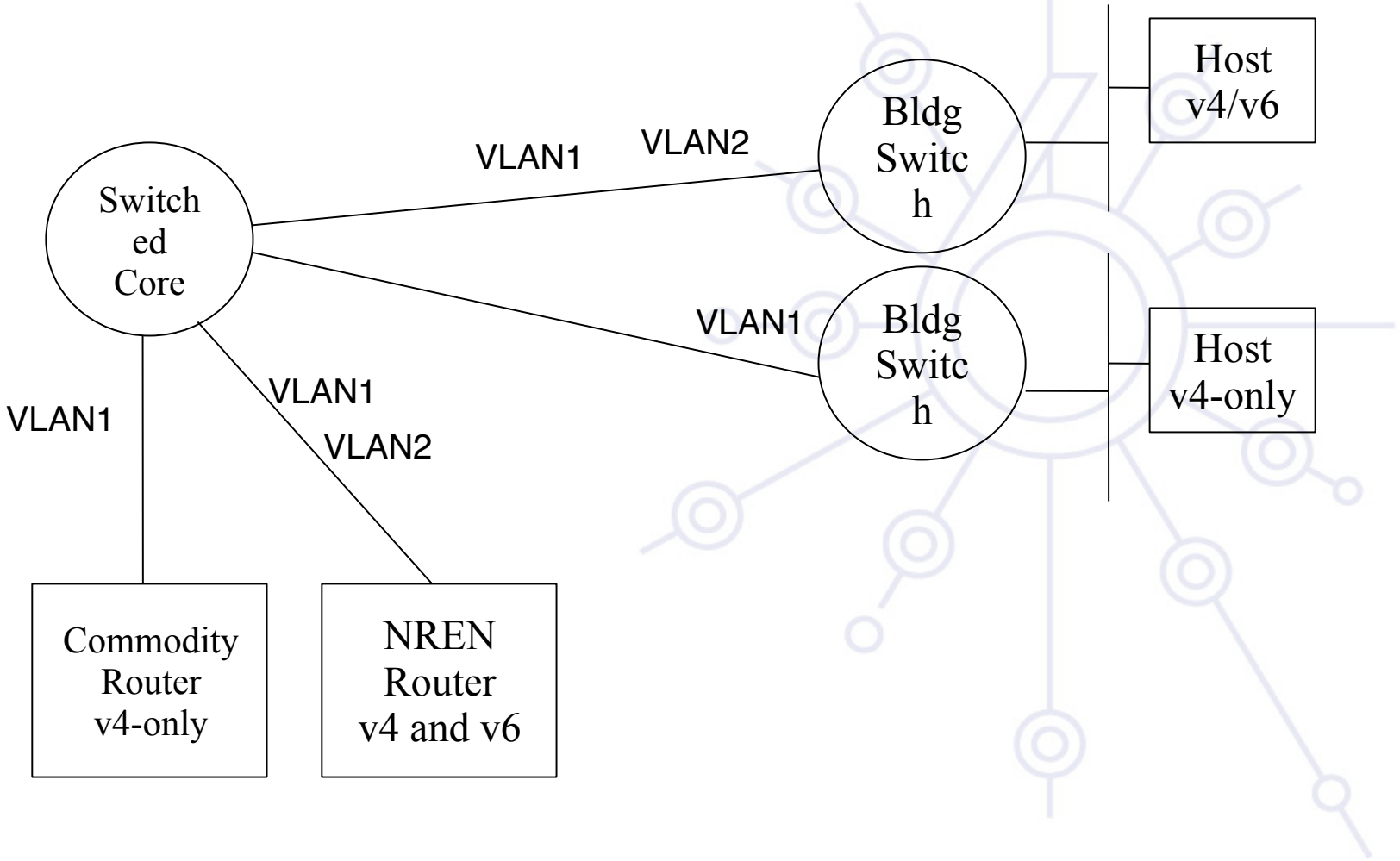


Layer-3 Campus



Terminate ISATAP/6to4 tunnels on platforms supporting tunnels in HW when attempting large scale (>100)deployments

Edge Router Options



Routing Protocols

iBGP and IGP (IS-IS/OSPFv3)

- IPv6 iBGP sessions in parallel with IPv4
- You need a 32 bit router-id for IPv6 BGP peering configuration

Static Routing

- all the obvious scaling problems, but works OK to get started, especially using a trunked v6 VLAN.

OSPFv3 might be good

- It will run in a ships-in-the-night mode relative to OSPFv2 for IPV4 - neither will know about the other.

Use the same (type) of protocol you used in IPv4.

Outline

Campus deployment strategy

Campus IPv6 address allocation

Campus deployment topology - options

Campus services

Service provider deployment considerations

Campus services –Road Map

- Name service - see DNS module
- Security policy - see security module
- Routing - see routing module
- (Mail) not considered here - see application module
- Proxying
- Remote access
- Monitoring the network and the services - see monitoring module

=> For most of these services, refer to the ad hoc modules on http://www.6deploy.org

How to enable IPv6 services ?

Add v6 testing service for different name first:

- service.v6.fqdn or service6.fqdn with AAAA + reverse PTR entry.
- Test it

Add v6 service under the same name:

- service.fqdn with A +AAAA and two PTR.

How to enable IPv6 services if you don't have an IPv6 capable server?

Use proxy (more exactly reverse-proxy) server

- Apache2.x proxy is a very good one

Use netcat

- Kind of hack ☺

Other proxies

Proxy solutions

Proxy

- Squid (http://devel.squid-cache.org/projects.html)

Web Cache

- NetCache C1300, C2300, C3300. BlueCoat SG
- WCCP does not have IPv6 support in CISCO yet

Apache2 reverse proxy

Configuration is very easy:

```
ProxyRequests Off
ProxyPass / http://ipv4address
ProxyPassReverse / http://ipv4address
ProxyPreserveHost On
```

Reverse proxy pros & cons

Advantage:

- Fast implementation, instantly provide web service over IPv6
- No modifications required in a production web server environment
- Allow for timely upgrading of systems
- Scalable mechanism: a central proxy can support many web sites

Disadvantage:

- Significant administrative overhead for large scale deployment
- May break advanced authentication and access control schemes
- Breaks statistics: all IPv6 requests seem to be coming from the same address
  - may be fixed with filtering and concatenation of logs or specialised module on proxy
- Not a long term solution overall, native IPv6 support is readily available in related applications and should be preferred whenever possible

Management and monitoring

- Device configuration and monitoring -SNMP
- Statistical monitoring e.g. Cricket/MRTG
- Service monitoring - Nagios
- Intrusion detection (IDS)
- More information
  - Module #060 : IPv6 Networks management
  - http://www.6deploy.org

Remote access via IPv6

Use native connectivity when available

- Rather easy if you are operating dial-in pool or you are an ADSL service provider
- … and even more easy if your home ISP provides IPv6 connectivity
  - Like Free and Nerim in France

Use (Open)VPN

Use tunnel broker service – rather suboptimal ?

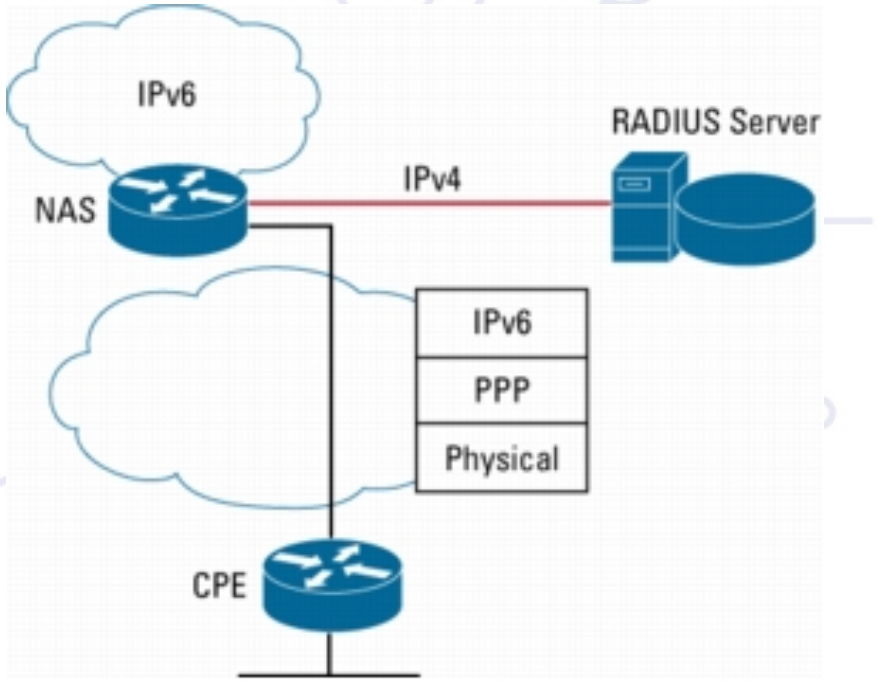
Use 6to4 if you have global IPv4 address

- Good 6to4 relay connectivity is a must

Use Teredo/softwire if you have NAT or multiple level of NATs

Remote access via IPv6 - PPP

- **The dial-up connection uses a modem and the PSTN service in order to get connection to remote devices.**
  - Most cases use PPP (Point-to-Point Protocol), which gives a standard method to transport the datagrams of several protocols over point-to-point links (RFC1661, 2153, 5342) - PPP has been updated to support the transport of IPv6 datagrams (RFC5072)



PPP and IPv6

PPP protocol has three main parts

- Definition of the encapsulation method of the IPv6 datagrams over the point- to-point link (IP6CP)
- LCP (Link Control Protocol) used to establish, configure and test the connection at link layer
- NCP (Network Control Protocol) used to establish and configure the connection at network layer

IPv6 operation:

- negotiates one link local address (fe80::/64) between the end points or peers
- Could negotiate datagram compression via IP6CP (IPv6 Control Protocol)
- PPP does not give global IPv6 addresses but link local - The global IPv6 addresses must be configured by other means
  - Manual configuration
  - Autoconfiguration (RA)
  - DHCPv6

PPP and IPv6 - implementations

Routers:

- Cisco
- Juniper

Hosts:

- Windows Vista and Microsoft Windows Server 2008
  - Windows XP: Cfos IPv6 link http://www.cfos.de/ipv6_link/ipv6_link_e.htm
- Linux, *BSD (including Mac OS X), Solaris

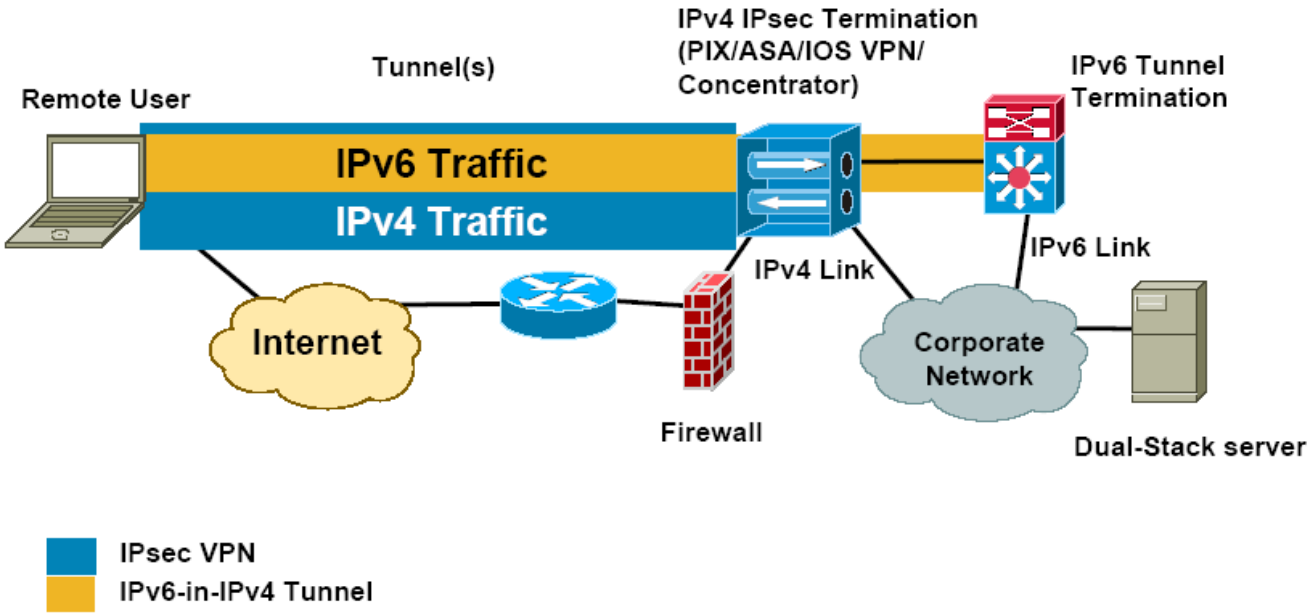
Opensource:

http://sourceforge.net/projects/pppcbcp

http://freshmeat.net/projects/pppd

Remote Access with IPSEC – or other VPNs

IPv6-in-IPv4 Tunnel Example



IPv6 load balancing

- Server clusters
  - Opensource solution: *BSD pf (http://www.openbsd.org/faq/pf/), Linux LVS after 2.6.28 (http://kb.linuxvirtualserver.org/wiki/IPv6_load_balancing)
  - Commercial platforms: Veritas Cluster Server, BigIron F5, Windows Server 2008 - Network Load Balancer
- First-Hop Redundancy:
  - HSRPv6 (Cisco only)
  - VRRPv6 - standardisation at IETF
  - NUD (Neighbor Unreachability detection)- see next slide
- Traffic loadbalancing
  - Multilink PPP - supported if multilink PPP supported
  - Equal-Cost Multi-Path routing - if IPv6 routing supported…
  - Ethernet Link Aggregations - L2 solution

Implementing default gateway redundancy

If HSRP,GLBP or VRRP for IPv6 are not available

NUD can be used for a good HA at the first-hop (today this only applies to the Campus/ Datacenters … HSRP is available on routers)

- (config-if)#ipv6 nd reachable-time 5000

Hosts use NUD "reachable time" to cycle to next known default gateway (30 seconds by default)

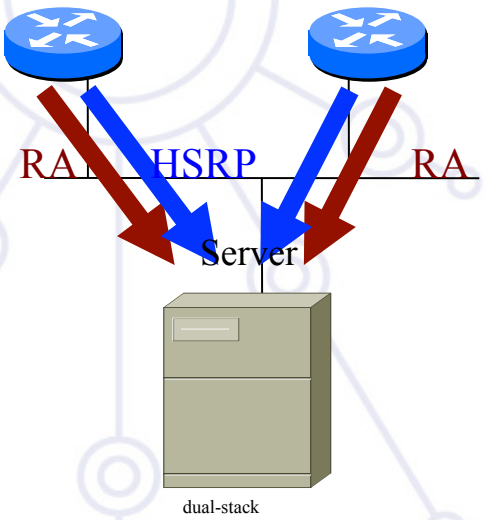
Default Gateway . . . . . : 10.121.10.1

fe80::211:bcff:fec0:d000%4

fe80::211:bcff:fec0:c800%4

Reachable Time : 6s

Base Reachable Time : 5s



Outline

Campus deployment strategy

Campus IPv6 address allocation

Campus deployment topology - options

Campus services

Service provider deployment considerations

Outline of NRENs/ISP IPv6 deployment

1. Obtain IPv6 address space
2. Plan the addressing
3. Plan the routing
4. Test in a small case
5. Deploy IPv6 (incrementally – dual-stack/6PE)
6. Enable IPv6 services

Getting IPv6 prefix for LIRs/ ISPs

Global IPv6 RIR rules

- http://www.ripe.net/ripe/docs/ipv6.html
- simple rules for LIRs
- IPv6 service should be provided
- detailed plan
- Usually /32 allocation

Establishing global rules was not easy.

- Different structure in different RIR regions: ISP, NIRs/LIRs, LIRs

What about IX? – slightly different rules

- Infrastructure addresses
- Routable /48 address

RIPE entries /1

```
whois -h whois.ripe.net 2001:0738::

inet6num:       2001:0738::/32
netname:        HU-HUNGARNET-20010717
descr:          Hungarnet IPv6 address block
                Hungarian Research & Educational Network
                Budapest, Hungary
country:        HU
mnt-by:         RIPE-NCC-HM-MNT   ←New mandatory
mnt-lower:      NIIF6-MNT         ←New mandatory
status:         ALLOCATED-BY-RIR  ←New
```

RIPE entries /2

possible values of STATUS field

- ALLOCATED-BY-RIR – Allocated address space by RIR to LIR.
- ALLOCATED-BY-LIR – Allocated address space by LIR to smaller registries/institutions
- ASSIGNED – Assigned to end-users

RPSLng is in production (at least in RIPE region)

Reverse delegation is strongly recommended

Summary

Campus deployment strategy

- Coexistence mechanism ?
- Getting an IPv6 prefix
- … and external IPv6 connectivity
- Decide a security policy for IPv6 traffic

Campus IPv6 address allocation and usage

- Work out an addressing plan
- Decide which address allocation mechanism will be used

Campus deployment topology - options

- Start IPv6 deployment
- How to remote access the campus ?

Campus services

- Enable services for IPv6
  - Starting with the DNS
- Enable management and monitoring tools
- Enable IPv6 on hosts

IPv6 a hálózati rendszergazda szemszögéből

IPv6 bevezetése Campus hálózatokban

Alapvető hálózati szolgáltatások

- DNS
- Egyéb szolgáltatások
- Hálózat felügyelet

IPv6 képes DNS software

BIND (Resolver & Server)

- http://www.isc.org/products/BIND/
- BIND 9 (kerüljük a régi verziókat)

Unix disztribúció

- Resolver Library (+ (adaptált) BIND)

NSD (authoritative server only)

- http://www.nlnetlabs.nl/nsd/

Microsoft Windows (Resolver & Server)

...

IPv6 DNS support

BIND8

- IPv6 RRs - only AAAA
- IPv4 transport (IPv6 transport with patch or since 8.4.0, resolver since 8.3.0)

BIND9

- All IPv6 RRs
- IPv4/IPv6 transport

NSD

- only authorative

PowerDNS – SQL backend

djbdns

- IPv6 RRs - only AAAA
- IPv4 transport only (IPv6 transport with patch)

Bind 9 configuration/1

named.conf entries

- More than one listen-on-v6 option can be used:

```
options {
        listen-on-v6 port   53 { any; };
        listen-on-v6 port 1234 { any; };
};
```

- In order the DNS server not to server IPv6 requests. (Before 9.2.0 – now it is the default):

```
options {
        listen-on-v6 { none; };
};
```

Bind9 configuration/2

Zone transfer:

```
transfer-source-v6 1:2:3:4:5:6:7:8;
```

Query over IPv6 enable:

```
query-source-v6 address * 53;
```

Don’t forget to update ACLs for IPv6 addresses!

IPv6 DNS és root serverek

DNS root serverek kiritikus infrastuktúra elemek

13 root – a Föld „körül” (#10 USA-ban)

Nem mind a 13 szerver IPv6 képes és érhető el IPv6-on

- http://www.root-servers.org komplett és up-to-date lista.

IPv6 a hálózati rendszergazda szemszögéből

IPv6 bevezetése Campus hálózatokban

Alapvető hálózati szolgáltatások

- DNS
- Egyéb szolgáltatások
- Hálózat felügyelet

Alkalmazások/1

Apache

- 2.0.x+ verziók támogatják az IPv6-ot automatikusan
  - --enable-v4-mapped
- Listen ::
  - Listen [::]:80
- NameVirtualHost (IPv6 cím szintén)
- Access control működik – Ne felejtsük el az ACL IPv6 címmel kiegészíteni
- WebDAV also working
- Apache 1.3.14-1.3.19- IPv6 patch elérhető

OpenSSH

- ListenAddress ::
- sshd -6 (-4)

Alkalmazások /2

Postfix

- **Postfix 2.2+ hivatalosan támogatja az IPv6-ot**
- **Postfix 2.1 - IPv6 patch és Ipv6+TLS patch elérhető: http://www.ipnet6.org/postfix/**
- **inet_interfaces = loopback-only" IP verzió független `/etc/postfix/main.cf`:**

```
inet_protocols = ipv4,ipv6,all
```

- **mynetworks [ipv6:addr:range]/plen**
- **smtp_bind_address6 forrás cím a kimenő SMTP kapcsolat esetén.**
- **lmtp_bind_address6 forrás cím a kimenő LMTP kapcsolat esetén**

Exim

- **HAVE_IPV6=YES Local/Makefile fileban**
- **dc_other_hostnames='…:host6.domain'**
- **dc_local_interfaces='ipv4address:2001::db8::ff47::1203::::5'**
- **dc_relay_nets='a.b.0.0/16:2001::db8::ff47::1203::::/64'**

Alkalmazások /3

Sendmail

- M4 konfigurációs file-ban definiiálni kell az IPv6 transportot
- DAEMON_OPTIONS('Name=MTA-v4, Family=inet')
- DAEMON_OPTIONS('Name=MTA-v6, Family=inet6')
- DBMs:
  - IPv6:2002:c0a8:51d2::23f4 REJECT
- Opció:
  - ResolverOptions=WorkAroundBrokenAAAA

Általában nincsen probléma, ha az MX-nek van IPv6 címe, de rossz MTA implementációk miatt célszerű, hogyha egy utolsó esély MX csak IPv4 címmel

- lásd RFC 3974

Alkalmazások /4

Inetd

- tcp → tcp6 vagy tcp46
- udp → udp6 vagy udp46

INN

- --enable-ipv6 a configure parancshoz

Diablo news server – IPv6-ot támogatja

FTP

- vsftpd,moftpd, pure-ftpd, tnftpd, wzdftpd, lukemftpd – supports IPv6

Alkalmazások /5

Web proxy-k

- Több web-proxy támogatja az IPv6 kapcsolatokat: wwwoffle v2.7, squid v2.5 patch-el, privoxy v3.1.1, www6to4 v1.5, Prometeo v1.4, ffproxy v1.6-RC1 és polipo v0.9.x
- Privoxy:
  - listen-address [2001:db8:ff47:1203:2::5]:8118
  - permit-access [2001:db8:ff47:1203::]/64

Alkalmazások /6

Adatbázisok

- Postgresql támogatja az IPv6-ot
  - pg_hba.conf - fájlban
    - CIDR-address – IPv6 támogatott
- MySQL terv a 5.x-ben (későbbi változatok)

Windows filesharing

- Windows 2003 server Site-Local addresses címekkel! – windows firewall letiltás (`netsh interface ipv6 set interface interface="Local*" firewall=disabled`) és IPv6 for Filesharing az Advanced settings fülben
- Windows Vista - OK
- Samba
  - patch-el: http://www.litech.org/samba/ vagy samba 3.3

Alkalmazások /7

NTP

- 4.x támogatja az IPv6-ot
- /etc/ntp.conf konfigurálás – fallback nehéz az UDP miatt
- Néhány IPv6 képes NTP szerver
  - time1.niif.hu (IPv6 and IPv4)
  - ntp.rhrk.uni-kl.de (IPv4 and IPv6)
  - ntp6.remco.org (IPv6)
  - chime3.ipv6.surfnet.nl (IPv6)
  - ntp.ipv6.viagenie.qc.ca (IPv6)

CUPS

- IPv6 támogatott 1.2b1 változat óta
  - /etc/cups/cupsd.conf fájlban:

    ```
    Listen [::1]:631
    ```

  - "/etc/cups/client.conf" fájlban:

    ```
    ServerName [2001:db8:ff47:1203::5]
    ```

Alkalmazások /8

TightVNC

- Engedélyezni kell a helyes működéshez a “Allow loopback connections” a Windows szerveren

Telnet

- Általában megszokott módon (néha -4 és -6)
- Windows 2003 Telnet szerver nem támogatja IPv6-ot még, de:

  netsh interface portproxy add v6tov4 23

Alkalmazások /9

OpenLDAP

- IPv6 támogatott az LDAP szerveren és kliensen is
  - Egyéb LDAP-ot használó alkalmazások is IPv6 képesek lesznek ha az OpenLDAP client library-t használják
- Sun ONE Directory szerver támogatja az IPv6-ot
- Fedora DS 1.0.3 szerver támogatja az IPv6-ot

GnomeMeeting/Ekiga + Polycom HDX

- H.323 VoIP és videokonferencia. IPv6 és *x támogatás. http://www.gnomemeeting.org/

Kphone

- IPv6 VoIP SIP alapú softphone

http://www.iptel.org/products/kphone/

Néhány programozási nyelv

Perl

- Speciális modulok mint Socket6 és IO::Socket::INET6

Python 2.3.4 és későbbi működik IPv6-al

- Habár, Windows binárisok a python.org-on nem támogatják. 2.4 binárisok IPv6 támogatással lesznek terjesztve.

PHP

- Részleges IPv6 támogatás
- Sok PHP szkript müködik IPv6-on mindenféle változtatás nélkül

Java

- SUN Java SDK 1.4 és később IPv6 támogatás
- A legtöbb Java alkalmazás müködik IPv6-al, mert a Java API magasabb szinten kezeli a dolgokat

További alkalmazások

Nagy lista az IPv6 képes alklamazásokról

http://www.deepspace6.net/docs/ipv6_status_page_apps.html

IPv6 Application and Patch Database

- kereshető

http://ipv6.niif.hu/ipv6_apps/

- konfigurációs leírások

http://ipv6.niif.hu/faq/

6NET alkalmazások

http://apps.6net.org/WP5Apps/Applications.html

IPv6 a hálózati rendszergazda szemszögéből

IPv6 bevezetése Campus hálózatokban

Alapvető hálózati szolgáltatások

- DNS
- Egyéb szolgáltatások
- Hálózat felügyelet

Management and monitoring

Device configuration and monitoring -SNMP
Statistical monitoring e.g. Cricket/MRTGv6
Service monitoring - Nagios
Intrusion detection (IDS) – Netflow
Services for others – Looking glass
Authentication systems

- For example, 802.1x + RADIUS for WLAN

Cricket

Cricket is a tool for storing and viewing time-series data.

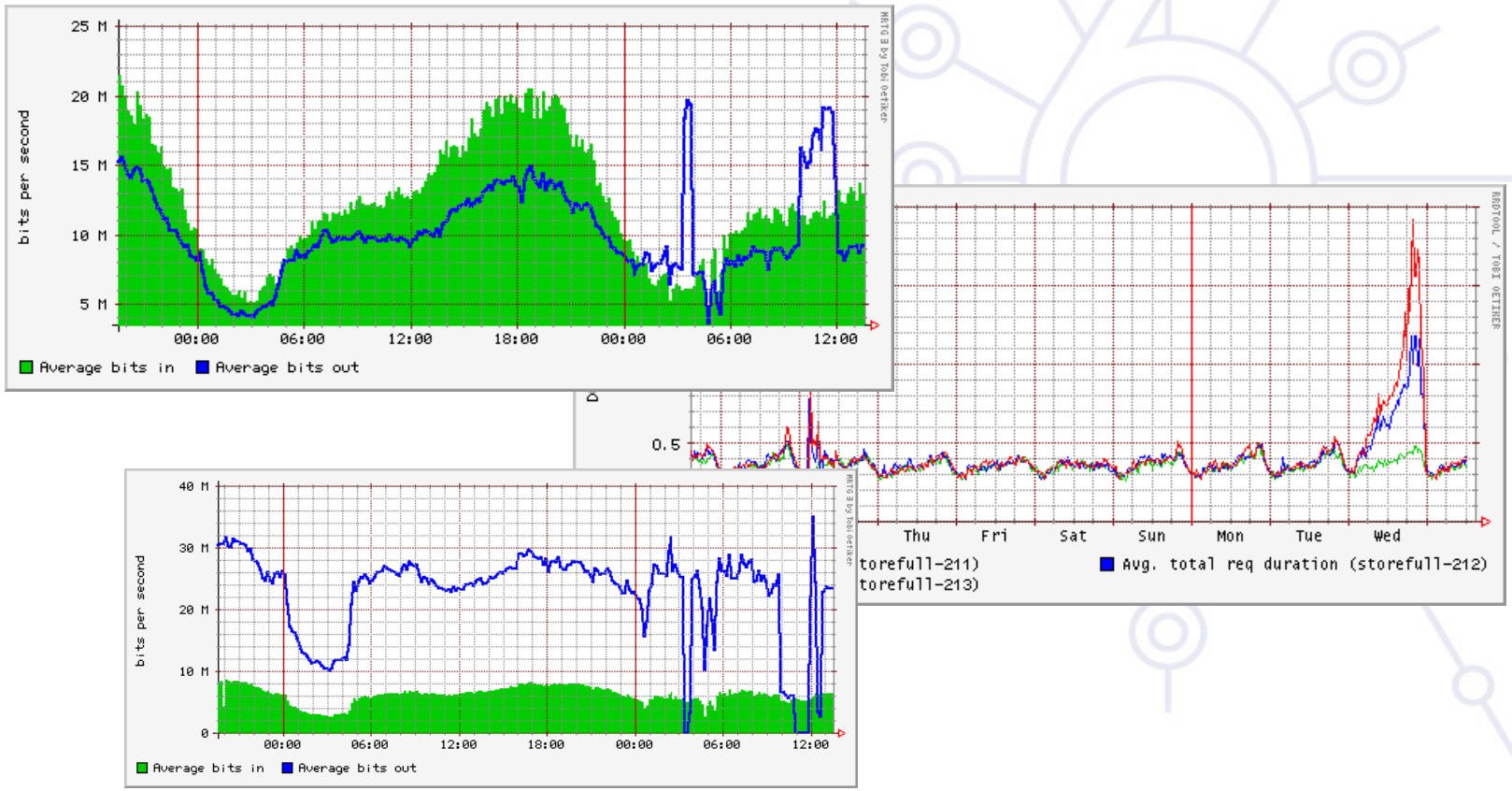
Very flexible

Extremely Legible Graphs

Space and Time efficient

Platform Independent

Example Graphs



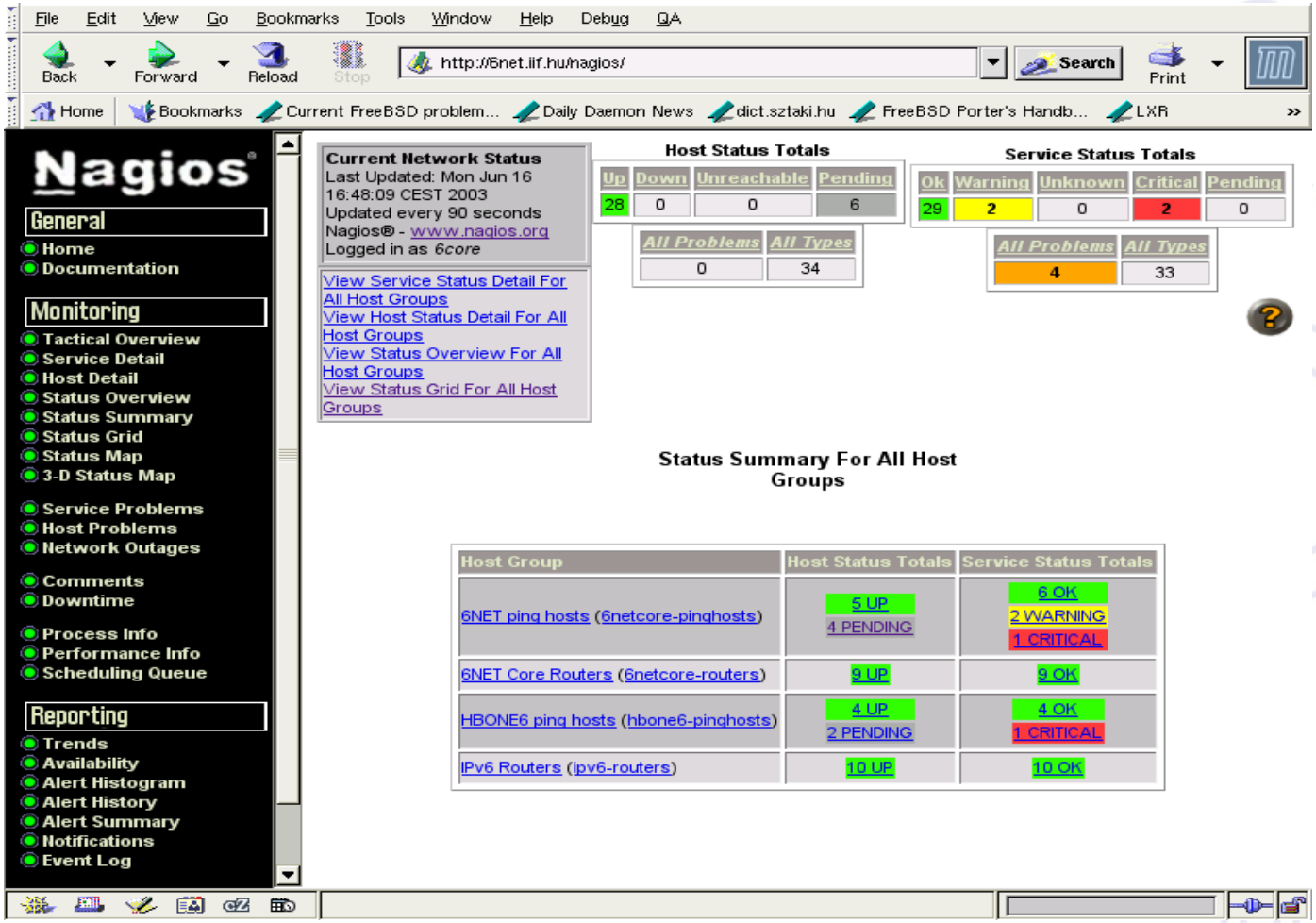
Cricket and IPv6

No separate SNMP MIBs for IPv6 traffic implemented yet

- Separate IPv6 infrastructure – easy to monitor
- Dual-stack infrastructure – no easy way to monitor
  - firewall filter and counters – hardly possible on Cisco
  - From CLI: show interface accounting – misleading implementations – only process switched packets on GSR+E3 cards it is correct

Nagios: Overview

- Web-based monitoring system
- Allows for monitoring of virtually any service the NOC provides
- Provides alerting and acknowledgment capabilities
- Provides logging of downtimes and reporting capabilities


File Edit View Go Bookmarks Tools Window Help Debug QA
Back Forward Reload Stop
http://6net.iif.hu/nagios/
Search Print
Home Bookmarks Current FreeBSD problem... Daily Daemon News dict.sztaki.hu FreeBSD Porter's Handb... LXR
Nagios
General
Home
Documentation
Monitoring
Tactical Overview
Service Detail
Host Detail
Status Overview
Status Summary
Status Grid
Status Map
3-D Status Map
Service Problems
Host Problems
Network Outages
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Reporting
Trends
Availability
Alert Histogram
Alert History
Alert Summary
Notifications
Event Log
Current Network Status
Last Updated: Mon Jun 16 16:48:09 CEST 2003
Updated every 90 seconds
Nagios® - www.nagios.org
Logged in as 6core
View Service Status Detail For All Host Groups
View Host Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Grid For All Host Groups
Host Status Totals
Up Down Unreachable Pending
28 0 0 6
All Problems All Types
0 34
Service Status Totals
Ok Warning Unknown Critical Pending
29 2 0 2 0
All Problems All Types
4 33
Status Summary For All Host Groups
Host Group
Host Status Totals
Service Status Totals
6NET ping hosts (6netcore-pinghosts)
5 UP
4 PENDING
6 OK
2 WARNING
1 CRITICAL
6NET Core Routers (6netcore-routers)
9 UP
9 OK
HBONE6 ping hosts (hbone6-pinghosts)
4 UP
2 PENDING
4 OK
1 CRITICAL
IPv6 Routers (ipv6-routers)
10 UP
10 OK

IPv6 status

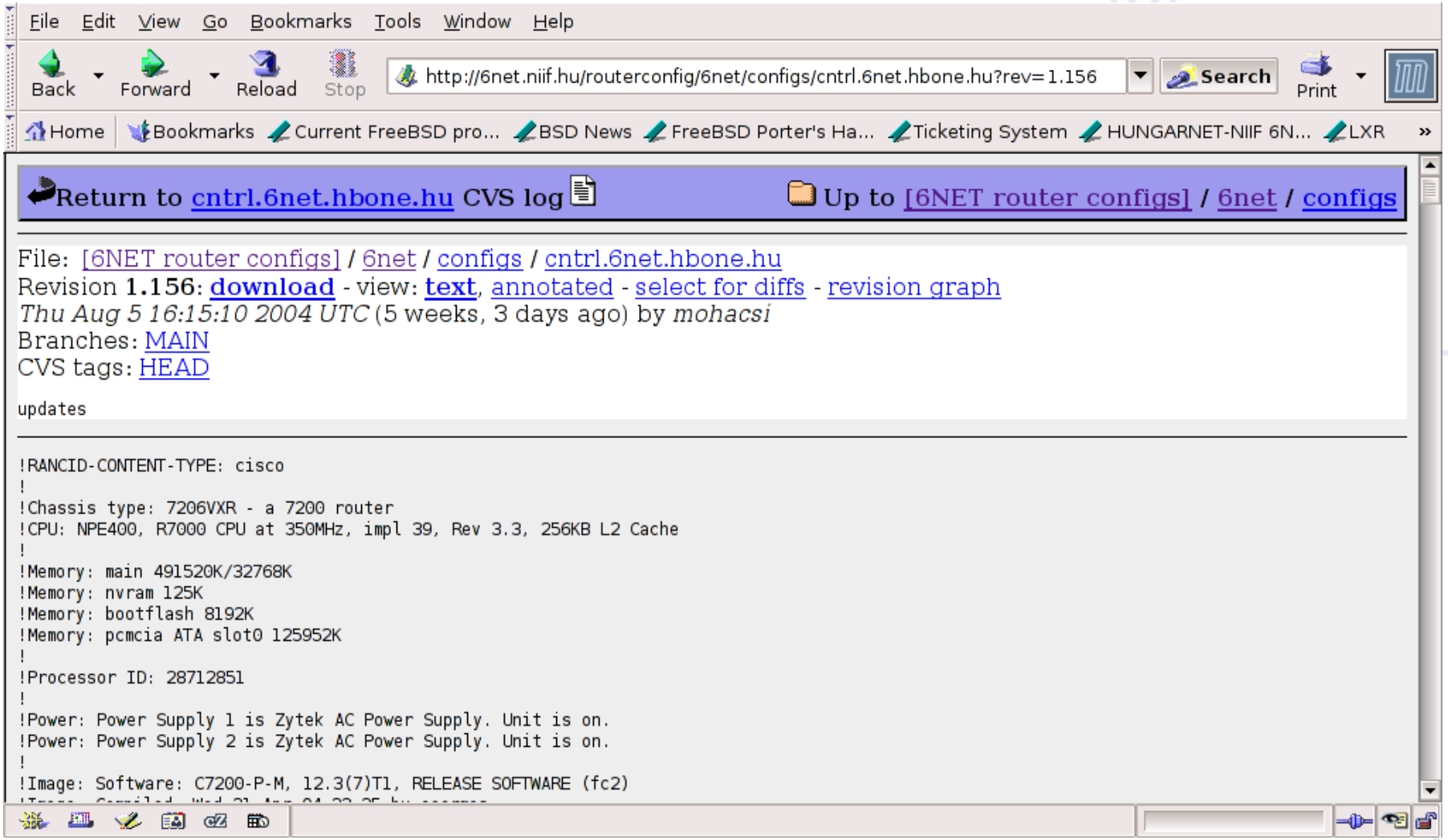
Monitoring

• Ping over IPv6	OK – with plugin
• TCP services over IPv6	OK – with plugin
• UDP services over IPv6	OK – with plugin
• SNMP over IPv6	Not yet - working on it

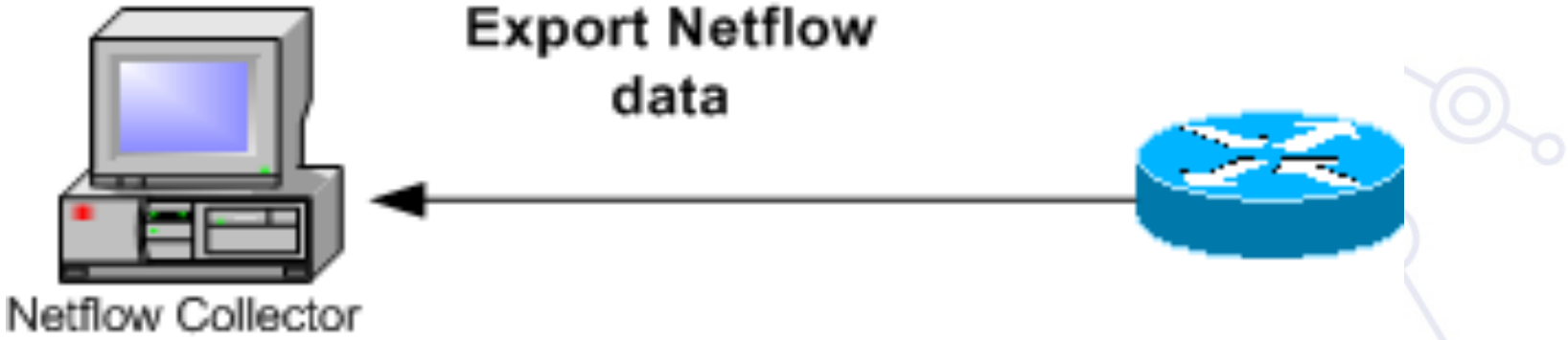
RANCID:
Really Awesome New Cisco ConfIg Differ

- Web-based CVS repository of configuration changes
- Unix cron jobs at regular intervals check configured routers for configuration changes
- If a change is detected, RANCID e-mails all the engineers with the changes and updates the CVS repository
- Web-based CVS repository allows engineers to choose arbitrary dates to view configuration changes
- Can alter scripts to grab any information from the router that you want to track

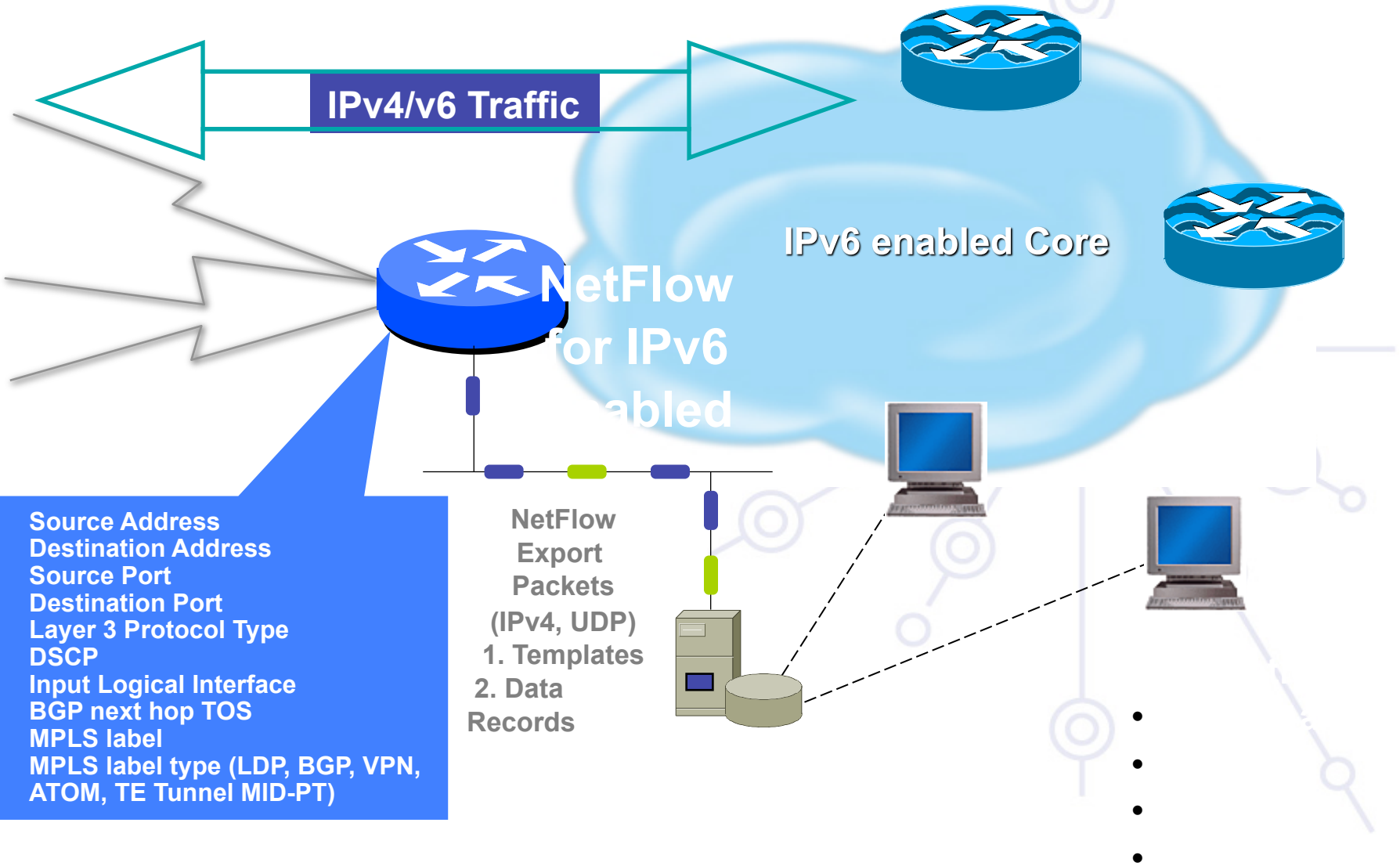
Output of Rancid



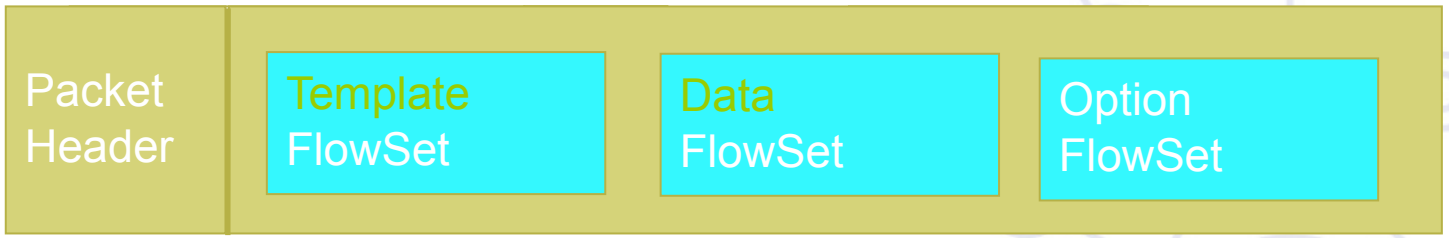
Netflow



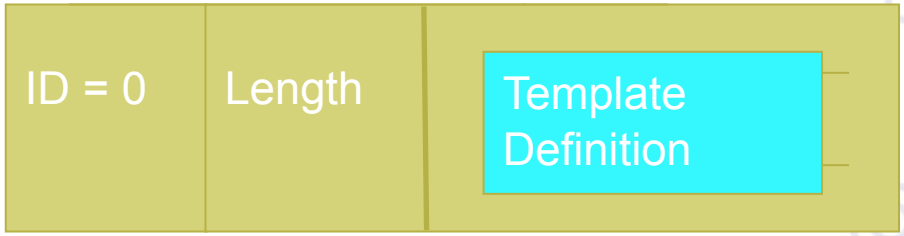
NetFlow for IPv6



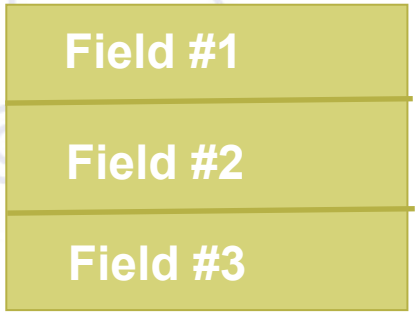
NetFlow Version 9



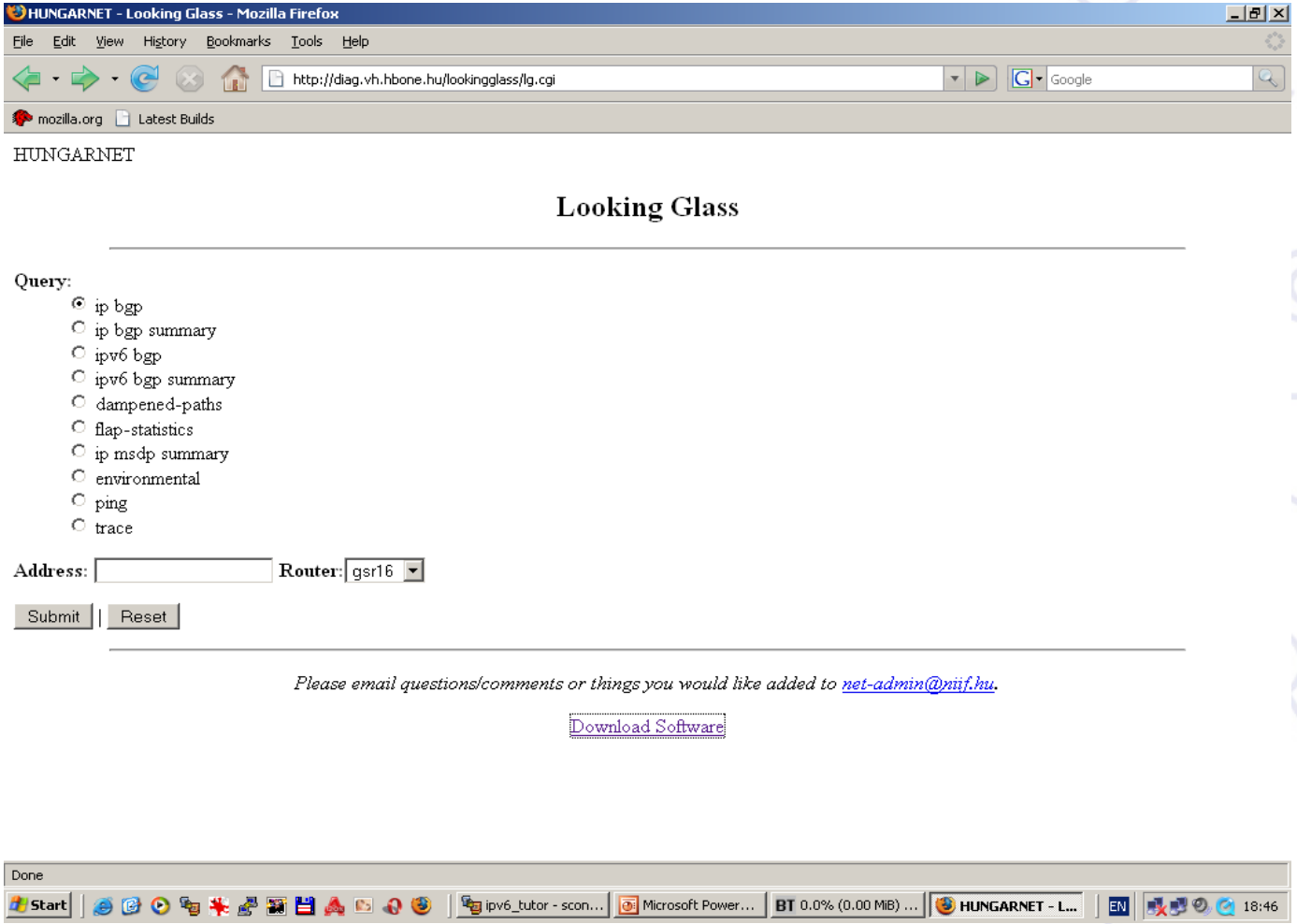
Template



Data



Looking Glass




deploy
IPv6 Security

What is new with IPv6?

Security was considered from the start in IPv6

Some of the key improvements:

- IPsec useable with the core protocols
- Cryptographically Generated Addresses (CGA)
- SEcure Neighbor discovery (SEND)
- Making scanning/intrusion harder

Scanning Gateways and Hosts

Subnet Size is much larger

- About 500,000 years to scan a /64 subnet@1M addresses/sec

But…

- NMAP does support IPv6 network scanning
- IPv6 Scanning methods are changing
  - DNS based, parallelised scanning, common numbering
- Compromising a router at key transit points
  - Can discover addresses in use

Security of IPv6 addresses

Cryptographically Generated Addresses (CGA) IPv6 addresses [RFC3972]

- Host-ID part of address is an encoded hash
  - Binds IPv6 address to public key
- Used for securing Neighbor Discovery [RFC3971]
- Is being extended for other uses [RFC4581]

Private addresses as defined [RFC 3041]

- prevents device/user tracking from
- makes accountability harder

Host-ID could be token to access network

Autoconfiguration/Neighbor Discovery

Neigbor Discovery

- Can suffer similar problems of ARP cache poisoning

Better solution with SEcure Neighbor Discovery (SEND) [RFC3971]

- Uses CGA
  - Linux implementation: DoCoMo's Open Source SEND Project

DHCPv6 with authentication is possible

ND with IPSec also possible

Preventing rogue RA

- RAGuard – draft at IETF

Unauthorised Access Control

Policy implementation in IPv6 with Layer 3 and Layer 4 is still done in firewalls

Some design considerations!

- Filter site-scoped multicast addresses at site boundaries
- Filter IPv4 mapped IPv6 addresses on the wire

Action	Src	Dst	Src port	Dst port
permit	a:b:c:d::e	x:y:z:w::v	any	ssh
deny	any	any		

Unauthorised Access control

Non-routable + bogon (unallocated) address filtering slightly different

- in IPv4 easier deny non-routable + bogons
- in IPv6 simpler to permit legitimate (almost)

Action	Src	Dst	Src port	Dst port
deny	2001:db8::/32	host/net		
permit	2001::/16	host/net	any	service
permit	2002::/16	host/net	any	service
permit	2003::/16	host/net	any	service
Deny	3ffe::/16	host/net	any	service
deny	any	any		

Amplification (DDoS) Attacks

There are no broadcast addresses in IPv6

- This would stop any type of amplification attacks that send ICMP packets to the broadcast address
- Global multicast addresses for special groups of devices, e.g. link-local addresses, etc.

IPv6 specifications forbid the generation of ICMPv6 packets in response to messages to global multicast addresses

- Many popular operating systems follow the specification
- Still uncertain on the danger of ICMP packets with global multicast source addresses

Mitigation of IPv6 amplification

Be sure that your host implementations follow the ICMPv6 spec [RFC 4443]

Implement Ingress Filtering

- Defeats Denial of Service Attacks which employ IP Source Address Spoofing [RFC 2827]

Implement ingress filtering of IPv6 packets with IPv6 multicast source address

Firewalls

IPv6 architecture and firewall - requirements

- No need to NAT – same level of security with IPv6 possible as with IPv4 (security and privacy)
  - Even better: e2e security with IPSec
- Weaknesses of the packet filtering cannot be hidden by NAT
- IPv6 does not require end-to-end connectivity, but provides end-to-end addressability
- Support for IPv4/IPv6 transition and coexistence
- Not breaking IPv4 security

There are IPv6-capable firewalls now

- Tested and used: Cisco ACL/PIX, iptables, ipfw, pf, Juniper NetScreen

Tűzfal követelmények

Nem lehet vakon kiszűrni ICMPv6-t:

Echo request/reply	Debug
No route to destination	Debug – jobb hiba indikáció mint ICMPv4 esetén
TTL exceeded	Hiba jelentés
Parameter problem	Hiba jelentés
NS/NA	Szükséges a helyes müködéshez – kivéve statikus ND bejegyzések esetén
RS/RA	Stateless Address Autoconfigration esetén szükséges
Packet too big	Path MTU discovery
MLD	Requirements in for multicast in architecture 1

IPv6 specifikus

szükséges

Tűzfal követelmények 2

Nem lehet vakon kiszűrni az IP opciókat (→ extension Header):

Hop-by-hop header	Mit kell tenni jumbogram-okkal és router alert opcióval? – multicast join üzenetekhez szükséges...
Routing header	Source routing – IPv4 esetén kártékonynak minősített, de szükséges IPv6 mobilitáshoz – csak a Home Agent-en szükséges engedélyezni a Type 2 típusú RH-t
ESP header	Biztonsági policy szerinti feldolgozás
AH header	Biztonsági policy szerinti feldolgozás
Fragment header	Minden fregmens kivéve az utolsót 1280 octetnél hoszabb kell, hogy legyen

IPv6 tűzfalak alkalmazás támogatása

FTP:

- Elég komplex: PORT, LPRT, EPRT, PSV, EPSV, LPSV (RFC 1639, RFC 2428)
- IPv6 tűzfalakban alig van támogatás
- HTTP tűnik a következő generációs fájltranszfer protokollnak különösen WEBDAV és DELTA kiegészítéssel

Egyéb nem triviálisan proxyzható protokoll pl. H.323:

- Nincs támogatás

Equipment Configuration: Hosts

6DEPLOY. IPv6 Deployment and Support

IPv6 Support – Hosts Operating Systems

Vendor	Versions supporting IPv6	More Information
Apple	MAC OS X 10.2	http://developer.apple.com/macosx/
BSD	FreeBSD 4.0 OpenBSD 2.7, NetBSD 1.5 BSD/OS 4.2	http://www.kame.net/
HP / Compaq	HP-UX 11i, Tru64 UNIX V5.1, OpenVMS V5.1	http://docs.hp.com/en/5990-7247/index.html
IBM	z/OS Rel. 1.4, AIX 4.3, OS/390 V2R6 eNCS	http://www-01.ibm.com/software/info/ipv6/compliance.jsp
Linux	Red Hat 6.2, Mandrake 8.0, SuSE 7.1, Debian 2.2	http://www.bieringer.de/linux/IPv6/status/IPv6+Linux-status-distributions.html
Microsoft	Windows Vista, XP, Server 2003, Server 2008, CE .NET, Mobile	http://www.microsoft.com/ipv6/
Novell	Netware 6.1	http://www.novell.com/documentation/oes2/ntwk_ipv6_nw/index.html?page=/documentation/oes2/ntwk_ipv6_nw/data/ai4x21f.html
Sun	Solaris 8, 9 and 10	http://docs.sun.com/app/docs/doc/817-0573?l=en

http://www.ipv6tf.org/index.php?page=guide/organizations/vendors/oss

Host Equipment

Windows

BSD

Linux

Solaris

Mac OS X

IPv6 Windows-on

Elérhető:

- NT4-hez béta (Microsoft Research)
- Win2000-hez technology preview (nem kompatibilis SP-kel)
- WindowsXP -installálható
- Windows Vista - default bekapcsolva
- Windows .Net Server 2003 - installálható

Támogatott:

- autoconfiguration, IPv4 tunnel, 6to4 tunnel, 6to4 relay, ISATAP tunnel, IPSec (kézi kulcs csere)

IPv6 in Windows Vista

IPv6 is enabled by default

It not only supports the basic functionalities as in previous versions (i.e. Windows XP and 2003) but also new advanced features such as

- Dual IP layer architecture Installed and enabled by default
- Graphical user interface (GUI)-based configuration
- Full Support for IPsec
- MLDv2
- DNS messages over IPv6
- LLMNR (Link Local Multicast Name Resolution)
- Literal IPv6 addresses in URLs
- Support for ipv6-literal.net names
- IPv6 over PPP
- DHCPv6

Windows Vista müködés

IPv6 be van kapcsolva - default!

IPv6 preferált IPv4-el szemben

- Vista IPv6 NA/NS/RS üzeneteket küld ha link-upba megy
  - DHCPv6-al probálkozik
  - Ha nincsen RA-ból tanult (globális vagy ULA) címet használ
  - Egyébként ISATAP
  - Egyébként Teredo
  - Egyébként IPv4 – Utolsó esély

Amelyik alkalmazás Peer-to-Peer Framework-ot használja az megköveteli az IPv6-ot, nem müködik IPv4 –el:
http://www.microsoft.com/technet/network/p2p/default.mspx

Windows Vista configuration (1)

- **Automatic address configuration**
  1. Stateless address autoconfiguration with IPv6 RA
  2. Stateful address autoconfiguration with DHCPv6
- **Manual address configuration**
  1. The GUI of the properties of TCP/IPv6 component
  2. Commands in the netsh interface ipv6 context

```
netsh interface ipv6 add address interface_name
ipv6_address
```

- **Address selection configuration**
  - RFC3484 provides a standardized method to choose source and destination IPv6 addresses with which to attempt connections
  1. A destination address selection algorithm to sort the list of possible destination addresses in order of preference
  2. A source address selection algorithm to choose the best source address to use with a destination address

Windows Vista configuration (2)

Unlike XP, IPv6 in Vista cannot be uninstalled

To disable IPv6 on a specific connection

- Network Connections folder > properties of the connection > clear the check box next to the TCP/IPv6 component
- This method disables IPv6 on your LAN interfaces and connections
- But does not disable IPv6 on tunnel interfaces or the IPv6 loopback interface

To selectively disable IPv6 components and configure behaviors

- Create and configure the following registry value (DWORD type)

  `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tcpip6\Parameters\DisabledComponents`

  `DisabledComponents` is set to 0 by default

IPv6 in Windows XP

Not installed by default, and installation varies on service packs

SP1 additions:

- vendor support
- GUI installation
- configuration via netsh command

SP2 additions

- Teredo client
- host-specific relay support
- IPv6 firewall

IPv6 installation in Windows XP

No service packs

- type ipv6 install from the command prompt

SP1

- install protocol "Microsoft IPv6 Developer Edition" from Connection Properties window

SP2

- install protocol "Microsoft TCP/IP version 6" from Connection Properties window

Windows XP configuration (1)

Command for IPv6 configuration

- ipv6 (will be discontinued, not present in Windows Server 2003)
- netsh interface ipv6

Autoconfiguration is working

- netsh interface ipv6 4
- interface 1 - loopback
- interface 2 - ISATAP
- interface 3 - 6to4 interface
- interface 4... – real network interfaces
- interface 5 – Teredo interface

Windows XP configuration (2)

Set manual address

- `netsh ipv6 interface {add|set} address [interface=] <interface> [address=] <address>`
- `<interface>` - interface name or index
- `<address>` - address in IPv6 format

Deleting manual address

- `netsh ipv6 interface delete address [interface=] <interface> [address=] <address>`

Windows XP configuration (3)

Set/remove static IPv6 route

- `netsh ipv6 interface {add|set|delete} route [prefix=]<prefix>/<length> [interface=] <interface> [[nexthop=] <address>]`

Applications

- ipconfig, netstat, ping6, tracert6, pathping
- All Wininet.dll based applications
  - ftp, telnet, IExplorer,

Windows 2003 server

- `netsh interface ipv6 (only!)`
- file/print sharing-et (site-local) supported over IPv6
- IIS and media server

Windows XP configuration (4)

Neighbor cache

- `netsh interface ipv6 show neighbors (ipv6 nc)`

IPv6 routing table

- `netsh interface ipv6 show routes (ipv6 rt)`

Reconfiguration

- `netsh interface ipv6 renew (ipv6 renew)`

Address selection policy

- `netsh interface ipv6 show prefixpolicy`
- `netsh interface ipv6 set prefixpolicy [prefix=]<prefix>/<length> [precedence=] precedence [label=]label`

What Windows cannot do with IPv6

DNS messages over IPv6

- not for Windows XP, but Windows Vista and Server 2003 can, there is a builtin proxy for it

IPv6 support for file and print sharing

- Windows 2003 can

IPv6 support for the WinInet, IPHelper, and DCOM APIs

Windows XP configuration (4)

IPSec

- `ipsec6 sp/sa/s/l`
- No ESP support by default

.NET

- IPv6 support, but IPv6 literal address does not work

IPv6 firewall support after SP2 or Advanced networking pack

IPv6 teredo support after SP2 or Advanced networking pack

Further information: http://www.microsoft.com/ipv6/

Important! You should switch on IPv6 support if you have IPv6 connectivity or you have to tweak RFC3484 knobs

Windows XP configuration (5)

Windows XP ICF – same rules for IPv4 and IPv6

- Show configuration:
  - `netsh firewall show globalport`
  - `netsh firewall show adapter`
- Set configuration
  - `set globalport [port#=enable|disable] [name=name] [protocol=tcp|udp]`
  - `set adapter [name] [icmp type#=enable|disable] [port port#=enable|disable [name=name] [protocol=tcp|udp]] [ignoreglobalport port#=enable|disable] [name=name] [protocol=tcp|udp]] [filtering=enable|disable]`
  - `set logging [filelocation=<location>] [filesize=integer] [droppedpackets=enable|disable] [successfulconnections=enable|disable]`

After SP2

- in the firewall you can configure Path MTU discovery support
- per process configuration possible

Further information:

http://www.microsoft.com/technet/community/columns/cableguy/cg0104.mspx

Windows XP/.Net/Vista configuration (netsh)

Configure an IPv6 in IPv4 tunnel

- `netsh interface ipv6 add v6v4tunnel Name [Your IPv4 Endpoint] [Server IPv4 Endpoint]`
- `netsh interface ipv6 add address Name [Your IPv6 Endpoint]`

Configure a default route

- `netsh interface ipv6 add route 0::/0 Name publish=yes`

Configure a static route

- `netsh interface ipv6 add route [Tunnel Prefix]/ [Prefix Length] Name`

Allow ICMP ping

- Windows XP SP1 and lower
  - `netsh firewall set adapter Name icmp all=enable`
- Windows XP SP2 and up, 2003 and Vista
  - `netsh firewall set icmpsetting Name enable all`

Reminder about RFC3484

(Default Address Selection for IPv6)

Multiple source addresses: - linklocal, global, tunneling, mobile, choosing IPv6 or IPv4 for communication – which one to select?

- implement sorting in getaddrinfo()- via policy table:

prefer native IPv6

Prefix	Precendence	Label
::1/128	50	0
::/0	40	1
2002::/16	30	2
::/96	20	3
::ffff:0:0:/96	10	4

prefer IPv4

Prefix	Precendence	Label
::1/128	50	0
::/0	40	1
2002::/16	30	2
::/96	20	3
::ffff:0:0:/96	100	4

IPv6 *BSD-n

Támogatott:

- autokonfiguráció, IPv4 tunnel, 6to4, MLDv1, IPSec, Jumbogram, ICMP mode information query, TRT, privacy extension

Elérhető: FreeBSD 4.0, OpenBSD 2.7, NetBSD 1.5 óta

KAME:

- NAT-PT, DHCPv6, PIM-(S)SM, multicast DNS, EDNS resolver, ISATAP, anycast

FreeBSD konfiguráció /1

Installálás: nem kell, a default kernel tartalmazza

Az installer felajánlja az ipv6-ot:

- `ipv6_enable="yes"`
- Autkonfiguráció működőképes

```
ifconfig -a
```

FreeBSD konfiguráció /2

Manuális cím konfiguráció

- `ipv6_prefix_fxp0="fec0:80::"`
- `ipv6_ifconfig_fxp0="fec0:80::1 prefixlen 64"`
- majd /`etc/netstart`
- vagy `ifconfig`

Neighbor cache:

- `ndp -a`

routing tábla:

- `route/netstat`

FreeBSD konfiguráció /3

További címek konfigurálása

- `ipv6_ifconfig_if0_alias0="fec0:0:0:5::2/64"`

Mi van ha nincs még sem IPv6 konnektivitás

- ip6addrctl(8) program - RFC3484 szerint állítható a cím választási szabály:

```
#preferip4connection_policy
#Prefix        Precedence Label
::1/128               50     0
::/0                  40     1
2002::/16             30     2
::/96                 20     3
::ffff:0:0/96        100     4
```

FreeBSD konfiguráció /3

Újra konfigurálás

- `rtsol fxp0`

Alkalmazások:

- ping6, traceroute6, ftp, telnet, r* parancsok, sendmail, apache, Mozilla, proftpd, OpenSSH, LPD, NFS/YP (FreeBSD 5.0 tól), courier-imap ,irc, openldap, tftp, tcpdump, inn, tin

További információk

- `http://www.freebsd.org`,
- `http://ipv6.niif.hu/m/FAQ`,
- `http://www.kame.net`

FreeBSD configuration/4

Configure an IPv6 in IPv4 tunnel

- ifconfig gif1 create
- ifconfig gif1 tunnel @IPv4_source @IPv4_dest
- ifconfig gif1 inet6 @IPv6_address up

Configure an IPv6 in IPv6 tunnel

- ifconfig gif1 create
- ifconfig gif1 tunnel @IPv6_source @IPv6_dest
- ifconfig gif1 inet6 @IPv6_address up

FreeBSD configuration /5

Configure a static route

- Default route

```
route add -inet6 default fe80::X:X:X:X%interface
route add -inet6 default X:X:X:X::X (if global address)
```

- Others

```
route add -inet6 X:X:X:X:: -prefixlen YY X:X:X:X::X
route add -inet6 X:X:X:X:: -prefixlen YY fe80::X:X:X:X%interface
```

%*interface* notation

If link-local address, need to specify on which interface the address is available

FreeBSD configuration /6

Router advertisement: /etc/rtadvdv.conf

```
default:\
        :chlim#64:raflags#0:rltime#1800:rtime#0:retrans#0:\
        :pinfoflags="la":vltime#2592000:pltime#604800:mtu#auto:
```

- ```
  ef0:\              :addr="2001:DB8:ffff:
  1000::":prefixlen#64:tc=default:
  ```
  ```

IPv6 Linux-on

Támogatott:

- autokonfiguráció, IPv4 tunnel, 6to4
- Kernel 2.2.x óta javasolt 2.4.8 legalább

USAGI patch (nagyjából bekerült 2.6.x)

- Node information query, anycast, ISATAP, privacy extension, IPSec, applications, bug-fix, mobile IP

Általános Linux konfiguráció/1

Kernel fordítási opciók:

- CONFIG_IPv6=m/y

Autokonfiguráció müködik

`ifconfig`

Általános Linux konfiguráció/2

Cím konfiguráció

- `ifconfig <interface> inet6 add <ipv6address>/<prefixlength>`

Neighbor cache:

- `ip -6 neigh show`

IPv6 routing tábla:

- `route -A inet6`

Redhat konfiguráció/1

`/etc/sysconfig/network` file:

- # Globális IPv6 támogatás bekapcsolása
- `NETWORKING_IPV6="yes"`

`/etc/sysconfig/network-scripts/ifcfg-eth0` file:

- # IPv6 támogatás bekapcsolása ezen az interface-n
- `IPV6INIT="yes"`
- # Az interface IPv6-os címének kézi konfigurálása
- `IPV6ADDR="3FFE:2F00:20::291D:6A83/48"`

`/etc/sysconfig/static-routes-ipv6` file:

- # Default route beállítások:
- `eth0 ::/0 3FFE:2F00:20::922:A678`

Redhat konfiguráció/2

Alkalmazások:

- ping6, traceroute6, tcpdump, tracepath6, apache, bind, imap (xinetd), sendmail, openssh, telnet, ftp, mozilla, lynx, wget, kde, xchat,

További információk:

- `http://www.bieringer.de/linux/IPv6/, http://ipv6.niif.hu/tipster6/, http://www.hs247.com, http://www.linux-ipv6.org/`

Debián konfiguráció/1

Main URL:

http://people.debian.org/~csmall/ipv6/

IPv6 engedélyezés

"`ipv6`" beírandó az "`/etc/modules`"-ba

Cím konfiguráció: "`/etc/network/interfaces`" :

```
iface eth0 inet6 static
address 2001:XXXX:YYYY:ZZZZ::1
netmask 64
```

Debián konfiguráció/2

Tunnel konfiguráció: "`/etc/network/interfaces`" :

```
iface tun0 inet6 v4tunnel
   endpoint A.B.C.D
   address 2001:XXXX:1:YYYY::2
   gateway 2001:XXXX:1:YYYY::1
   netmask 64
```

Debián konfiguráció/3

RA konfiguráció Debian routeren "/etc/radvd.conf"-ban :

```
interface eth0
{
    AdvSendAdvert on;
    AdvLinkMTU 1472;
    prefix 2001:XXXX:YYYY:ZZZZ:/64 {
        AdvOnLink on;
        AdvPreferredLifetime 3600;
        AdvValidLifetime 7200;
    };
};
```

Solaris configuration (1)

Supported since Solaris 8

- autoconfiguration, IPv4 tunnel, 6to4, IPSec, applications

Solaris configuration (2)

Autoconfiguration

existing "`/etc/hostname6.<intf>`"

Static address configuration "`/etc/hostname6.<intf>`" :

```
addif 2001:DB8:1:2::100 up
```

Static name ↔IPv6 address resolution:

```
in /etc/inet/ipnodes
```

DNS resolution should be enabled

/etc/nsswitch.conf

ipnodes: files dns

Mac OS X configuration (1)

Supported since Mac OS X 10.2 (since Darwin kernel version 6)

- autoconfiguration, IPv4 tunnel, 6to4, IPSec, applications, Apple Filing Protocol (since AFP version 3.1)
- Rendez-vous point supports IPv6
- Basically – what you can expect from *BSD

Mac OS X configuration (2)

Enabled by ip6config command

`ip6config command interface`

- commands:
  - start-v6 –enable IPv6 on given (all) interface
  - stop-v6 –disable IPv6 on given (all) interface
  - start-stf – enable IPv6 as defined in /etc/6to4.conf
  - start-rtadvd – start router advertisement daemon and enable IPv6 packet forwarding between interfaces
- ip6 – enable disable per interface

Autoconfiguration

enabled by default


deploy
Questions?
6DEPLOY Project Web Site:
http://www.6deploy.org
mohacsi@niif.hu