IPv6 ajánlások NIIF tagintézmények számára

IPv6 támogatás konfigurációja Window környezetben

Teljeskörű IPv6 protokoll támogatást érhető el Windows 7, Vista , Windows XP SP1 és utánna valamint Windows Server 2003 és 2008 környezetben.

Windows 7 specifikus információk

IPv6 alap értelmezés szerint installált és engedélyezett. Konfigurálni GUI-n és `netsh` interfészen is is lehet.
IPv6 támogatés mint Vista esetében, de fő különbségek.

- Random Interface ID alkalmazása alapbeállításként engedélyezett -RFC 3041
    - Konfiguráció: netsh interface ipv6 set global [[randomizeidentifiers=]enabled|disabled]
- Új lehetőség: IP-HTTPS (IP over Secure HTTP)
- Új lehetőség: DirectAccess • Transzparens VPN kétirányú kommunikáció támogatással Windows Server 2008 szükséges

Vista specifikus információk

IPv6 alap értelmezés szerint installált és engedélyezett
Nem csak az alap IPv6 funkciók működnek mint Windows XP és 2003 esetén, hanem a következő funkciók is.

- Új protokoll stack elválasztott IPv4 és IPv6 socketekkel.
- GUI alapú konfiguráció
- IPsec támogatás
- MLDv2
- IPv6 transzport fölött DNS üzenetek támogatása
- LLMNR (Link Local Multicast Name Resolution)
- Literal IPv6 cím az URL-ekben
- IPv6 over PPP
- DHCPv6

Windows XP specifikus információk

Nem installált alap esetben – funkciók SP-nként különbözőek
SP1:

- GUI installáció
- `netsh` interfész támogatása

SP2:

- Teredo kliens
- host-specific relay támogatás
- IPv6 tűzfal

IPv6 installáció Windows XP-n

Service pack nélkül:
• „ipv6 install” parancs kiadása parancs sorból
SP1:
• “Microsoft IPv6 Developer Edition” protokoll installálás a Connection Properties ablakból
SP2:
• “Microsoft TCP/IP version 6” protokoll installálás a Connection Properties ablakból

Windows Konfiguáció

Statikus cím beállítás

```
netsh ipv6 interface {add|set} address [interface=] <interface> [address=]
<address> <interface> - interface name vagy index –  <address> - IPv6 cím
```

Statikus cím törlés

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

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
```

Windows alkalmazások

ipconfig, netstat, ping6, tracert6, pathping
Minden Wininet.dll alkalmazás

- ftp, telnet, Iexplorer, Firefox, Thunderbird

IPv6 támogatás Linux könyezetben

Támogatás Kernel 2.2.x óta – javasolt legalább2.4.8

- autoconfiguration,
- IPv4 tunnel,
- 6to4

USAGI patch (mostly included in 2.6.x series)

- Node information query,
- anycast,
- ISATAP,
- privacy extension,
- IPSec

IPv6 installáció

Kernel fordítási opció:
`CONFIG_IPv6=m/y`
Ha az IPv6 modul működik, akkor a `/proc/net/if_inet6` fájl létezik.
IPv6 module betöltése
`modprobe ipv6`
Utánna autokonfiguráció működik
`ifconfig`

Linux konfiguráció

Statikus cím konfiguráció
`ifconfig <interface> inet6 add <ipv6address>/<prefixlength>`
statikus cím törlése
`ifconfig <interface> inet6 add <ipv6address>/<prefixlength>`

Neighbor cache
`ip -6 neigh show`

IPv6 routing table
`route -A inet6/netstat`

Debian Linux konfiguráció

IPv6 engedélyezés:
`"ipv6"` module a `"/etc/modules"` fájlban

Statikus cím konfiguráció
`"/etc/network/interfaces"`

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

Autokonfigurált cím letiltása:

`net.ipv6.conf.<all|eth0>.autoconf = 0`

(- /etc/sysctl.conf fájlban)

Router adevertisment generálása:

```
"/etc/radvd.conf" :
interface eth0 {
      AdvSendAdvert on;
      AdvLinkMTU 150

      prefix 2001:738:YYYY:ZZZZ:/64 {
      AdvOnLink on;
      AdvPreferredLifetime 3600;
      AdvValidLifetime 7200;
      };
};
```

Továbbis Debian specifikus információk

http://wiki.debian.org/DebianIPv6

Alkalmazások

Bind 9 konfiguráció

named.conf bejegyzések:

• Több mint 1 listen-on-v6 opció:

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

};
```

IPv6 kiszolgálás letiltás

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

Zone transfer:

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

Lekérdezés IPv6 felett:

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

Inetd

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

Apache

2.0.x+ verziók támogatják az IPv6-ot automatikusan
–--enable-v4-mapped

Konfigurációs direktiva

```
Listen ::
Listen [::]:80
```
NameVirtualHost (IPv6 cím szintén)
Access control működik – Ne felejtsük el az ACL IPv6 címmel kiegészíteni
WebDAV szintén

OpenSSH

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

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

Squid

Támogatott 3.1 után
Az ACL-ben be lehet írni IPv6 címeket:

```
acl localnet src 2001:738:ffff::/48
```

További információk:

Nagy lista az IPv6 képes alklamazásokról

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

Kesehető IPv6 Application and Patch Database

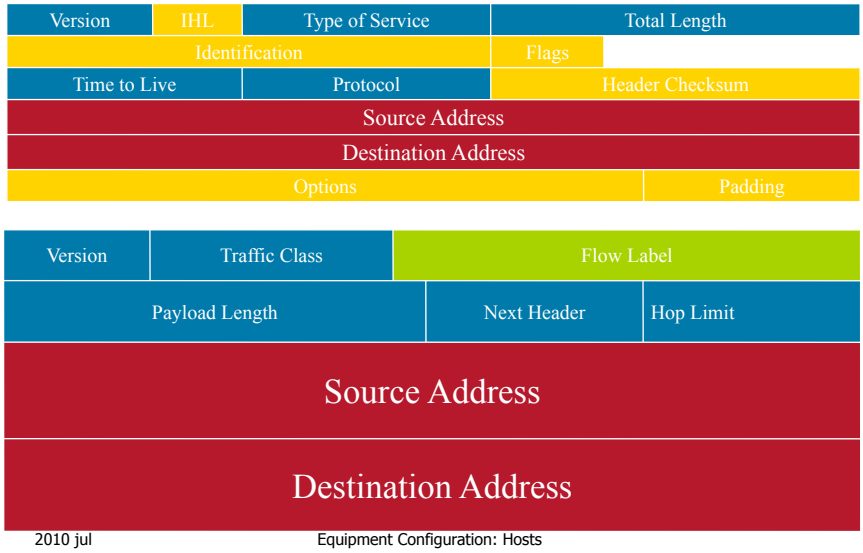
`http://ipv6.niif.hu/ipv6_apps/`

konfigurációs leírások

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

Melléklet angolul

IPv4 & IPv6 Fejléc összehasonlítás



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	**0..0:1111 1111:1111 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)

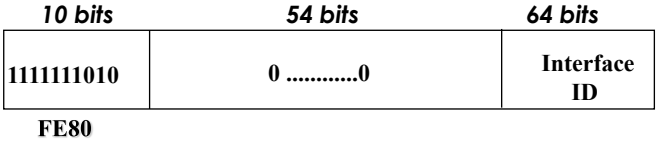
The **loopback address,** for sending packets to itself:

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

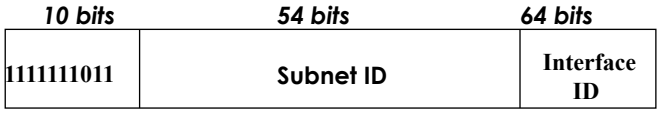
The **documentation prefix [RFC3849]:**
2001:db8::/32

Link-Local & Site-Local Unicast Addresses

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



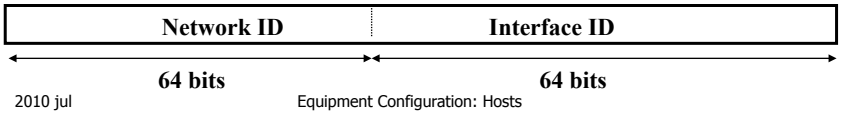
Site-local addresses for independence from changes of TLA / NLA* **(FEC0::/10):** (deprecated by RFC3879)



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

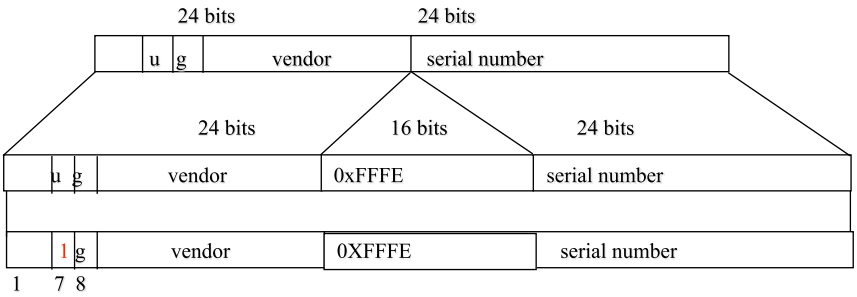


Autoconfigured Interface IDs (1)

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)



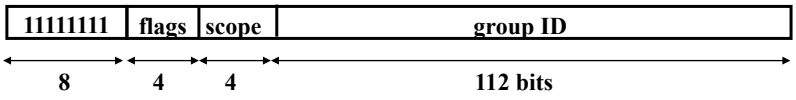
Autoconfigured Interface IDs (2)

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

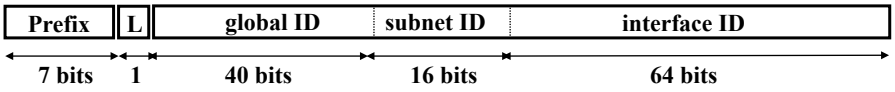
Unique Local IPv6 Unicast Addresses (2)

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**

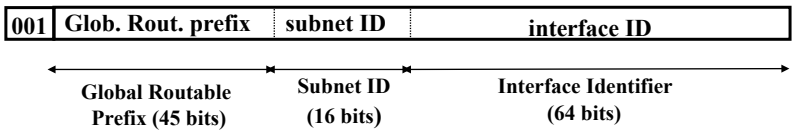
L = 0 may be defined in the future (in practice used for **centrally assigned** prefixes)

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

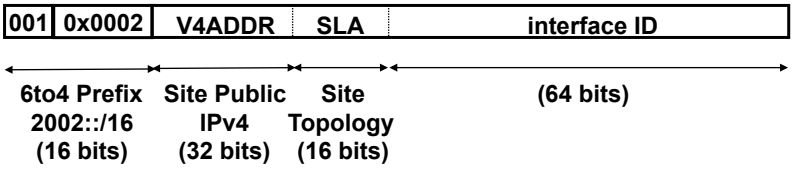
- Has been designed to be an hierarchical structure from the site administrator perspective

6to4 Addresses

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

Assigned Prefix: 2002::/16

To assign to sites 2002:V4ADDR::/48



6to4 Prefix 2002::/16 (16 bits) — **Site Public IPv4 (32 bits)** — **Site Topology (16 bits)** — **(64 bits)**

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)

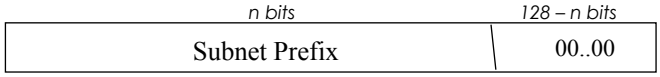
Taken from the unicast address space (of any scope).**Not syntactically distinguishable from unicast addresses**

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

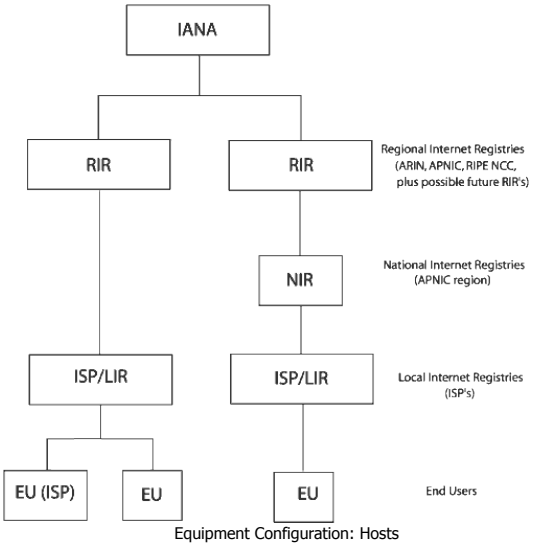
Anycast address cannot be source address of a packet

Reserved anycast addresses are defined in **RFC2526**

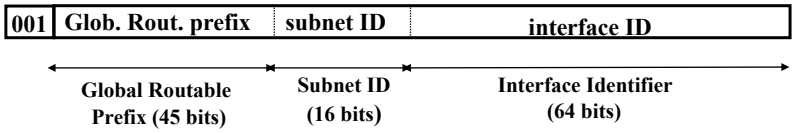
The Subnet-Router anycast address is predefined (mandatory on all routers):



Production Addressing Scheme (1)



Production Addressing Scheme (2)



LIRs receive by default /32

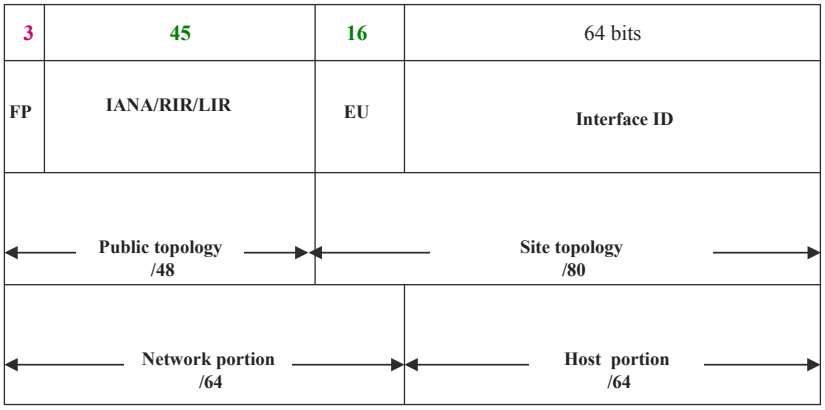
- Production addresses today are from prefixes 2001, 2003, 2400, etc.
- Can request for more if justified

/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 (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/

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.
```


Outline

Campus deployment strategy

Campus IPv6 address allocation

Campus deployment topology - options

Campus services

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

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

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
…	…

- 16 bits = 65535 subnets

⇒ *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. . .

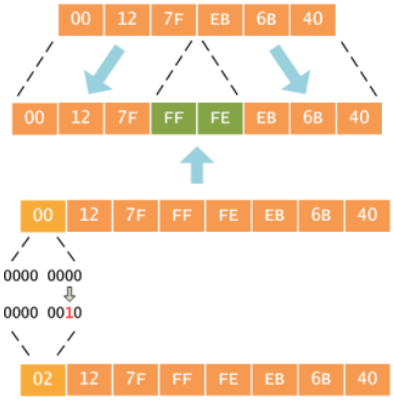
- Avoid using numeric addresses at all costs
- Avoid hard-configured addresses on hosts except for servers (this is very important for DNS servers) – use the feature that you can assign more than one IPv6 address to an interface (IPv6 alias address for servers)
- Anticipate that changing ISPs will mean 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**

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

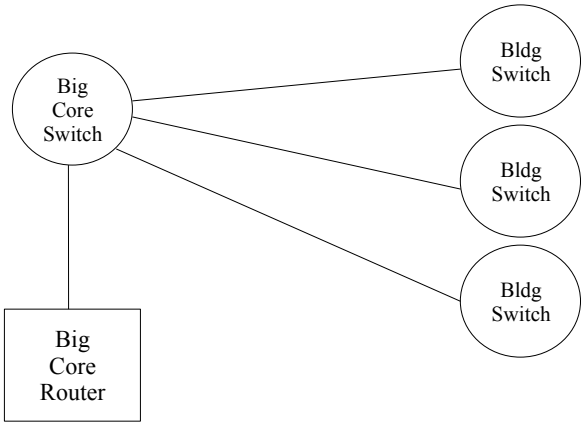
If the hosts/services are not dual stack enabled

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

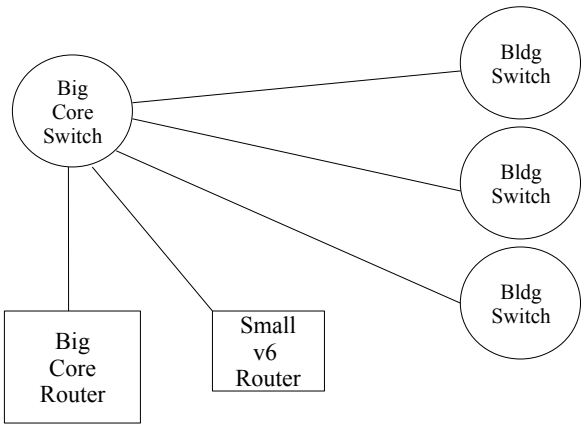
If the L3 devices cannot cope with IPv6 or administrators are not in favor of upgrading the router

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

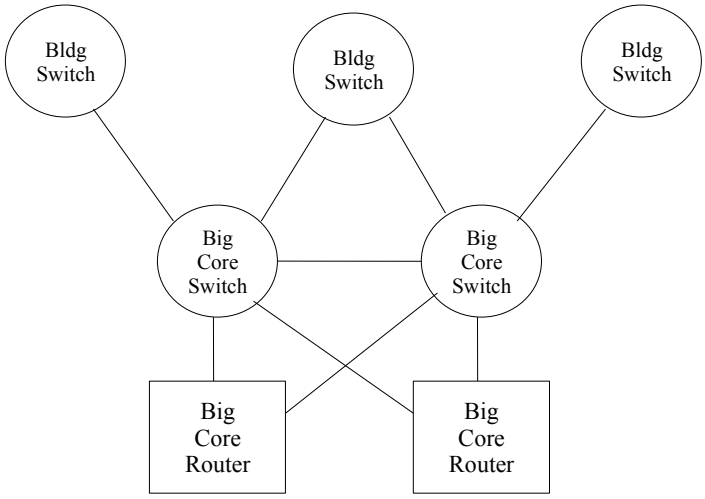
Layer-2 Campus - 1 Switch



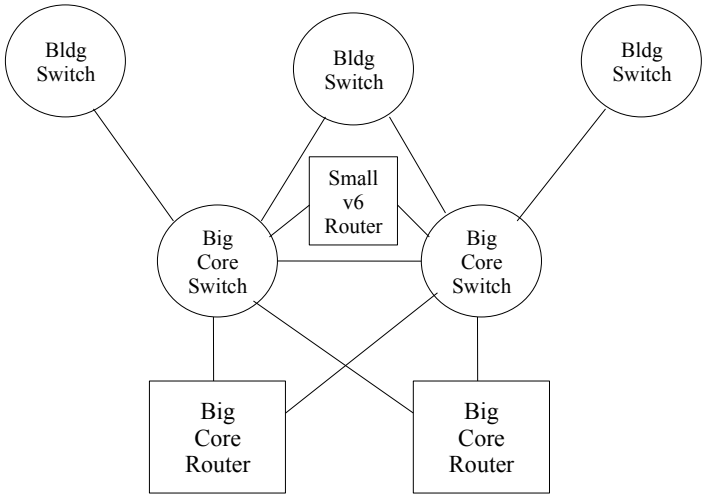
Layer-2 Campus - 1 Switch



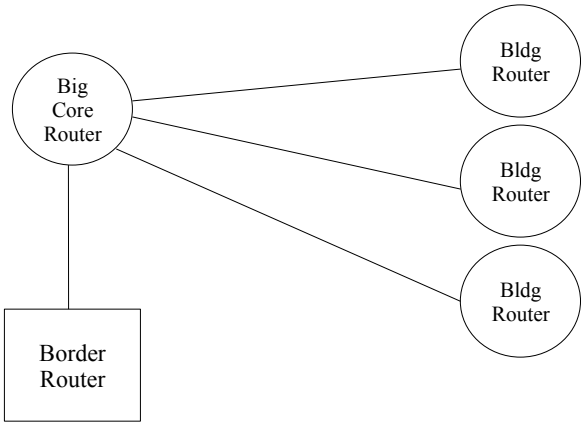
Layer-2 Campus
- Redundant Switches



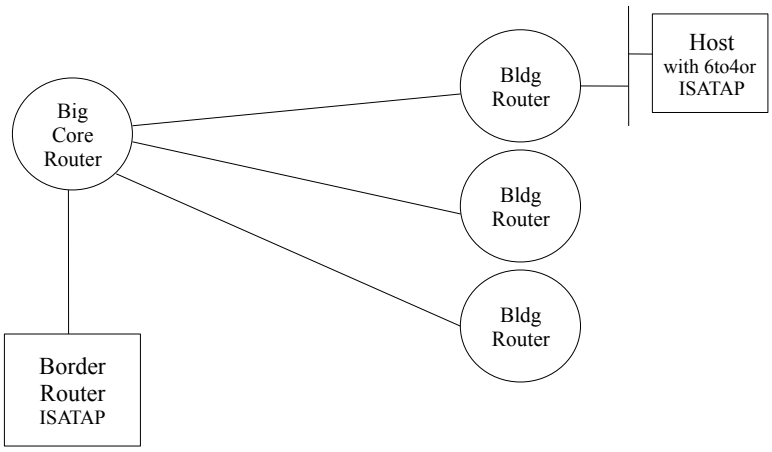
Layer-2 Campus
Redundant Switches



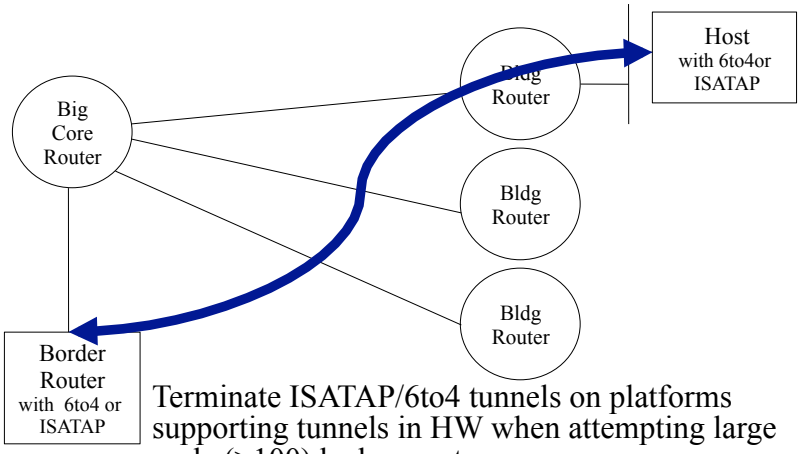
Layer-3 Campus



Layer-3 Campus

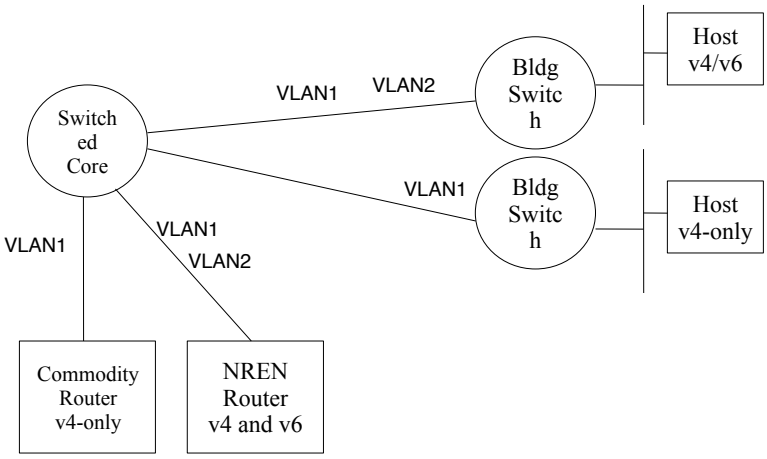


Layer-3 Campus



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

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

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)

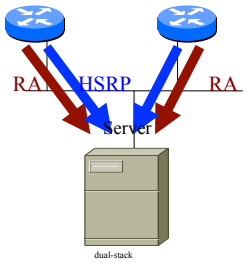
Default Gateway . . . . . : 10.121.10.1

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

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

Reachable Time : 6s

Base Reachable Time : 5s



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

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:

IPv6 specifikus	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: Szükséges a helyes müködéshez – kivéve statikus ND bejegyzések esetén
	RS/RA	szükséges: Stateless Address Autoconfigration esetén szükséges
	Packet too big	szükséges: Path MTU discovery
	MLD	Requirements in for multicast in architecture 1

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

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

IPv6 on Windows

Full support

- Windows 7, Vista and XP SP1 and later
- Windows Server 2003 and 2008

Technology preview

- Windows XP without SP
- Windows 2000 (not compatible with SP2 or later)

Developer Edition

- Windows NT 4.0 (source was available)

No official support but third party products available

- Windows 95/98/ME

Supported features:

- autoconfiguration, 6in4 tunnel, 6to4 tunnel, 6to4 relay, TEREDO tunnel, ISATAP tunnel, IPSec (manual keying)

IPv6 in Windows 7

IPv6 is installed by default

Configuration is based on GUI and netsh (see Vista)

IPv6 Support similar to Vista, differences are:

- Change: Random Interface ID is on by default (RFC 3041)
  - Doesn't use EUI-64 by default to get the interface ID in autoconfigured addresses: netsh interface ipv6 set global [[randomizeidentifiers=]enabled|disabled]
- New Feature: IP-HTTPS (IP over Secure HTTP)
- New Feature: DirectAccess
  - Transparent VPN allowing communication in both directions
  - Needs Windows Server 2008

IPv6 in Windows Vista

IPv6 is installed 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 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

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:

IPv6 on *BSD

Supported

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

Available since FreeBSD 4.0, OpenBSD 2.7, NetBSD 1.5

KAME extension

- NAT-PT, DHCPv6, PIM-(S)SM, multicast DNS, EDNS resolver, ISATAP (not any more), anycast (integrated)

FreeBSD configuration (1)

Installation: not necessary, the default kernel has it

The installer asking for IPv6 support

- `ipv6_enable="yes" in /etc/rc.conf`
- Autoconfiguration is working
- `ifconfig -a`

FreeBSD configuration (2)

Manual address configuration

- `ipv6_prefix_fxp0="2001:DB8:1:2"`
- `ipv6_ifconfig_fxp0="2001:DB8:1:2::1 prefixlen 64"`
- then `/etc/netstart`
- or `ifconfig`

Neighbor cache

- `ndp -a`

Routing table

- `route/netstat`

FreeBSD configuration (3)

Configuration of further addresses

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

What about if you don't have IPv6 connectivity

- ip6addrctl(8) program – according RFC3484 you can adjust default address selection

```
#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 configuration (4)

Reconfiguration

- `rtsol fxp0`

Applications

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

Further information

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

FreeBSD configuration (5)

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 (6)

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 (7)

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:
  ```


FreeBSD configuration (8)

RIPng: route6d daemon

`route6d`

`-L` *`IPv6_prefix,interface`* (receives only prefixes derived from *`IPv6_prefix`* on interface *`interface`*)

`-N` *`interface`* (do not receive and advertise routes on interface)

`-O` *`IPv6_prefix, interface`* (advertise only on interface the IPv6 prefix)

BGP: bgpd daemon

Better to use Zebra/Quaggua BGP daemon
  ```

IPv6 on Linux

Supported

- autoconfiguration, IPv4 tunnel, 6to4
- since Kernel 2.2.x recommended at least 2.4.8

USAGI patch (mostly included in 2.6.x series)

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

General Linux configuration (1)

Kernel compile options

- CONFIG_IPv6=m/y
- If the IPv6 module is loaded, file `/proc/net/if_inet6` should be present
- IPv6 module can be loaded by `modprobe ipv6`

Autoconfiguration supported

- `ifconfig`

General Linux configuration (2)

Address configuration

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

Neighbor cache

- `ip -6 neigh show`

IPv6 routing table

- `route -A inet6/netstat`

Red Hat configuration (1)

Enabling Global IPv6 support

`/etc/sysconfig/network` file:

```
NETWORKING_IPV6="yes"
```

Enabling IPv6 support on a particular interface

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

```
IPV6INIT="yes"
```

Configuring IPv6 interface address

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

```
IPV6ADDR="2001:DB8:20::291D:6A83/48"
```

Default route configuration

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

```
eth0 ::/0 2001:DB8:20::922:A678
```

Red Hat configuration (2)

Applications

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

Further information

- http://www.bieringer.de/linux/IPv6
- http://www.linux-ipv6.org/

Fedora configuration (1)

(Fedora Core 2 only) append to `/etc/sysconfig/network`:

- `NETWORKING_IPV6=yes`
- `IPV6_DEFAULTDEV="your exit device e.g. tun6to4"`

(Fedora Core 1 only) append to /etc/sysconfig/network

- `NETWORKING_IPV6=yes`
- `IPV6_GATEWAYDEV="your exit device e.g. tun6to4"`

6to4 gateway- append to /etc/sysconfig/network-scripts/ifcfg-eth0

- `IPV6INIT=yes`
- `IPV6TO4INIT=yes`

Debian configuration (1)

Enabling IPv6

You should put "`ipv6`" in "`/etc/modules`"

Address configuration

"/etc/network/interfaces":

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

Further information

http://wiki.debian.org/DebianIPv6

Debian configuration (2)

Tunnel configuration

"/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
```

Debian configuration (3)

Disable autoconfigured addresses

```
net.ipv6.conf.<all|eth0>.autoconf = 0
```


Debian configuration (4)

RA configuration on Debian router

"/etc/radvd.conf" :

```
interface eth0
{
   AdvSendAdvert on;
   AdvLinkMTU 1500;
   prefix 2001:XXXX:YYYY:ZZZZ:/64 {
      AdvOnLink on;
      AdvPreferredLifetime 3600;
      AdvValidLifetime 7200;
   };
};
```

Debian configuration (5)

Configuration on router

```
net.ipv6.conf.all.autoconf = 0
net.ipv6.conf.all.accept_ra = 0
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.all.forwarding = 1
net.ipv6.conf.all.router_solicitations = 0
```

Firewalls – to allow tunnels

```
iptables -I INPUT -j ACCEPT --proto 41
```


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