# IPv6: Introduction

Bartłomiej Rodek
Bartosz Mazurczyk

Inter Projekt S.A.
IDEA4PRO Sp. z o.o.

10 March 2011, Budapest

# Outline

1 Introduction to IPv6

2 RouterOS IPv6 support

3 Routing protocols

4 How to start

# Introduction to IPv6

# Puproses of IPv6 design

The IPv4 address space is too small.
Along with extended address space size there are introduced some new capabilities and improvements.

# Main advantages of the new protocol

- Extended addressing capabilies

# Main advantages of the new protocol

- Extended addressing capabilies
    - Extended address space (128bit against 32bit in IPv4)

# Main advantages of the new protocol

- Extended addressing capabilies
  - Extended address space (128bit against 32bit in IPv4)
  - New type of addresses - anycast

# Main advantages of the new protocol

- Extended addressing capabilies
  - Extended address space (128bit against 32bit in IPv4)
  - New type of addresses - anycast
  - Multicast addresses have ,,scope” field

# Main advantages of the new protocol

- Extended addressing capabilies
  - Extended address space (128bit against 32bit in IPv4)
  - New type of addresses - anycast
  - Multicast addresses have ,,scope" field
- Header format simplification - faster processing in most common cases

# Main advantages of the new protocol

- Extended addressing capabilies
  - Extended address space (128bit against 32bit in IPv4)
  - New type of addresses - anycast
  - Multicast addresses have ,,scope” field
- Header format simplification - faster processing in most common cases
  - No checksums in IPv6 header

# Main advantages of the new protocol

- Extended addressing capabilies
    - Extended address space (128bit against 32bit in IPv4)
    - New type of addresses - anycast
    - Multicast addresses have ,,scope" field
- Header format simplification - faster processing in most common cases
    - No checksums in IPv6 header
    - Optional fields are moved to the extension headers

# Main advantages of the new protocol

- Extended addressing capabilies
  - Extended address space (128bit against 32bit in IPv4)
  - New type of addresses - anycast
  - Multicast addresses have ,,scope" field
- Header format simplification - faster processing in most common cases
  - No checksums in IPv6 header
  - Optional fields are moved to the extension headers
- Privacy and authentication

# Main advantages of the new protocol

- Extended addressing capabilies
  - Extended address space (128bit against 32bit in IPv4)
  - New type of addresses - anycast
  - Multicast addresses have ,,scope" field
- Header format simplification - faster processing in most common cases
  - No checksums in IPv6 header
  - Optional fields are moved to the extension headers
- Privacy and authentication
  - Authentication Header is not any longer optional

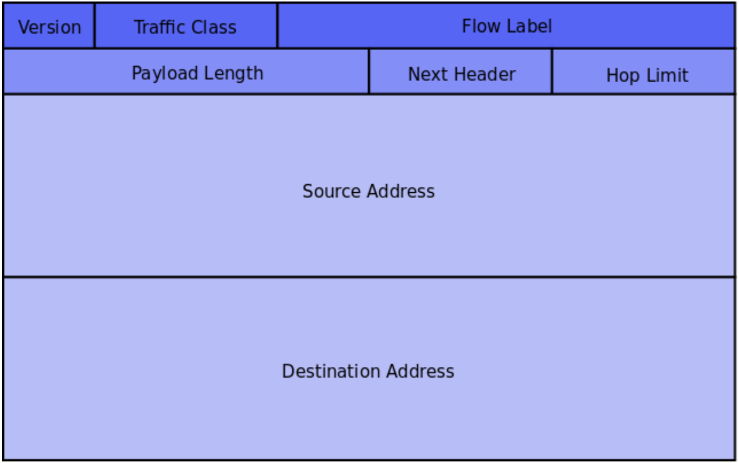
# Main advantages of the new protocol

- Extended addressing capabilies
    - Extended address space (128bit against 32bit in IPv4)
    - New type of addresses - anycast
    - Multicast addresses have ,,scope" field
- Header format simplification - faster processing in most common cases
    - No checksums in IPv6 header
    - Optional fields are moved to the extension headers
- Privacy and authentication
    - Authentication Header is not any longer optional
    - Support for ESP

# Address space

- IPv4 address space (32 bits):
  $2^{32} = 4294967296$ addresses
- IPv6 address space (128 bits):
  $2^{128} = 340282366920938463463374607431768211456$ addresses

# IPv6 Header Format



Version

Traffic Class

Flow Label

Payload Length

Next Header

Hop Limit

Source Address

Destination Address

# IPv6 Header Format

IPv6 header fields explained:

| | |
|---|---|
| Version | 4-bit Internet Protocol number = 6 |
| Traffic Class | 8-bit traffic class field |
| Flow Label | 20-bit flow label |
| Payload Length | Length of the payload |
| Next Header | 8-bit identificator of the next header |
| Hop Limit | 8-bit field. Equivalent for TTL from IPv4 |
| Source Address | 128-bit originator addresses |
| Destination Address | 128-bit recipent address |

# IPv6 addresses

- 128bit length

# IPv6 addresses

- 128bit length
- Hexadecimal notation - every two bytes are separated by ,,:" sign

# IPv6 addresses

- 128bit length
- Hexadecimal notation - every two bytes are separated by ,,:" sign
- Three types of addresses:

# IPv6 addresses

- 128bit length
- Hexadecimal notation - every two bytes are separated by ,,:" sign
- Three types of addresses:
    - Unicast

# IPv6 addresses

- 128bit length
- Hexadecimal notation - every two bytes are separated by ,,:" sign
- Three types of addresses:
  - Unicast
  - Multicast

# IPv6 addresses

- 128bit length
- Hexadecimal notation - every two bytes are separated by ,,:" sign
- Three types of addresses:
  - Unicast
  - Multicast
  - Anycast

# IPv6 addresses

- 128bit length
- Hexadecimal notation - every two bytes are separated by ,,:" sign
- Three types of addresses:
  - Unicast
  - Multicast
  - Anycast

**Note:**

There is no broadcast address in IPv6. It's replaced by multicast address ,,all nodes on link"

# Notation of IPv6 address

128bit IPv6 address is represented by 8 groups of hexadecimal digits separated by colon.

# Notation of IPv6 address

128bit IPv6 address is represented by 8 groups of hexadecimal digits separated by colon.

**Example of global unicast IPv6 address:**

2001:06a0:0176:0010:0000:0000:0000:0234

# Notation of IPv6 address

128bit IPv6 address is represented by 8 groups of hexadecimal digits separated by colon.

### Example of global unicast IPv6 address:

2001:06a0:0176:0010:0000:0000:0000:0234

### Leading zeros:

All leading zeroes can be ommited:
2001:6a0:176:10:0000:0000:0000:234

# Notation of IPv6 address

128bit IPv6 address is represented by 8 groups of hexadecimal digits separated by colon.

### Example of global unicast IPv6 address:

2001:06a0:0176:0010:0000:0000:0000:0234

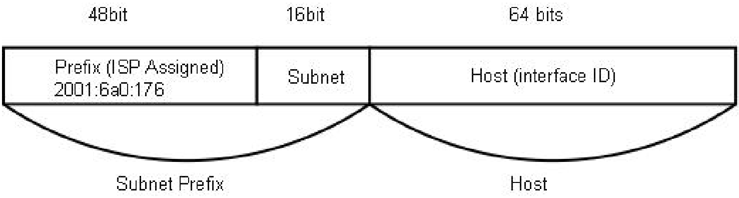
### Leading zeros:

All leading zeroes can be ommited:
2001:6a0:176:10:0000:0000:0000:234

### Group of four zeroes:

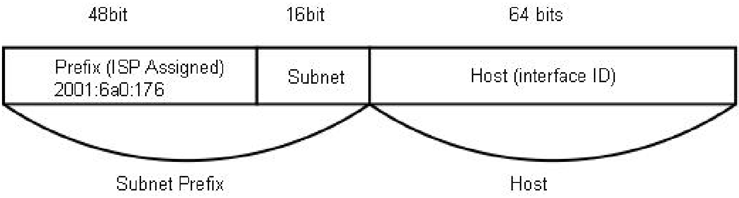
All groups of four zeroes can be shorten to double colon:
2001:6a0:176:10::234 (it's still the same address)

## Global unicast address



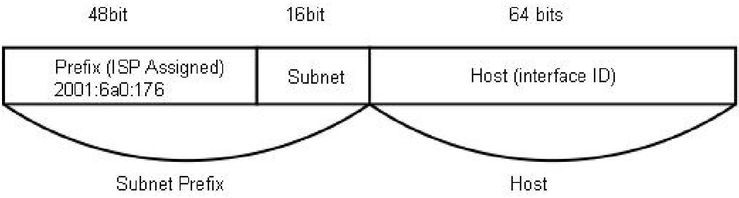
- Network portion

# Global unicast address



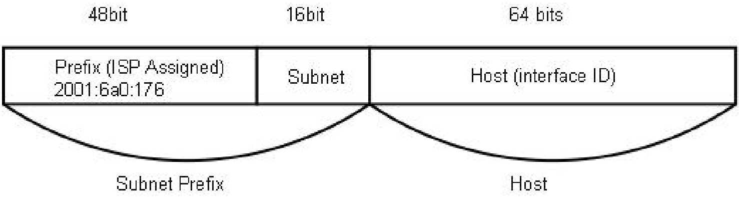
- Network portion
  - Prefix - globally routeable prefix assigned to the site

# Global unicast address



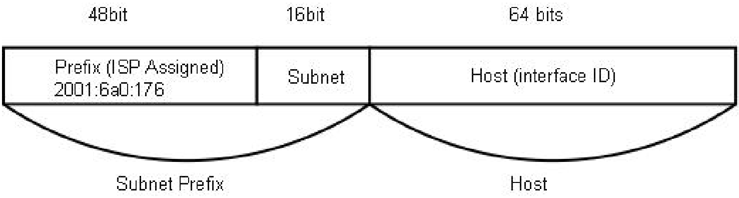
- Network portion
  - Prefix - globally routeable prefix assigned to the site
  - Subnet - identifies subnet within the site

# Global unicast address



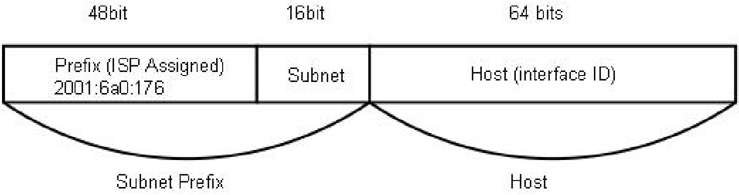
- Network portion
  - Prefix - globally routeable prefix assigned to the site
  - Subnet - identifies subnet within the site
- Host portion

# Global unicast address



- Network portion
  - Prefix - globally routeable prefix assigned to the site
  - Subnet - identifies subnet within the site
- Host portion
  - Interface ID - unique identifier (within the site) of the interface (host)

# Global unicast address



example prefix: 2001:6a0:176::/48

| | | |
|---|---|---|
| Registry prefix | ICANN to RIRa | 2001::/12 |
| ISP prefix | RIR to ISP or LIR | 2001:6a0::/32 |
| Site prefix | ISP to customer | 2001:6a0:176::/48 |
| Subnet prefix | admin for link | 2001:6a0:176:10::/64 |

# Interface ID

- 64 bits length

# Interface ID

- 64 bits length
- manualy configured

# Interface ID

- 64 bits length
- manualy configured
- assigned by DHCP

# Interface ID

- 64 bits length
- manualy configured
- assigned by DHCP
- auto-assigned from 48-bit MAC address (EUI-64)
  seventh bit of first part of MAC reversed + FFFE + second part of MAC address
  00:34:56:78:9A:BC will be changed to 0234:56FF:FE78:9ABC

# IPv6 and VLSM

- Do you need it at all?

# IPv6 and VLSM

- Do you need it at all?

# IPv6 and VLSM

- Do you need it at all?

**Note:**

In worst case you should get /48 prefix. It means you have 16 bits for subnets and 64 bits for interface id. I think it's enough. Do you think? If you have 65536 subnets available even point-to-point links can be addressed as /64.

# Multicast addresses

Multicast prefix begins from ffxy where y is a scope of the address. Some widely used scopes:

- ffx2::/16 - link-local - this packets might not be routed to anywhere
- ffx5::/16 - site-local - packets restricted to the local physical network
- ffxe::/16 - global scope - it can be routed through the Internet

# Multicast addresses

Well know multicast IPv6 addresses:

- ff02::1 - all nodes on the local network segement (equivalent of the IPv4 broadcast address)
- ff02::2 - all routers on the local network segment
- ff02::5 - AllSPF routers (OSPFv3)
- ff02::6 - AllDR routers (OSPFv3)
- ff02::9 - RIP routers
- ff05::1 - all nodes on the local network site

# Link-local addresses

- fe80::/10

# Link-local addresses

- fe80::/10
- valid only on a single link

# Link-local addresses

- fe80::/10
- valid only on a single link
- autoassigned

# Link-local addresses

- fe80::/10
- valid only on a single link
- autoassigned
- not routable in the Internet

# Link-local addresses

- fe80::/10
- valid only on a single link
- autoassigned
- not routable in the Internet
- can be uesd as next hop

# Other special address types

- Unspecified address:
  0:0:0:0:0:0:0:0/128 (or ::/128)
- Loopbak address:
  0:0:0:0:0:0:0:1/128 (::1/128)
- Link-local:
  fe80::/10
- Address reserved for documentation purposes:
  2001:db8::/32

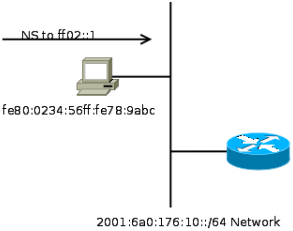
# Why you don't need DHCP (at last in some cases)

One of the integral part of IPv6 is stateless auto-configuration. Host (node) is able to configure IPv6 global address by itself. This means that in most cases you don't need DHCP.
The stateless auto-configuration is performed in two main steps:

- Link-local address generation
- Global unicast address generation

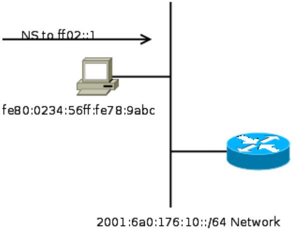
# Link-local assignment

1. Host is turned on,
   it generates
   link-local address



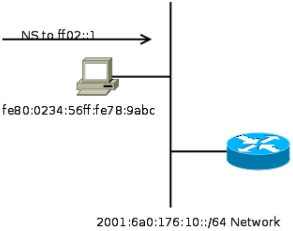
# Link-local assignment

1. Host is turned on, it generates link-local address
2. DAD is performed - host sends Neighbor Solicitation message to all nodes



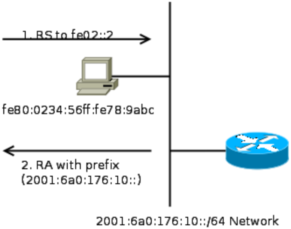
# Link-local assignment

1. Host is turned on, it generates link-local address
2. DAD is performed - host sends Neighbor Solicitation message to all nodes
3. If no response - generated address is unique and address is assigned to the host



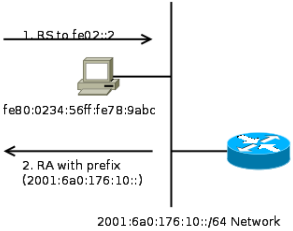
# Global unicast assignment

1. Host sends Router Solicitation message to all routers



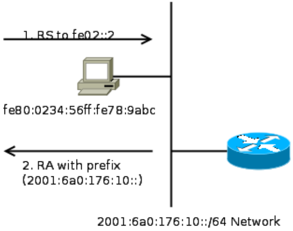
# Global unicast assignment

1. Host sends Router Solicitation message to all routers
2. Router replys with Router Advertisement message



# Global unicast assignment

1. Host sends Router Solicitation message to all routers
2. Router replys with Router Advertisement message
3. Host learns the global prefix, new address is generated, performs DAD and assignement



# IPv6 and NAT

There's no NAT in IPv6 (howewer it is being discussed[1]).
But, hey, you don't need NAT...

[1] https://datatracker.ietf.org/doc/draft-mrw-nat66/

# RouterOS IPv6 support

# RouterOS services and protocols

MikroTik RouterOS currently supports:

# RouterOS services and protocols

MikroTik RouterOS currently supports:

- Addressing and routing

# RouterOS services and protocols

MikroTik RouterOS currently supports:

- Addressing and routing
- Stateless autoconfiguration

# RouterOS services and protocols

MikroTik RouterOS currently supports:

- Addressing and routing
- Stateless autoconfiguration
- Neighbor Discovery Protocol

# RouterOS services and protocols

MikroTik RouterOS currently supports:

- Addressing and routing
- Stateless autoconfiguration
- Neighbor Discovery Protocol
- Tunneling IPv6 over IPv4 (6to4 interface)

# RouterOS services and protocols

MikroTik RouterOS currently supports:

- Addressing and routing
- Stateless autoconfiguration
- Neighbor Discovery Protocol
- Tunneling IPv6 over IPv4 (6to4 interface)
- Extension headers handling in Firewall (in limited form)

# RouterOS services and protocols

MikroTik RouterOS currently supports:

- Addressing and routing
- Stateless autoconfiguration
- Neighbor Discovery Protocol
- Tunneling IPv6 over IPv4 (6to4 interface)
- Extension headers handling in Firewall (in limited form)
- IPSec since 5.0rc8 version

# RouterOS services and protocols

MikroTik RouterOS currently supports:

- Addressing and routing
- Stateless autoconfiguration
- Neighbor Discovery Protocol
- Tunneling IPv6 over IPv4 (6to4 interface)
- Extension headers handling in Firewall (in limited form)
- IPSec since 5.0rc8 version
- Winbox, telnet, SSH, ping, traceroute

# RouterOS services and protocols

MikroTik RouterOS currently supports:

- Addressing and routing
- Stateless autoconfiguration
- Neighbor Discovery Protocol
- Tunneling IPv6 over IPv4 (6to4 interface)
- Extension headers handling in Firewall (in limited form)
- IPSec since 5.0rc8 version
- Winbox, telnet, SSH, ping, traceroute
- Routing protocols RIPng, OSPFv3, BGP-MP

# RouterOS services and protocols

MikroTik RouterOS currently supports:

- Addressing and routing
- Stateless autoconfiguration
- Neighbor Discovery Protocol
- Tunneling IPv6 over IPv4 (6to4 interface)
- Extension headers handling in Firewall (in limited form)
- IPSec since 5.0rc8 version
- Winbox, telnet, SSH, ping, traceroute
- Routing protocols RIPng, OSPFv3, BGP-MP
- DNS and WebProxy

# What's in plans

To be done in the nearest future:

- DHCP server

# What's in plans

To be done in the nearest future:

- DHCP server
- Policy Routing

# What's in plans

To be done in the nearest future:

- DHCP server
- Policy Routing
- Multicast Routing

# What's in plans

To be done in the nearest future:

- DHCP server
- Policy Routing
- Multicast Routing
- Pools

# MikroTik RouterOS and IPv6

Make sure you have *ipv6* package installed, if you plan to use routing protocols you need also the *routing* package

```
[admin@MikroTik] > system package print
Flags: X - disabled
 #   NAME                     VERSION                    SCHEDULED
...
 5   security                 5.0rc10
 6   routing                  5.0rc10
 7   ipv6                     5.0rc10
 8   advanced-tools           5.0rc10
 9   wireless                 5.0rc10
...
[admin@MikroTik] >
```

# Static addressing and routing

Adding and printing the IPv6 address:

```
[admin@MikroTik] > ipv6 address add address=2001:6a0:176:1::2/64\
          interface=sit1 advertise=no
[admin@MikroTik] > ipv6 address print
Flags: X - disabled, I - invalid, D - dynamic, G - global, L - link-local
#    ADDRESS                                     INTERFACE           ADVERTISE
 0  G 2001:6a0:176:1::2/64                        sit1                no
 1  G 2001:6a0:176:10::1/64                       ether3              no
 ...
10 DL fe80::20c:42ff:fe21:c053/64                 ether3              no
```

Adding a default route:

```
[admin@MikroTik] > ipv6 route add dst-address=:: gateway=2001:6a0:176:1::1
```

# Static addressing and routing

Adding and printing the IPv6 address:

```
[admin@MikroTik] > ipv6 address add address=2001:6a0:176:1::2/64\
          interface=sit1 advertise=no
[admin@MikroTik] > ipv6 address print
Flags: X - disabled, I - invalid, D - dynamic, G - global, L - link-local
#    ADDRESS                                        INTERFACE           ADVERTISE
 0  G 2001:6a0:176:1::2/64                           sit1                 no
 1  G 2001:6a0:176:10::1/64                          ether3               no
 ...
10 DL fe80::20c:42ff:fe21:c053/64                    ether3               no
```

Adding a default route:

```
[admin@MikroTik] > ipv6 route add dst-address=:: gateway=2001:6a0:176:1::1
```

# IPv6 routing table

Routing table for new version of the IP protocol:

```
[admin@MikroTik] > ipv6 route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, o - ospf, b - bgp, U - unreachable
 #      DST-ADDRESS              GATEWAY                  DISTANCE
 0  ADo  ::/0                     fe80::5be0:8e04%sit1     110
 1  ADC  2001:6a0:176:1::/64      sit1                     0
 2  ADC  2001:6a0:176:2::/64      ether2                   0
 3  ADo  2001:6a0:176:4::/64      fe80::20c:42ff:fe38:9... 110
...
 8  ADo  2001:6a0:200:bd::/64     fe80::5be0:8e04%sit1     110
```

# IPv6 routing table

Routing table for new version of the IP protocol:

```
[admin@MikroTik] > ipv6 route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, o - ospf, b - bgp, U - unreachable
 #      DST-ADDRESS              GATEWAY                  DISTANCE
 0 ADo  ::/0                     fe80::5be0:8e04%sit1     110
 1 ADC  2001:6a0:176:1::/64      sit1                     0
 2 ADC  2001:6a0:176:2::/64      ether2                   0
 3 ADo  2001:6a0:176:4::/64      fe80::20c:42ff:fe38:9... 110
...
 8 ADo  2001:6a0:200:bd::/64     fe80::5be0:8e04%sit1     110
```

# RouterOS services ready for IPv6

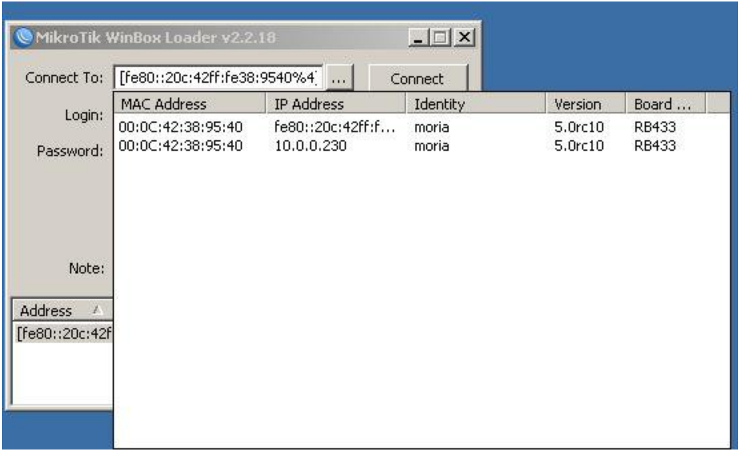
- ssh
- telnet:

```
[root@cor(pts/0)] telnet stargate
Trying 2001:6a0:176:1::2...
Connected to stargate
Escape character is ’^]’.
Password:
```

- ftp:

```
[root@cor(pts/0)] ftp 2001:6a0:176:1::2
Connected to 2001:6a0:176:1::2.
220 stargate FTP server (MikroTik 5.0rc10) ready
Name (2001:6a0:176:1::2:root):
```

# WinBox and IPv6



Connect To: [fe80::20c:42ff:fe38:9540%4]

Login:

Password:

Note:

| MAC Address | IP Address | Identity | Version | Board ... |
|---|---|---|---|---|
| 00:0C:42:38:95:40 | fe80::20c:42ff:f... | moria | 5.0rc10 | RB433 |
| 00:0C:42:38:95:40 | 10.0.0.230 | moria | 5.0rc10 | RB433 |

| Address |
|---|
| [fe80::20c:42f |

# IPv6 Firewall filter and mangle

- Most of the matchers in filter remain the same
- There is new matcher *header* - you can match by extension header type: hop, dst, route, frag, ah, esp and proto.

```
[admin@MikroTik] > ipv6 firewall filter add chain=input action=accept  \
protocol=tcp src-address=2001:6a0:176::/48 dst-port=22
```

# Dual stack RouterOS

```
[admin@MikroTik] > ping 2001:838:2:1::30:67
HOST                                    SIZE  TTL TIME  STATUS
2001:838:2:1::30:67                     56    50  254ms echo reply
2001:838:2:1::30:67                     56    50  370ms echo reply
    sent=2 received=2 packet-loss=0% min-rtt=254ms avg-rtt=312ms
   max-rtt=370ms

[admin@MikroTik] > ping 4.2.2.2
HOST                                    SIZE  TTL TIME  STATUS
4.2.2.2                                 56    245 47ms
4.2.2.2                                 56    245 37ms
4.2.2.2                                 56    245 36ms
4.2.2.2                                 56    245 79ms
    sent=4 received=4 packet-loss=0% min-rtt=36ms avg-rtt=49ms max-rtt=79ms

[admin@MikroTik] >
```

# Routing protocols

- RIPng
- OSPFv3
- BGP

# RIPng

- IPv6 implementation of RIP protocol
- Same advantages and disadvantages as RIP

# RIPng

- IPv6 implementation of RIP protocol
- Same advantages and disadvantages as RIP
  - slow convergence time

# RIPng

- IPv6 implementation of RIP protocol
- Same advantages and disadvantages as RIP
  - slow convergence time
  - temporary loops

# RIPng

- IPv6 implementation of RIP protocol
- Same advantages and disadvantages as RIP
  - slow convergence time
  - temporary loops
  - 16 hop limit

# RIPng

- IPv6 implementation of RIP protocol
- Same advantages and disadvantages as RIP
    - slow convergence time
    - temporary loops
    - 16 hop limit
    - does not scale well

# Differences from previous versions

- No authentication!

# Differences from previous versions

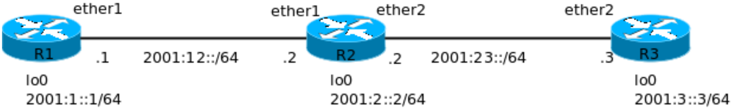
- No authentication!
- easier to configure (no network command, it's just interface)

# Differences from previous versions

- No authentication!
- easier to configure (no network command, it's just interface)
- uses multicast address ff02::9

# RIPng example - Topology

Simple topology for RIPng and OSPFv3 examples:



Lo0 interfaces are bridges without ports.

# RIPng example

Routing table before enabling RIPng on R1

```
[admin@R1] > ipv6 route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, o - ospf, b - bgp, U - unreachable
 #      DST-ADDRESS              GATEWAY                  DISTANCE
 0 ADC  2001:1::/64              loopback0                0
 1 ADC  2001:12::/64             ether1                   0
```

# RIPng example

Routing table before enabling RIPng on R2

```
[admin@R2] > ipv6 route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, o - ospf, b - bgp, U - unreachable
 #      DST-ADDRESS               GATEWAY                   DISTANCE
 0 ADC  2001:2::/64               loopback0                 0
 1 ADC  2001:12::/64              ether1                    0
 2 ADC  2001:23::/64              ether2                    0
```

# RIPng example

## Routing table before enabling RIPng on R3

```
[admin@R2] > ipv6 route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, o - ospf, b - bgp, U - unreachable
 #      DST-ADDRESS              GATEWAY                  DISTANCE
 0 ADC  2001:2::/64              loopback0                0
 1 ADC  2001:12::/64             ether1                   0
 2 ADC  2001:23::/64             ether2                   0
```

# RIPng example

Let's enable RIPng

```
[admin@R1] > routing ripng interface add interface=ether1
[admin@R1] > routing ripng interface add interface=loopback0 passive=yes

[admin@R2] > routing ripng interface add interface=ether1
[admin@R2] > routing ripng interface add interface=ether2
[admin@R2] > routing ripng interface add interface=loopback0 passive=yes

[admin@R3] > routing ripng interface add interface=ether2
[admin@R3] > routing ripng interface add interface=loopback0 passive=yes
```

# RIPng example

Routing Table on R1 after enabling RIP

```
[admin@R1] > ipv6 route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, o - ospf, b - bgp, U - unreachable
   #      DST-ADDRESS              GATEWAY                  DISTANCE
   0 ADC  2001:1::/64              loopback0                0
   1 ADC  2001:12::/64             ether1                   0
   2 ADr  2001:23::/64             fe80::20c:42ff:fe0e:f... 120
```

Why are loopbacks missing?

# RIPng example

Loopback are not advertised because now they do not have link local address

```
[admin@R1] > ipv6 add print
Flags: X - disabled, I - invalid, D - dynamic, G - global, L - link-local
 #    ADDRESS                                   INTERFACE        ADVERTISE
 0 DL fe80::20c:42ff:fe3e:f41c/64               ether1           no
 1 DL fe80::20c:42ff:fe3e:f41f/64               man_bridge       no
 2  G 2001:1::1/64                              loopback0        yes
 3  G 2001:12::1/64                             ether1           yes
```

# RIPng example

Loopback is a bridge interface without any port.
We need to specify admin-mac to get the link local Ipv6 address

```
[admin@R1] > interface bridge set loopback0 admin-mac=02:11:11:11:11:11
```

```
[admin@R1] > ipv6 add print
Flags: X - disabled, I - invalid, D - dynamic, G - global, L - link-local
 #    ADDRESS                                     INTERFACE           ADVERTISE
 0 DL fe80::20c:42ff:fe3e:f41c/64                 ether1              no
 1 DL fe80::20c:42ff:fe3e:f41f/64                 man_bridge          no
 2  G 2001:1::1/64                                loopback0           yes
 3  G 2001:12::1/64                               ether1              yes
 4 DL fe80::11:11ff:fe11:1111/64                  loopback0           no
```

# RIPng example

Loopback is a bridge interface without any port.
We need to specify admin-mac to get the link local Ipv6 address

```
[admin@R2] > interface bridge set loopback0 admin-mac=02:22:22:22:22:22
```

```
[admin@R2] > ipv6 add print
Flags: X - disabled, I - invalid, D - dynamic, G - global, L - link-local
 #    ADDRESS                                      INTERFACE           ADVERTISE
 0 DL fe80::20c:42ff:fe0e:f2f5/64                  ether2              no
 1 DL fe80::20c:42ff:fe0e:f2f6/64                  ether3              no
 2 DL fe80::20c:42ff:fe0e:f2f4/64                  ether1              no
 3  G 2001:2::2/64                                 loopback0           yes
 4  G 2001:12::2/64                                ether1              yes
 5  G 2001:23::2/64                                ether2              yes
 6 DL fe80::22:22ff:fe22:2222/64                   loopback0           no
```

# RIPng example

Loopback is a bridge interface without any port.
We need to specify admin-mac to get the link local Ipv6 address

```
[admin@R3] > interface bridge set loopback0 admin-mac=02:33:33:33:33:33
```

```
[admin@R3] > ipv6 address print
Flags: X - disabled, I - invalid, D - dynamic, G - global, L - link-local
 #    ADDRESS                                     INTERFACE          ADVERTISE
 0 DL fe80::20c:42ff:fe07:d48e/64                 ether2             no
 1 DL fe80::20c:42ff:fe07:d48f/64                 ether3             no
 2  G 2001:3::3/64                                loopback0          yes
 3  G 2001:23::3/64                               ether2             yes
 4 DL fe80::33:33ff:fe33:3333/64                  loopback0          no
```

# RIPng example

Routing Table on R1 after configuring admin-mac

```
[admin@R1] > ipv6 route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, o - ospf, b - bgp, U - unreachable
 #      DST-ADDRESS              GATEWAY                  DISTANCE
 0 ADC  2001:1::/64              loopback0                0
 1 ADr  2001:2::/64              fe80::20c:42ff:fe0e:f... 120
 2 ADr  2001:3::/64              fe80::20c:42ff:fe0e:f... 120
 3 ADC  2001:12::/64             ether1                   0
 4 ADr  2001:23::/64             fe80::20c:42ff:fe0e:f... 120
```

Now subnets of loopbacks are present.
Next-hop is a link local address!

# OSPFv3

- IPv6 implementation of OSPF protocol
- Same advantages and disadvantages as OSPFv2 (for IPv4)

# OSPFv3

- IPv6 implementation of OSPF protocol
- Same advantages and disadvantages as OSPFv2 (for IPv4)
    - fast convergence time

# OSPFv3

- IPv6 implementation of OSPF protocol
- Same advantages and disadvantages as OSPFv2 (for IPv4)
  - fast convergence time
  - loop free

# OSPFv3

- IPv6 implementation of OSPF protocol
- Same advantages and disadvantages as OSPFv2 (for IPv4)
  - fast convergence time
  - loop free
  - inter area summarization

# OSPFv3

- IPv6 implementation of OSPF protocol
- Same advantages and disadvantages as OSPFv2 (for IPv4)
    - fast convergence time
    - loop free
    - inter area summarization
    - scales very well

# OSPFv3 - Differences from previous versions

- No authentication! (but it makes the header shorter)

# OSPFv3 - Differences from previous versions

- No authentication! (but it makes the header shorter)
- Easier to configure (no network command, it's just interface)

# OSPFv3 - Differences from previous versions

- No authentication! (but it makes the header shorter)
- Easier to configure (no network command, it's just interface)
- Uses multicast address ff02::5 and ff02::6

# OSPFv3 - Differences from previous versions

- No authentication! (but it makes the header shorter)
- Easier to configure (no network command, it's just interface)
- Uses multicast address ff02::5 and ff02::6
- LSA9 carries prefix information

# OSPFv3 - Differences from previous versions

- No authentication! (but it makes the header shorter)
- Easier to configure (no network command, it's just interface)
- Uses multicast address ff02::5 and ff02::6
- LSA9 carries prefix information
- Instance number has global meaning inside the routing domain

# OSPFv3 - Differences from previous versions

- No authentication! (but it makes the header shorter)
- Easier to configure (no network command, it's just interface)
- Uses multicast address ff02::5 and ff02::6
- LSA9 carries prefix information
- Instance number has global meaning inside the routing domain
- Instance number between routers must match

# OSPFv3 - Differences from previous versions

- No authentication! (but it makes the header shorter)
- Easier to configure (no network command, it's just interface)
- Uses multicast address ff02::5 and ff02::6
- LSA9 carries prefix information
- Instance number has global meaning inside the routing domain
- Instance number between routers must match
- Admin-mac has to be configured for loopback

# OSPFv3 - RouterID

- Router ID is still a 32-bit value
- Needs to be configured manually otherwise the adjacency will not come up

```
[admin@R1] > routing ospf-v3 instance set default router-id=1.1.1.1
[admin@R2] > routing ospf-v3 instance set default router-id=2.2.2.2
[admin@R3] > routing ospf-v3 instance set default router-id=3.3.3.3
```

# OSPFv3 - basic configuration

Let's enable OSPFv3

```
[admin@R1] > routing ospf-v3 interface add interface=ether1 area=backbone
[admin@R1] > routing ospf-v3 interface add interface=loopback0 area=backbone
             passive=yes
```

```
[admin@R2] > routing ospf-v3 interface add interface=ether1 area=backbone
[admin@R2] > routing ospf-v3 interface add interface=ether2 area=backbone
[admin@R2] > routing ospf-v3 interface add interface=loopback0 area=backbone
             passive=yes
```

```
[admin@R3] > routing ospf-v3 interface add interface=ether2 area=backbone
[admin@R3] > routing ospf-v3 interface add interface=loopback0 area=backbone
             passive=yes
```

# OSPFv3 - basic configuration

Routing table on R1

```
[admin@R1] > ipv6 route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip,
 o - ospf, b - bgp,  U - unreachable
 #      DST-ADDRESS              GATEWAY                  DISTANCE
 0 ADC  2001:1::/64              loopback0                0
 1 ADo  2001:2::/64              fe80::20c:42ff:fe0e:f... 110
 2  Dr  2001:2::/64              fe80::20c:42ff:fe0e:f... 120
 3 ADo  2001:3::/64              fe80::20c:42ff:fe0e:f... 110
 4  Dr  2001:3::/64              fe80::20c:42ff:fe0e:f... 120
 5 ADC  2001:12::/64             ether1                   0
 6 ADo  2001:23::/64             fe80::20c:42ff:fe0e:f... 110
 7  Dr  2001:23::/64             fe80::20c:42ff:fe0e:f... 120
```

RIPng routes became incatcive because they have higher distance than OSPFv3.

# OSPFv3 - basic configuration

Adjacency table on R2

```
[admin@R2] > routing ospf-v3 neighbor print
 0 instance=default router-id=3.3.3.3 address=fe80::20c:42ff:fe07:d48e
   interface=ether2 priority=1 dr=2.2.2.2 backup-dr=3.3.3.3 state="Full"
   state-changes=4 ls-retransmits=0 ls-requests=0 db-summaries=0
   adjacency=7m13s

 1 instance=default router-id=1.1.1.1 address=fe80::20c:42ff:fe3e:f41c
   interface=ether1 priority=1 dr=1.1.1.1 backup-dr=2.2.2.2 state="Full"
   state-changes=5 ls-retransmits=0 ls-requests=0 db-summaries=0
   adjacency=10m14s
```

Router address is a link local address
Next-hop address is a link local as well

# BGP for IPv6

- Multi Protocol BGP, supports IPv6
- It's like BGP for IPv4 but it carries IPv6 prefixes
- Path selection algorithm remains the same

# BGP for IPv6 - Simple configuration

We need to configure

- BGP instance

# BGP for IPv6 - Simple configuration

We need to configure

- BGP instance
  - AS Number

# BGP for IPv6 - Simple configuration

We need to configure

- BGP instance
  - AS Number
  - Router ID

# BGP for IPv6 - Simple configuration

We need to configure

- BGP instance
  - AS Number
  - Router ID
- BGP peer (address, remote ASN, address-family)

# BGP for IPv6 - Simple configuration

We need to configure

- BGP instance
  - AS Number
  - Router ID
- BGP peer (address, remote ASN, address-family)
  - Remote address

# BGP for IPv6 - Simple configuration

We need to configure

- BGP instance
  - AS Number
  - Router ID
- BGP peer (address, remote ASN, address-family)
  - Remote address
  - Remote ASN

# BGP for IPv6 - Simple configuration

We need to configure

- BGP instance
  - AS Number
  - Router ID
- BGP peer (address, remote ASN, address-family)
  - Remote address
  - Remote ASN
  - Address family

# BGP for IPv6 - Simple configuration

We need to configure

- BGP instance
  - AS Number
  - Router ID
- BGP peer (address, remote ASN, address-family)
  - Remote address
  - Remote ASN
  - Address family
  - Optionally routing filters

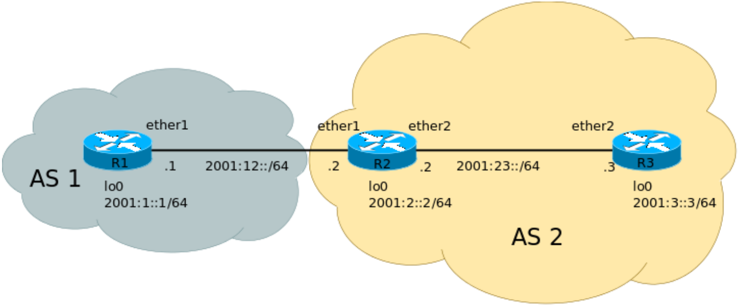
# BGP for IPv6 - Simple configuration

We need to configure

- BGP instance
  - AS Number
  - Router ID
- BGP peer (address, remote ASN, address-family)
  - Remote address
  - Remote ASN
  - Address family
  - Optionally routing filters
- Networks to advertise

# BGP for IPv6 configuration example - Topology

Topology used for BGP configuration example:



Lo0 interfaces are bridges without ports.

# BGP for IPv6 configuration example

- R1 belongs to AS1
- R2, R3 belong to AS2
- RIP and OSPF between R1 and R2 is disabled

# BGP for IPv6 configuration example

Let's configure BGP instance, peer and networks advertised on R1

```
[admin@R1] > routing bgp instance set default router-id=1.1.1.1 as=1

[admin@R1] > routing bgp peer add remote-address=2001:12::2 remote-as=2
               address-families=ipv6

[admin@R1] > routing bgp network add network=2001:1::/64
```

# BGP for IPv6 configuration example

Let's configure BGP instance, peers and networks advertised on R2

```
[admin@R2] > routing bgp instance set default router-id=2.2.2.2 as=2

[admin@R2] > routing bgp peer add remote-address=2001:12::1 remote-as=1
             address-families=ipv6
[admin@R2] > routing bgp peer add remote-address=2001:23::3 remote-as=2
             address-families=ipv6

[admin@R2] > routing bgp network add network=2001:23::/64
[admin@R2] > routing bgp network add network=2001:2::/64
```

# BGP for IPv6 configuration example

Let's configure BGP instance, peer and networks advertised on R3

```
[admin@R3] > routing bgp instance set default router-id=3.3.3.3 as=2

[admin@R3] > routing bgp peer add remote-address=2001:23::2 remote-as=2
             address-families=ipv6

[admin@R3] > routing bgp network add network=2001:23::/64
[admin@R3] > routing bgp network add network=2001:3::/64
```

# BGP for IPv6 configuration example

Routing table on R1

```
[admin@R1] > ipv6 route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip,
 o - ospf, b - bgp,  U - unreachable
 #      DST-ADDRESS             GATEWAY                 DISTANCE
 0 ADC  2001:1::/64             loopback0               0
 1 ADb  2001:2::/64             fe80::20c:42ff:fe0e:f... 20
 2 ADC  2001:12::/64            ether1                  0
 3 ADb  2001:23::/64            fe80::20c:42ff:fe0e:f... 20/64
```

# BGP for IPv6 configuration example

Routing table on R2

```
[admin@R2] > ipv6 route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip,
 o - ospf, b - bgp,  U - unreachable
 #      DST-ADDRESS              GATEWAY                  DISTANCE
 0 ADb  2001:1::/64              fe80::20c:42ff:fe3e:f... 20
 1 ADC  2001:2::/64              loopback0                0
 2 ADo  2001:3::/64              fe80::20c:42ff:fe07:d... 110
 3  Dr  2001:3::/64              fe80::20c:42ff:fe07:d... 120
 4  Db  2001:3::/64              fe80::20c:42ff:fe07:d... 200
 5 ADC  2001:12::/64             ether1                   0
 6 ADC  2001:23::/64             ether2                   0
 7  Db  2001:23::/64             fe80::20c:42ff:fe07:d... 200
```

# BGP for IPv6 configuration example

Routing table on R3

```
[admin@R3] > ipv6 route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip,
 o - ospf, b - bgp,  U - unreachable
 #      DST-ADDRESS              GATEWAY                   DISTANCE
 0 ADb  2001:1::/64              fe80::20c:42ff:fe0e:f... 200
 1 ADo  2001:2::/64              fe80::20c:42ff:fe0e:f... 110
 2  Dr  2001:2::/64              fe80::20c:42ff:fe0e:f... 120
 3  Db  2001:2::/64              fe80::20c:42ff:fe0e:f... 200
 4 ADC  2001:3::/64              loopback0                0
 5 ADC  2001:23::/64             ether2                   0
 6  Db  2001:23::/64             fe80::20c:42ff:fe0e:f... 200
```

# How to start

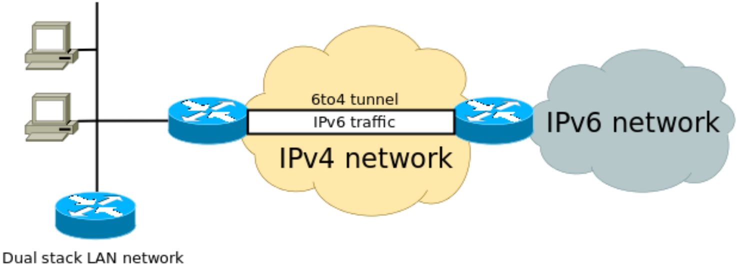
# How to connect to IPv6 Internet

There are two main possibilities to connect to the IPv6 Internet:

- Native IPv6 connectivity provided by our ISP
- Tunnel to the PoP (Point of Presence)

We will focus on the second case.

# Tunneling over IPv4



# Tunnel brocker and PoP

First, we need the Point of Presence which will be our far tunnel endpoint. We can use one of the Tunnel Brokers - organizations who take care on tunnel assigning. We use, as an example SixxS. You can find it on: http://www.sixxs.net/

# The procedure

- Account creation

# The procedure

- Account creation
- Tunnel request - when you request the tunnel you need to shortly explain why you need it. There are few types of tunnels - probably the widely used it static tunnel (you need a public IPv4 address)

## The procedure

- Account creation
- Tunnel request - when you request the tunnel you need to shortly explain why you need it. There are few types of tunnels - probably the widely used it static tunnel (you need a public IPv4 address)
- After tunnel approval you need to establish 6to4 (sit) tunnel to the IPv4 address provided by SixxS and configure the IPv6 address. Far endpoint will be pinging your router to check tunnel availability

# The procedure

- Account creation
- Tunnel request - when you request the tunnel you need to shortly explain why you need it. There are few types of tunnels - probably the widely used it static tunnel (you need a public IPv4 address)
- After tunnel approval you need to establish 6to4 (sit) tunnel to the IPv4 address provided by SixxS and configure the IPv6 address. Far endpoint will be pinging your router to check tunnel availability
- After a week you can request the /48 network. Again you need to write short explanation.

# Configuration

We need to configure the tunel to the PoP

```
[admin@R1] > interface 6to4 add  local-address=91.224.142.4 \
         remote-address=193.219.28.26 name=sit1
```

# Configuration

Put the IPv6 address received from Sixxs to the sit1 tunnel

```
[admin@R1] > ipv6 address add address=2001:6a0:200:bd::2/64 \
         eui-64=no interface=sit1
```

# Configuration

Create the default route to the address provided from Sixxs

```
[admin@R1] > ipv6 route add dst-address=::/0 gateway=2001:6a0:200:bd::1
```

# Configuration

When you are assigned the IPv6 address space, configure your LAN

```
[admin@R1] > ipv6 address add address=2001:6a0:176:1::1/64 interface=ether3
```

# Questions ?

Any questions?

# Questions ?

Thank you!