Certified IPv6 Engineer (MTCIPv6E)

Riga, Latvia

June 23 - June 24, 2016

MTCIPv6E

About the Trainer

- Name
- Experience
- …

Your photo

Course Objectives

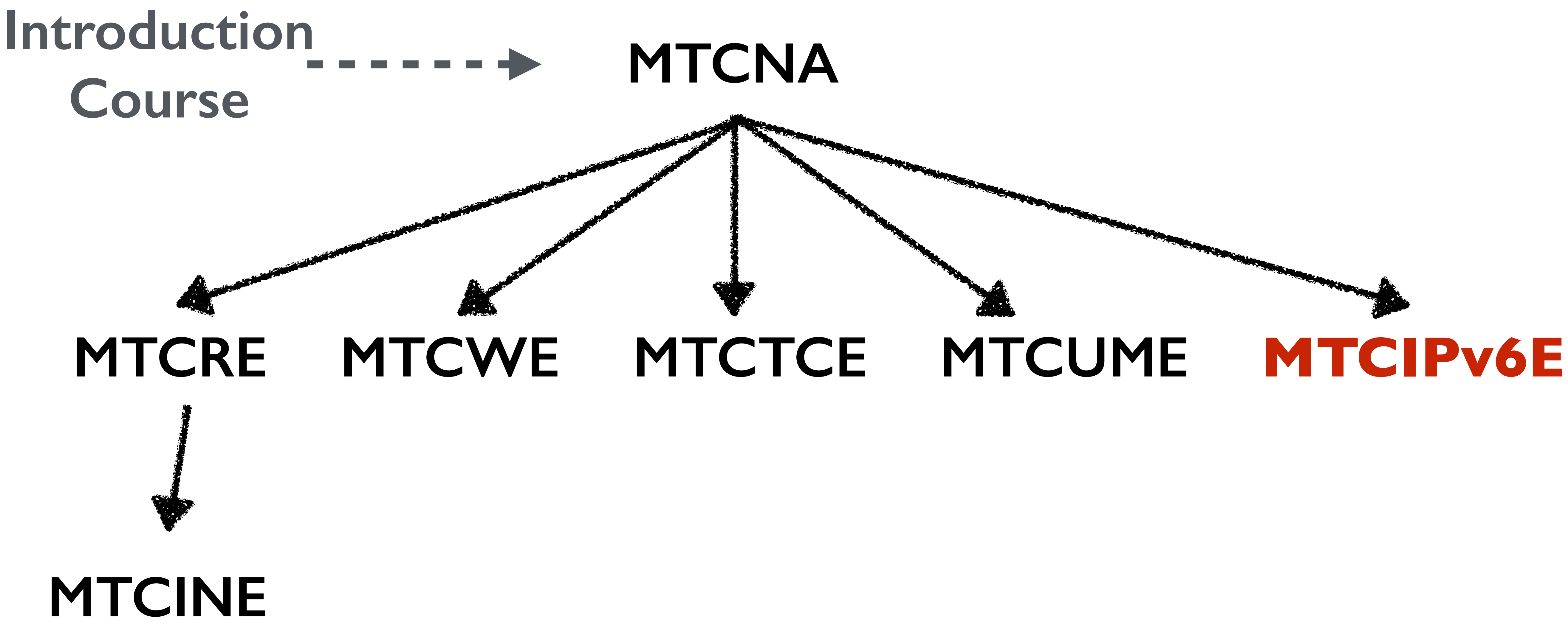
- Provide an overview of IPv6, most common transition mechanisms and how to implement it on RouterOS
- Hands-on training for MikroTik RouterOS IPv6 configuration, maintenance and troubleshooting

Learning Outcomes

The student will:

- Be able to configure, manage and do basic troubleshooting of an IPv6 network on a MikroTik RouterOS device
- Be able to provide IPv6 services to clients
- Have a solid foundation and valuable tools to manage an IPv6 network

MikroTik Certified Courses



Introduction Course

MTCNA

MTCRE

MTCWE

MTCTCE

MTCUME

MTCIPv6E

MTCINE

For more info see: training.mikrotik.com

MTCIPv6E Outline

- Module 1: Introduction to IPv6
- Module 2: IPv6 Protocol
- Module 3: IPv6 Packet
- Module 4: IPv6 Security
- Module 5: Transition Mechanisms
- Module 6: Interoperability

Schedule

- Training day: 9AM - 5PM
- 30 minute breaks: 10:30AM and 3PM
- 1 hour lunch: 12:30PM
- Certification test: last day, 1 hour

Housekeeping

- Emergency exits
- Bathroom location
- Food and drinks while in class
- Please set phone to 'silence' and take calls outside the classroom

Introduce Yourself

- Your name and company
- Your prior knowledge about IPv6 networking
- Your prior knowledge about IPv6 in RouterOS
- What do you expect from this course?
- Please, note your number (XY): ____

Certified IPv6 Engineer (MTCIPv6E)

Module 0

Recap from MTCNA

MTCIPv6E

About MikroTik

- Router software and hardware manufacturer
- Products used by ISPs, companies and individuals
- Mission: to make Internet technologies faster, more powerful and affordable to a wider range of users

About MikroTik

- 1996: Established
- 1997: RouterOS software for x86 (PC)
- 2002: First RouterBOARD device
- 2006: First MikroTik User Meeting (MUM)
    - Prague, Czech Republic
- 2015: Biggest MUM: Indonesia, 2500+

About MikroTik

- Located in Latvia
- 160+ employees
- mikrotik.com
- routerboard.com

MikroTik RouterOS

- Is the operating system of MikroTik RouterBOARD hardware
- Can also be installed on a PC or as a virtual machine (VM)
- Stand-alone operating system based on the Linux kernel

RouterOS Features

- IPv6 support
- Full 802.11 a/b/g/n/ac support
- Firewall/bandwidth shaping
- Point-to-Point tunnelling (PPTP, PPPoE, SSTP, OpenVPN), DHCP/Proxy/HotSpot
- And many more… see: wiki.mikrotik.com

MikroTik RouterBOARD

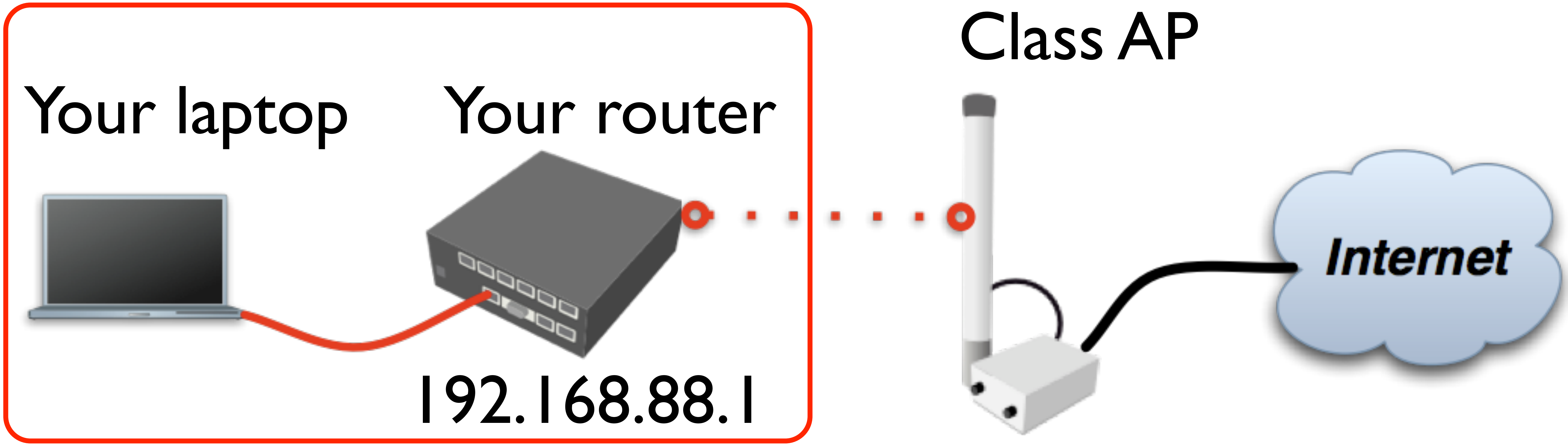
- A family of hardware solutions created by MikroTik that run RouterOS
- Ranging from small home routers to carrier-class access concentrators
- Millions of RouterBOARDs are currently routing the world

MikroTik RouterBOARD

- Integrated solutions - ready to use
- Boards only - for assembling own system
- Enclosures - for custom RouterBOARD builds
- Interfaces - for expanding functionality
- Accessories

LAB

Internet Access



Laptop - Router

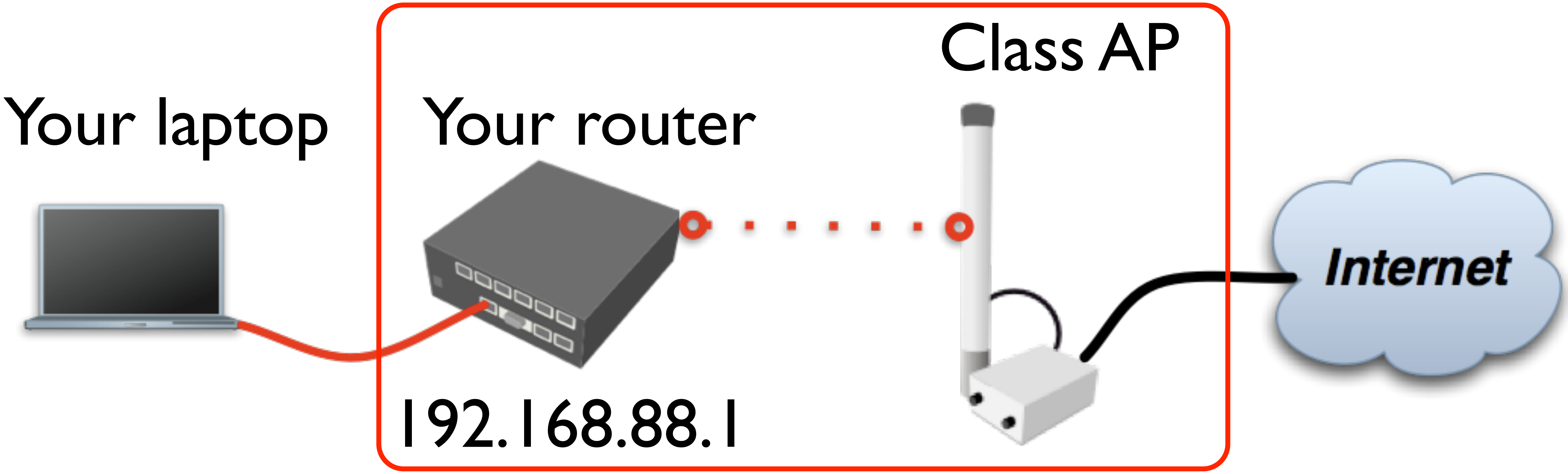
LAB

- Connect laptop to the router with a cable, plug it in any of LAN ports (2-5)
- Disable other interfaces (wireless) on your laptop
- Make sure that Ethernet interface is set to obtain IP configuration automatically (via DHCP)

MTCIPv6E

Router - Internet

- The Internet gateway of your class is accessible over wireless - it is an access point **(AP)**



LAB

Router - Internet

- To connect to the AP you have to:
  - Remove the wireless interface from the bridge interface (used in default configuration)
  - Configure **DHCP client** to the wireless interface

LAB

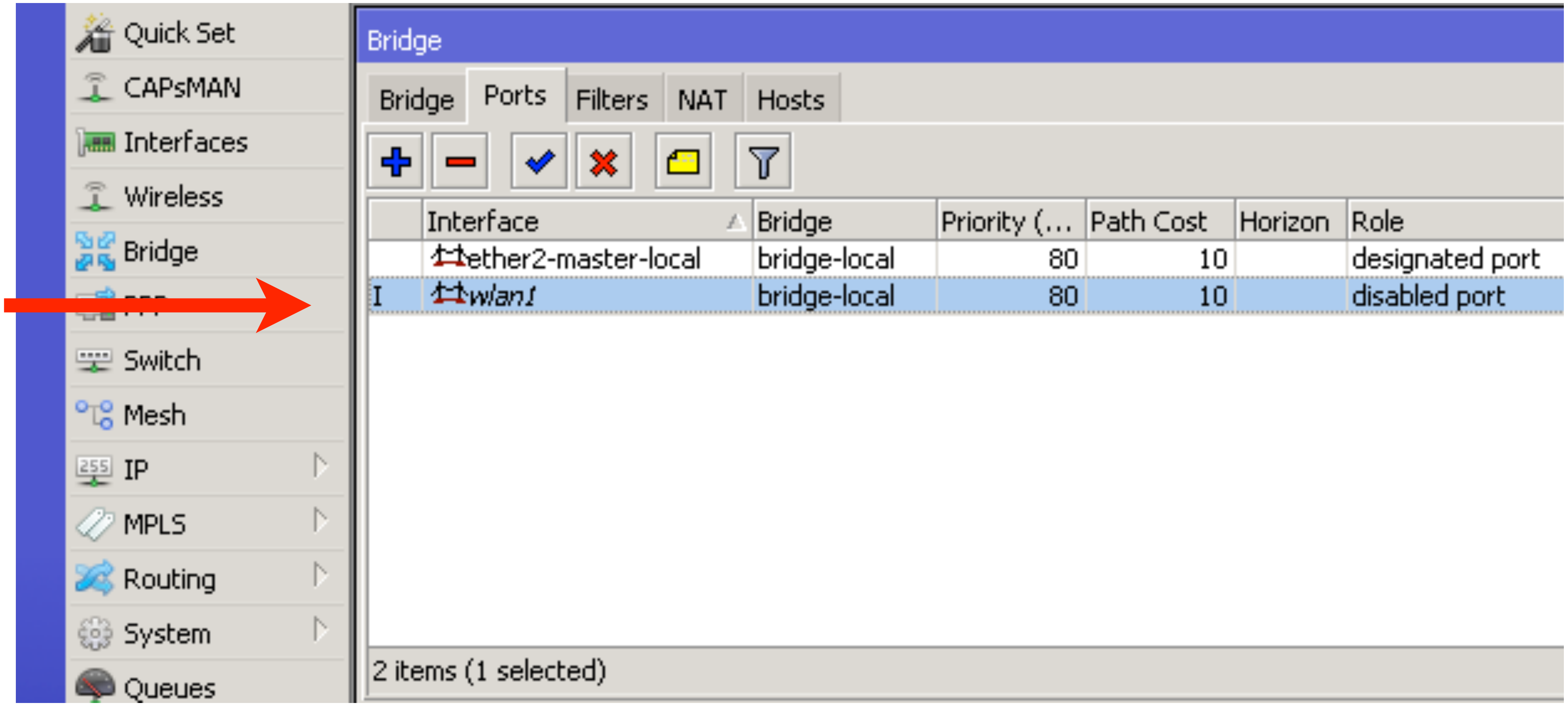
Router - Internet

- To connect to the AP you have to:
  - Create and configure a wireless **security profile**
  - Set the wireless interface to **station** mode
  - And configure **NAT masquerade**

LAB

Router - Internet

Remove the WiFi interface from the bridge



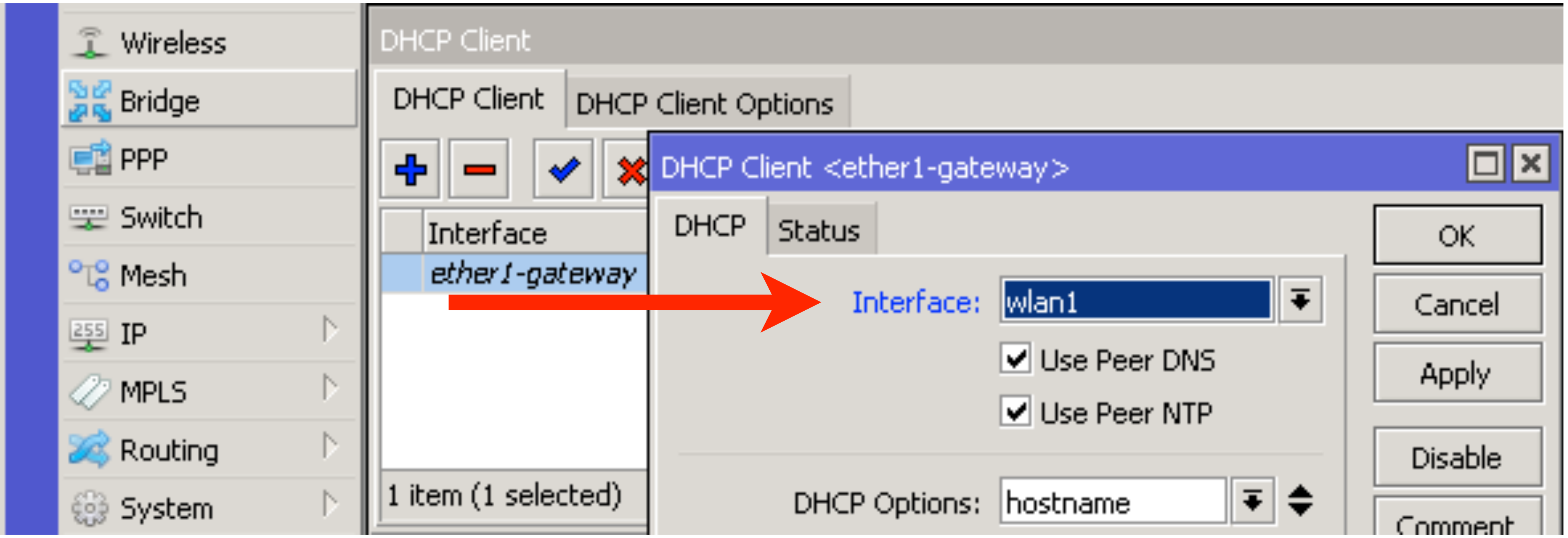
	Interface	Bridge	Priority (...	Path Cost	Horizon	Role
	ether2-master-local	bridge-local	80	10		designated port
I	*wlan1*	bridge-local	80	10		disabled port

2 items (1 selected)

Bridge → Ports

Router - Internet

Set DHCP client to the WiFi interface



Interface: wlan1

☑ Use Peer DNS

☑ Use Peer NTP

DHCP Options: hostname

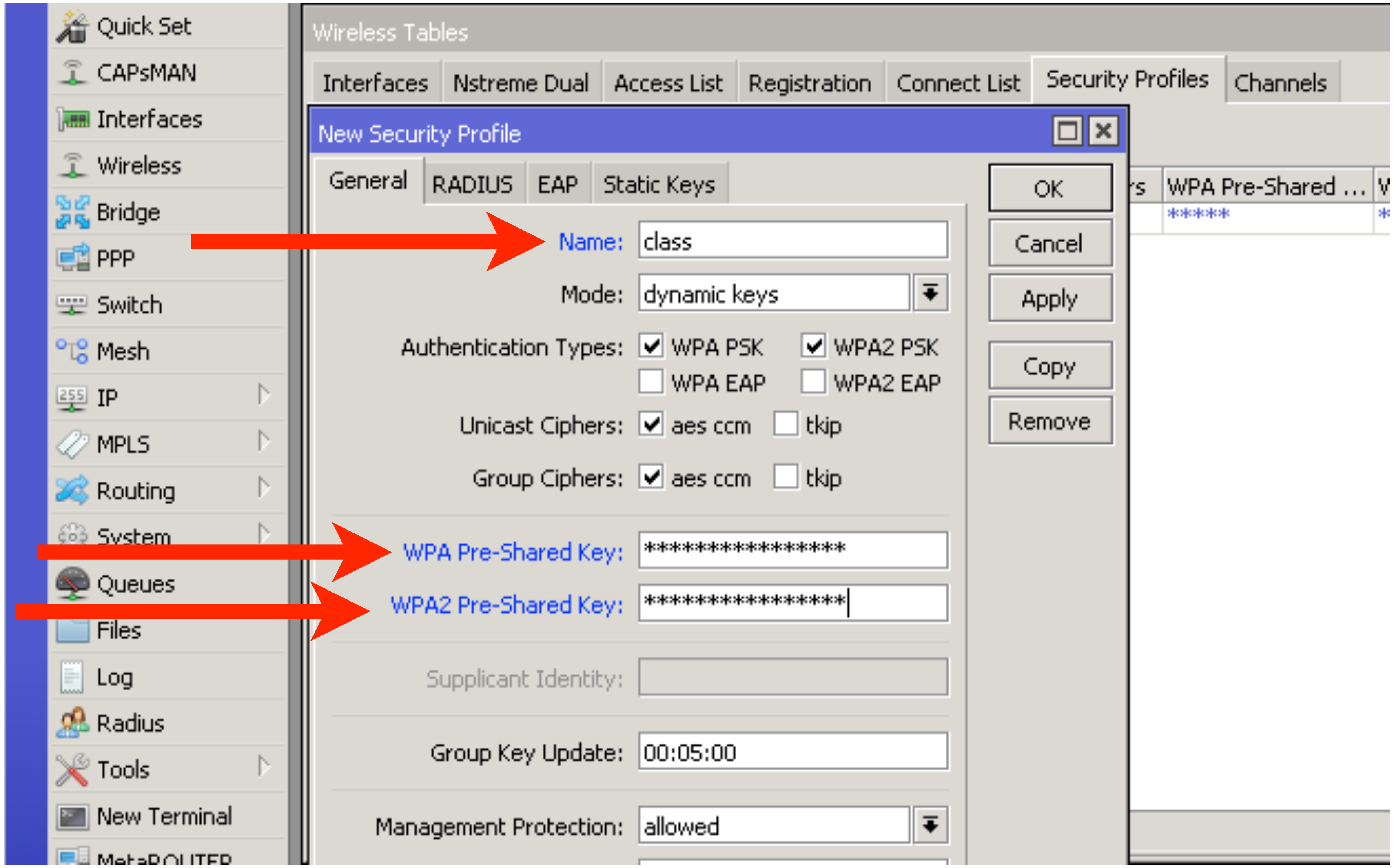
IP → DHCP Client

MikroTik
MTCIPv6E

LAB

Router - Internet

Set Name and Pre-Shared Keys



Name: class

Mode: dynamic keys

Authentication Types: ☑ WPA PSK ☑ WPA2 PSK ☐ WPA EAP ☐ WPA2 EAP

Unicast Ciphers: ☑ aes ccm ☐ tkip

Group Ciphers: ☑ aes ccm ☐ tkip

WPA Pre-Shared Key: ***************

WPA2 Pre-Shared Key: ***************

Supplicant Identity:

Group Key Update: 00:05:00

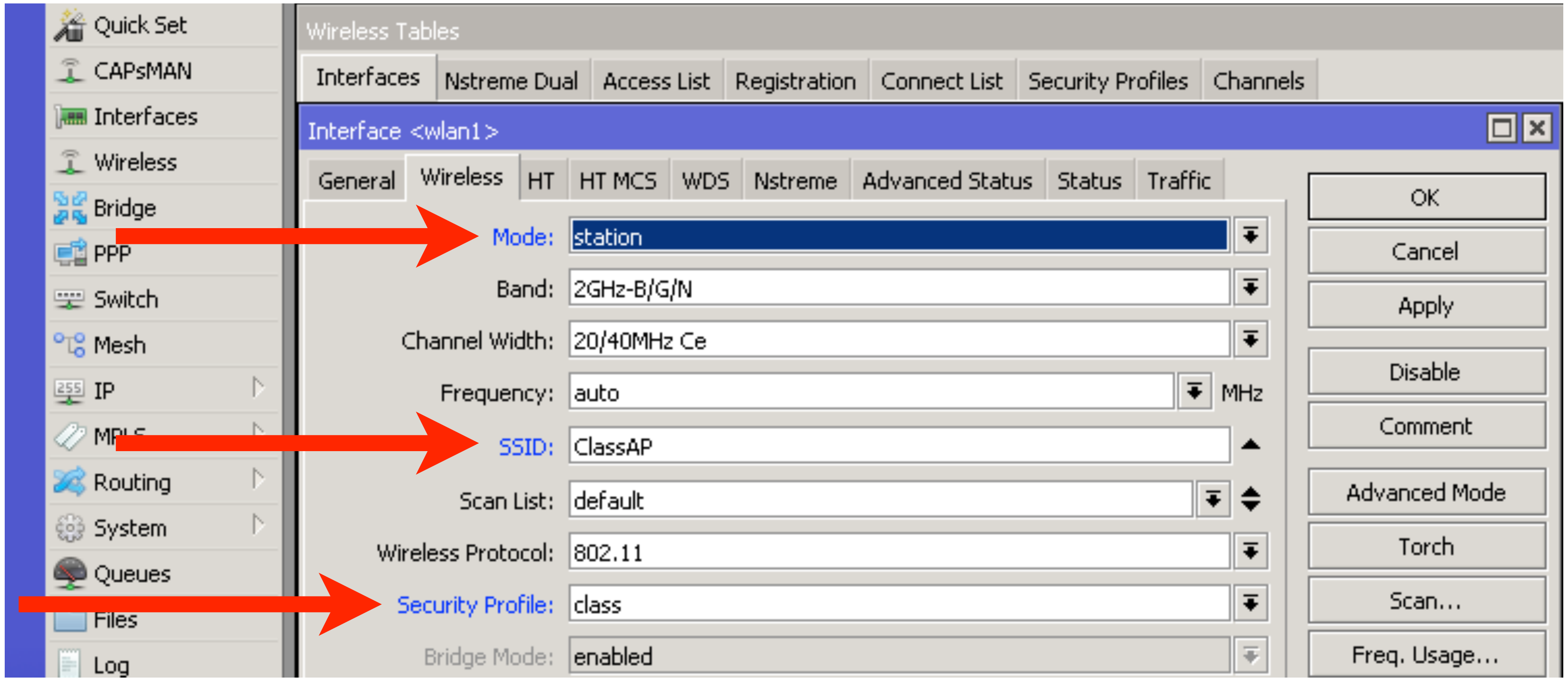
Management Protection: allowed

Wireless → Security Profiles

LAB

Router - Internet

Set Mode to ‘station', SSID to 'ClassAP' and Security Profile to 'class'



Mode: station

Band: 2GHz-B/G/N

Channel Width: 20/40MHz Ce

Frequency: auto MHz

SSID: ClassAP

Scan List: default

Wireless Protocol: 802.11

Security Profile: class

Bridge Mode: enabled

Wireless → Interfaces

- “Scan…” tool can be used to see and connect to available APs

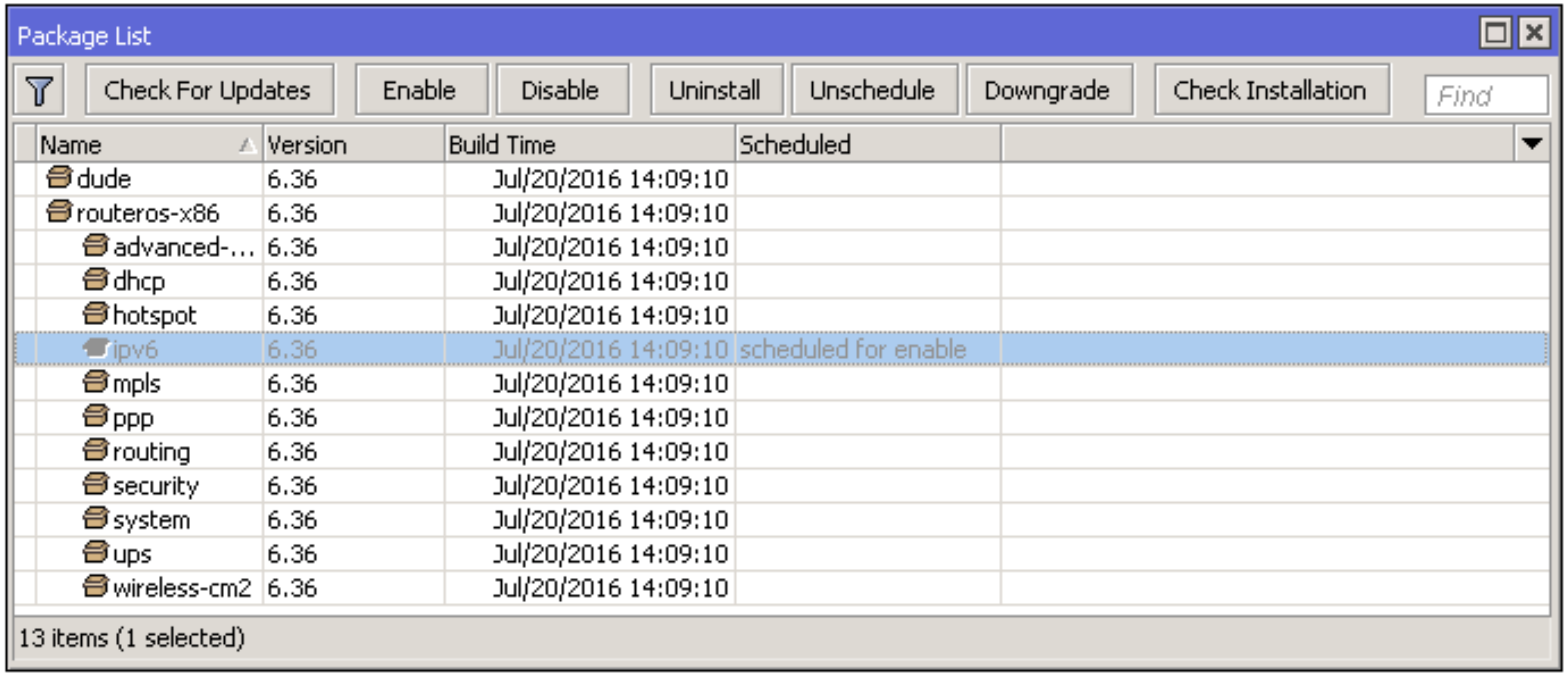
MikroTik
MTCIPv6E

IPv6 on RouterOS

- IPv6 support is not enabled by default
- The package is included
- To enable go to System → Packages
- Select ‘ipv6’ and click Enable
- Reboot the router
- New menu ‘IPv6’ will appear in WinBox

IPv6 on RouterOS

- RouterOS functions are enabled/disabled by packages. Enable ‘ipv6’ and reboot



Name	Version	Build Time	Scheduled
dude	6.36	Jul/20/2016 14:09:10	
routeros-x86	6.36	Jul/20/2016 14:09:10	
advanced-...	6.36	Jul/20/2016 14:09:10	
dhcp	6.36	Jul/20/2016 14:09:10	
hotspot	6.36	Jul/20/2016 14:09:10	
ipv6	6.36	Jul/20/2016 14:09:10	scheduled for enable
mpls	6.36	Jul/20/2016 14:09:10	
ppp	6.36	Jul/20/2016 14:09:10	
routing	6.36	Jul/20/2016 14:09:10	
security	6.36	Jul/20/2016 14:09:10	
system	6.36	Jul/20/2016 14:09:10	
ups	6.36	Jul/20/2016 14:09:10	
wireless-cm2	6.36	Jul/20/2016 14:09:10	

13 items (1 selected)

System → Packages

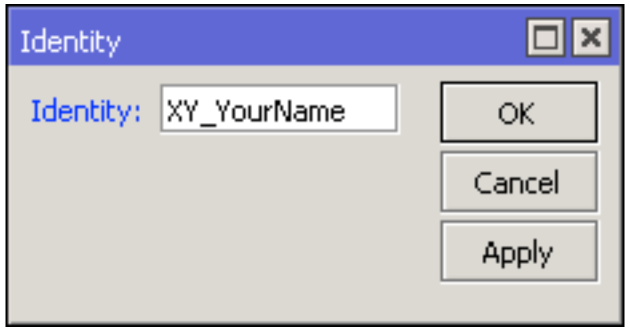
RouterOS Packages

Package	Functionality
advanced-tools	Netwatch, wake-on-LAN
dhcp	DHCP client and server
hotspot	HotSpot captive portal server
ipv6	IPv6 support
ppp	PPP, PPTP, L2TP, PPPoE clients and servers
routing	Dynamic routing: RIP, BGP, OSPF
security	Secure WinBox, SSH, IPsec
system	Basic features: static routing, firewall, bridging, etc.
wireless	802.11 a/b/g/n/ac support, CAPsMAN v2, repeater

- For more info see packages wiki page

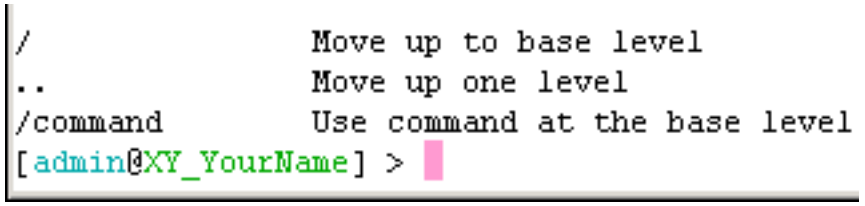
Router Identity

- Option to set a name for each router
- Identity information available in different places

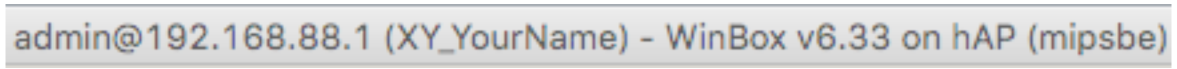


Identity: XY_YourName

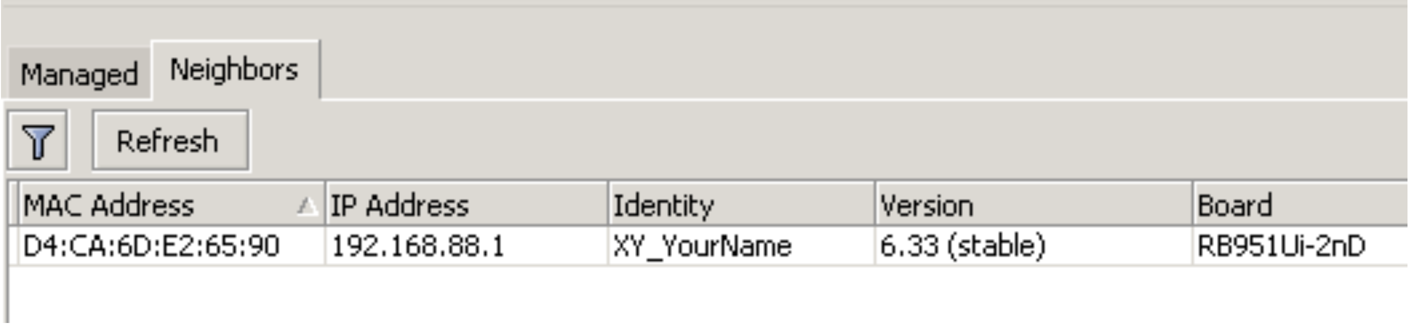
System → Identity



```
/                 Move up to base level
..                Move up one level
/command          Use command at the base level
[admin@XY_YourName] >
```



admin@192.168.88.1 (XY_YourName) - WinBox v6.33 on hAP (mipsbe)



MAC Address	IP Address	Identity	Version	Board
D4:CA:6D:E2:65:90	192.168.88.1	XY_YourName	6.33 (stable)	RB951Ui-2nD

LAB

Router Identity

- Set the identity of your router as follows: **YourNumber(XY)_YourName**
- For example: **13_JohnDoe**
- Observe the WinBox title menu

Additional Information

- wiki.mikrotik.com - RouterOS documentation and examples
- forum.mikrotik.com - communicate with other RouterOS users
- mum.mikrotik.com - MikroTik User Meeting page
- Distributor and consultant support
- support@mikrotik.com

Certified IPv6 Engineer (MTCIPv6E)

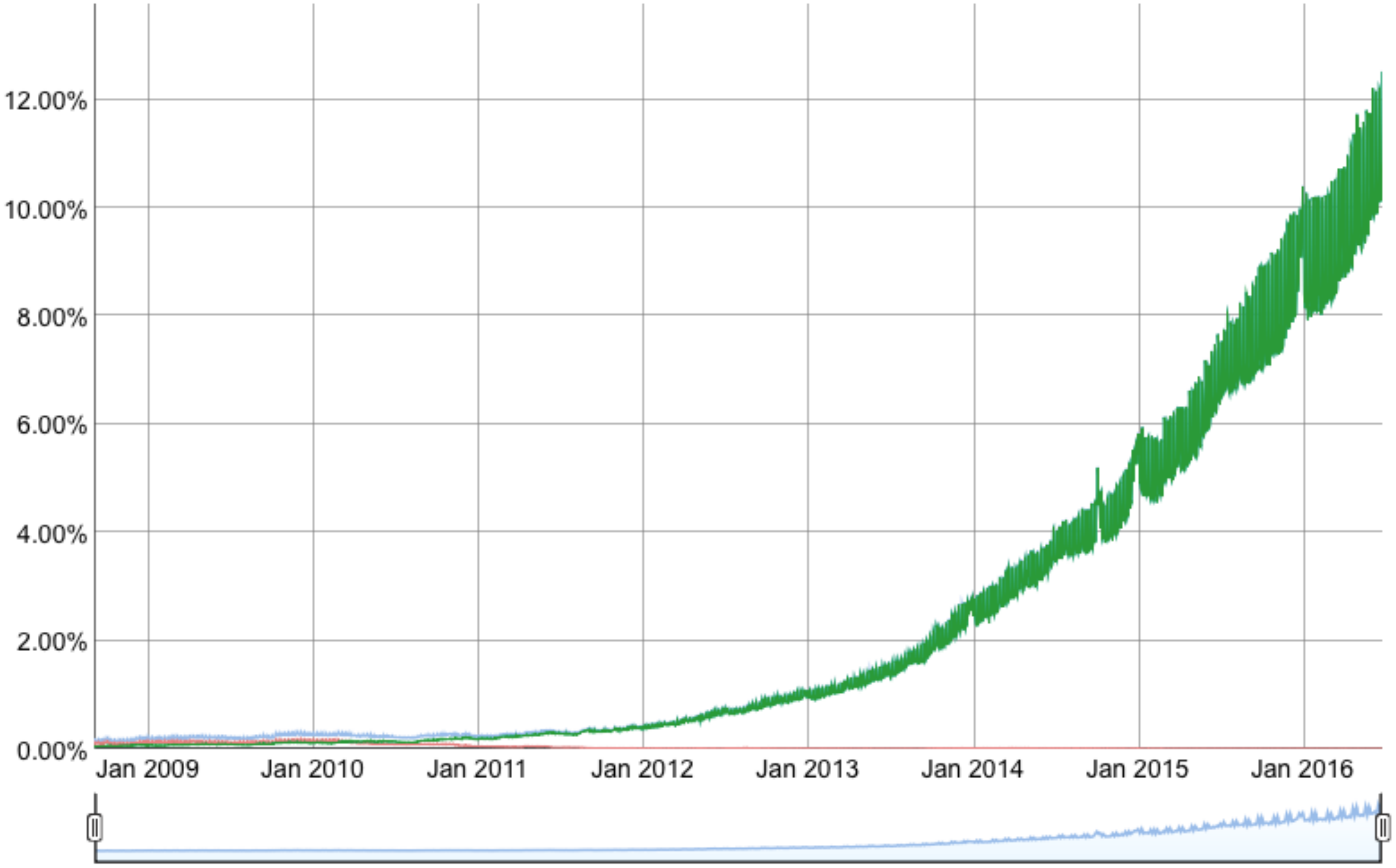
Module 1

Introduction to IPv6

IPv6

- Internet Protocol version 6
- Designed as the successor to IPv4
- Development started in 1996
- First IPv6 specification in 1998 (RFC 2460)

IPv6 Adoption



Current numbers according to Google can be seen here

Comparison

	IPv4	IPv6
Address space	32 bits	128 bits
Possible addresses	2^{32}	2^{128}
Address format	192.0.2.1	2001:db8:3:4:5:6:7:8
Header length	20bytes	40bytes
Header fields	14	8
IPsec	optional	SHOULD*

IPsec on IPv6

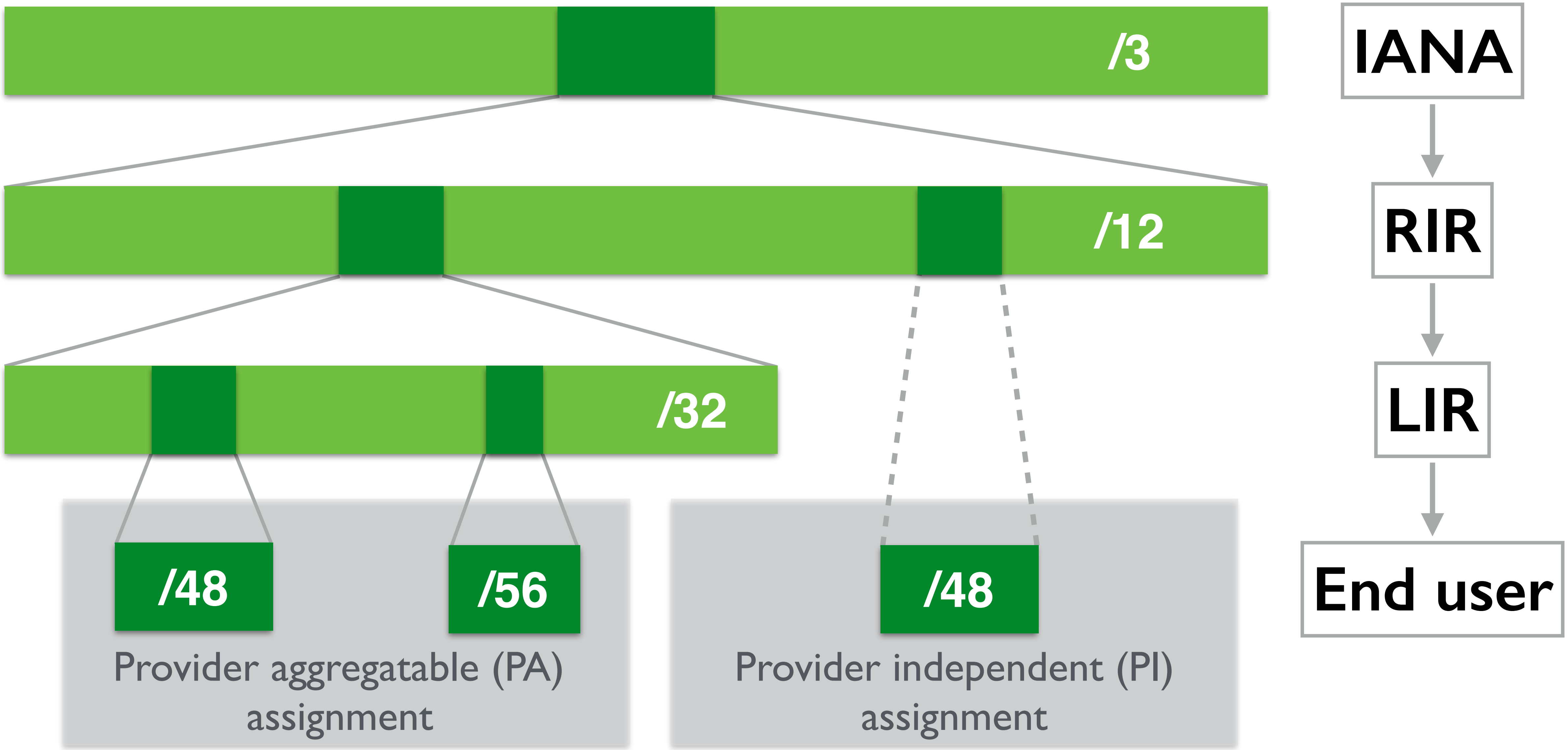
- IPv6 Node Requirements (RFC6434) states that all IPv6 nodes SHOULD support IPsec

SHOULD - means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course

Terminology

- **node** - a device that implements Internet protocol (IP)
- **router** - a node that forwards IP packets not explicitly addressed to itself
- **host** - any node that is not a router
- RFC4861 - Terminology

Address Distribution


/3
IANA
/12
RIR
/32
LIR
/48
/56
/48
End user
Provider aggregatable (PA) assignment
Provider independent (PI) assignment

Source: RIPE NCC

Address Notation

- IPv6 consists of 8 fields each 16 bits long
- Written in hexadecimal numerals (base 16)
- Separated by a colon “:”

2001:0db8:1234:5678:9abc:def0:1234:5678

Address Notation

Field (16 bits)	Hexadecimal	Binary
1	2001	0010 0000 0000 0001
2	0db8	0000 1101 1011 1000
3	0be0	0000 1011 1110 0000
4	75a1	0111 0101 1010 0001
5	0000	0000 0000 0000 0000
6	0000	0000 0000 0000 0000
7	0000	0000 0000 0000 0000
8	0001	0000 0000 0000 0001

2001:0db8:0be0:75a1:0000:0000:0000:0001

Address Notation

2001:0db8:0be0:75a2:0000:0000:0000:0001

Leading zeros can be left out

2001:db8:be0:75a2:0:0:0:1

Consecutive fields of zeros can be replaced with ::

2001:db8:be0:75a2::1

Address Notation

2001:0db8:0000:0000:0010:0000:0000:0001

If there are several consecutive fields of zeros only one can be replaced with ::

2001:db8::10:0:0:1

You can choose which one

2001:db8:0:0:10::1

The same IP address. Both notations are valid but the first one is recommended

For more info see "A Recommendation for IPv6 Address Text Representation (RFC5952)"

LAB

Address Notation

Compress the following IPv6 addresses
to shortest form possible

2001:0db8:0ab0:0d00:0000:0000:0000:0c01

2001:0db8:0000:4c05:0000:0000:05ad:0bb1

2001:0db8:0000:0000:1234:0000:0000:da61

Answers are on the next slide

MikroTik
MTCIPv6E

LAB

Address Notation

2001:db8:ab0:d00::c01

2001:db8:0:4c05::5ad:bb1

2001:db8::1234:0:0:da61

or 2001:db8:0:0:1234::da61

LAB

Address Notation

Expand the following IPv6 addresses
to full notation

2001:db8:ab::bc0:c1ab

2001:db8:a000:c05:b0::1

2001:db8:0:1234::61

Answers are on the next slide

MTCIPv6E

LAB

Address Notation

2001:0db8:00ab:0000:0000:0000:0bc0:c1ab

2001:0db8:a000:0c05:00b0:0000:0000:0001

2001:0db8:0000:1234:0000:0000:0000:0061

EUI-64

- 64-bit extended unique identifier (EUI)
- Derived from 48-bit MAC address

00:0c:29:0c:47:d5

+ ff:fe

00:0c:29:ff:fe:0c:47:d5

Modified EUI-64

- Used in stateless address autoconfiguration (SLAAC)
- 7th bit from the left, the universal/local (U/L) bit, needs to be inverted

00 (L) → 02 (U)

02:0c:29:ff:fe:0c:47:d5

Modified EUI-64

IPv6 prefix

2001:db8:be0:75a2::/64

and modified EUI-64 from MAC address

02:0c:29:ff:fe:0c:47:d5

Results in the following IPv6 address

2001:db8:be0:75a2:020c:29ff:fe0c:47d5

SLAAC Address Construction

Routing prefix	Subnet	Interface identifier
0-64 bits	0-64 bits	64 bits

- Routing prefix + subnet identifier = 64 bits
- /64 is the smallest prefix that can be assigned to a customer
- Usually a customer is assigned /48 - /64 subnet

Subnetting

2001:0db8:0be0:	75a2:	0000:0000:0000:0001
Routing prefix: 48 bits	Subnet: 16	65536 x /64

2001:0db8:0be0:7	5a2:	0000:0000:0000:0001
Routing prefix: 52 bits	12	4096 x /64

2001:0db8:0be0:75	a2:	0000:0000:0000:0001
Routing prefix: 56 bits	8	256 x /64

2001:0db8:0be0:75a	2:	0000:0000:0000:0001
Routing prefix: 60 bits	4	16 x /64

Address Types

Type	Range
Link local	fe80::/10
Global unicast	2000::/3
Multicast	ff00::/8
Unique local	fc00::/7

Special Addresses

Type	Range
Loobpack	::1/128
Documentation	2001:db8::/32
6to4	2002::/16
Unspecified address	::/128
Teredo	2001::/32
Anycast	2001:db8:db1b:1e3::/64

Unique Local Address

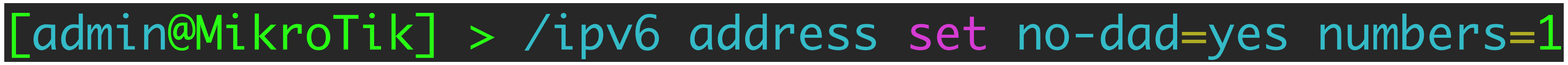
- Meant to never be used on the Internet
- fc00::/7 prefix is reserved for ULA
- Divided into fc00::/8 and fd00::/8
- fd00::/8 currently is the only valid ULA prefix
  - fc00::/8 prefix has not been defined

Anycast Address

- Multiple hosts can have the same anycast address
- Send to any one member of this group (usually the nearest)
- Indistinguishable from a unicast address

Anycast Address

- Use cases: load balancing, content delivery networks (CDN)
- When using anycast address, Duplicate Address Detection has to be disabled for that IP



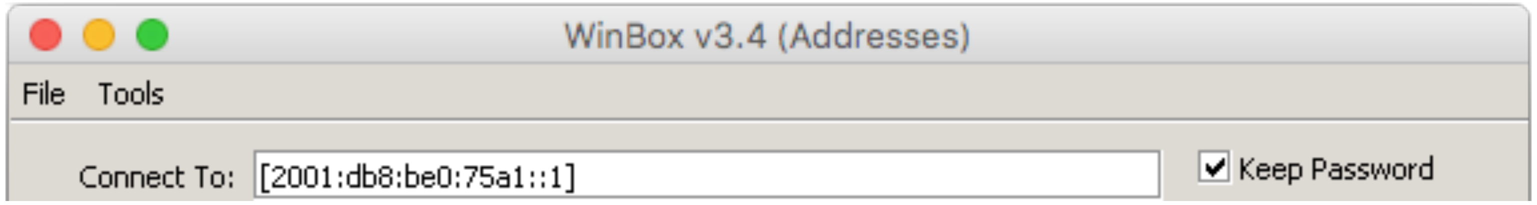
```
[admin@MikroTik] > /ipv6 address set no-dad=yes numbers=1
```

IPv4-mapped IPv6 address

- IPv6 address that holds an embedded IPv4 address
- Is used to represent the addresses of IPv4 nodes as IPv6 addresses

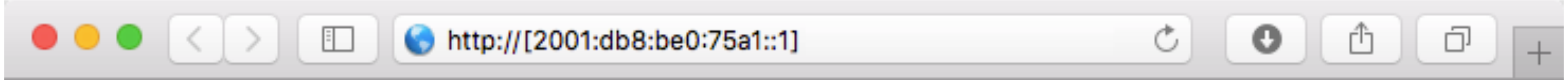
IPv4 address	IPv4-mapped IPv6 address
192.0.2.123	::ffff:192.0.2.123
	::ffff:c000:027b

Connecting to Global IPv6 host

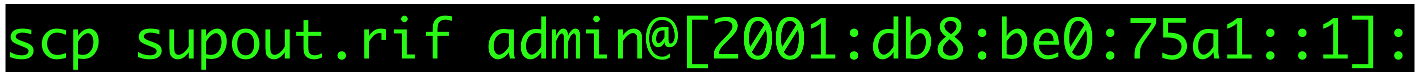


Connect To: [2001:db8:be0:75a1::1]

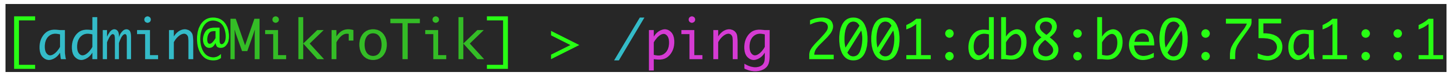
☑ Keep Password



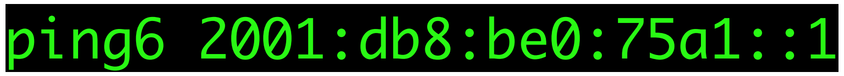
http://[2001:db8:be0:75a1::1]



```
scp supout.rif admin@[2001:db8:be0:75a1::1]:
```



```
[admin@MikroTik] > /ping 2001:db8:be0:75a1::1
```



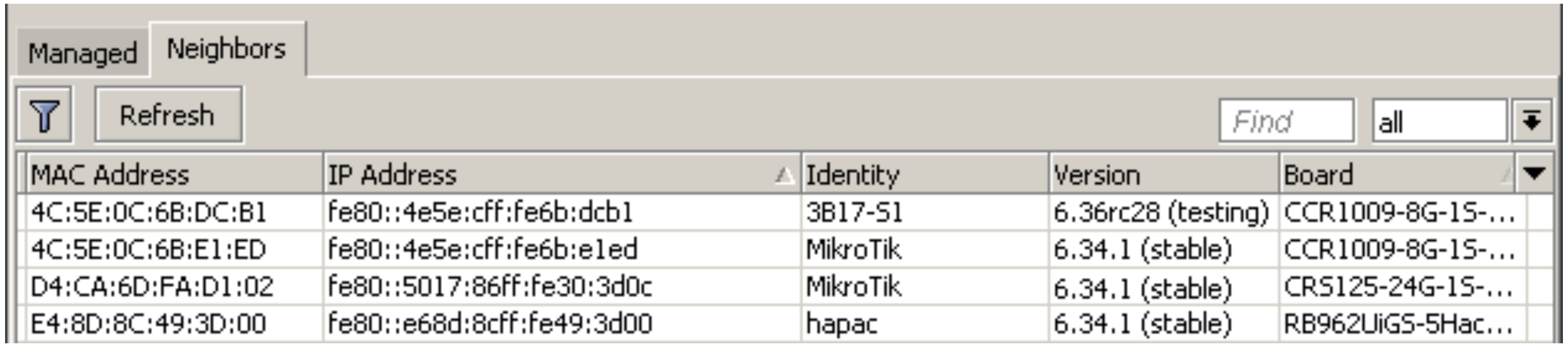
```
ping6 2001:db8:be0:75a1::1
```

Depending on the context IPv6 address is written with or without brackets

IPv6 Connectivity

- Link-local address can be used to connect when the device has no globally routed IPv6 address

- Alternative to MAC WinBox



MAC Address	IP Address	Identity	Version	Board
4C:5E:0C:6B:DC:B1	fe80::4e5e:cff:fe6b:dcb1	3B17-S1	6.36rc28 (testing)	CCR1009-8G-1S-...
4C:5E:0C:6B:E1:ED	fe80::4e5e:cff:fe6b:e1ed	MikroTik	6.34.1 (stable)	CCR1009-8G-1S-...
D4:CA:6D:FA:D1:02	fe80::5017:86ff:fe30:3d0c	MikroTik	6.34.1 (stable)	CRS125-24G-1S-...
E4:8D:8C:49:3D:00	fe80::e68d:8cff:fe49:3d00	hapac	6.34.1 (stable)	RB962UiGS-5Hac...

WinBox → Neighbors

Module 1 Summary

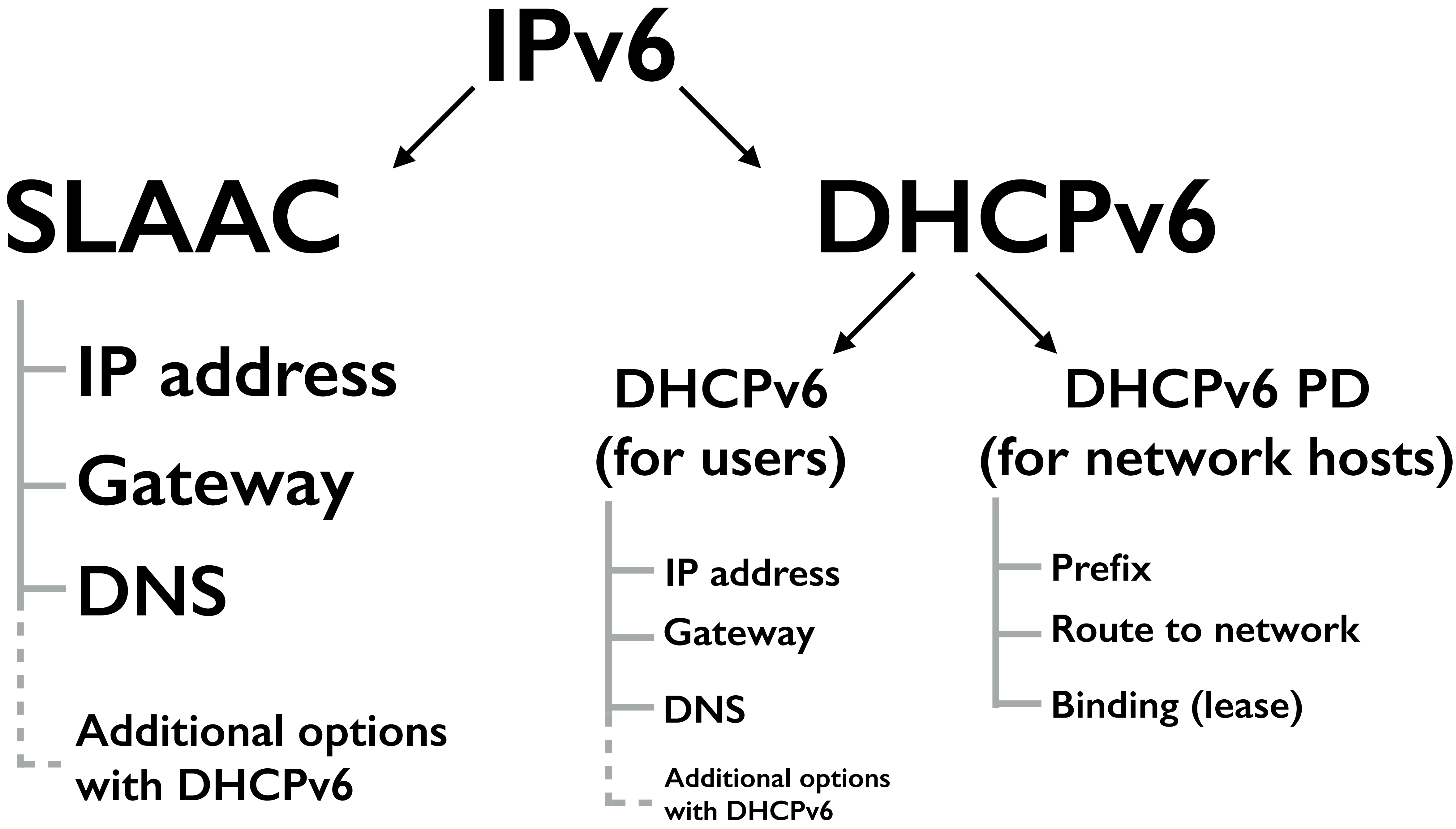
Certified IPv6 Engineer (MTCIPv6E)

Module 2

IPv6 Protocol

Address Configuration

- Auto configuration of link local address
- Stateless
  - Stateless address autoconfiguration (SLAAC)
  - Additional options with DHCPv6
- Stateful
  - DHCPv6
- Static



IPv6

SLAAC

- IP address
- Gateway
- DNS
- Additional options with DHCPv6

DHCPv6

DHCPv6 (for users)

- IP address
- Gateway
- DNS
- Additional options with DHCPv6

DHCPv6 PD (for network hosts)

- Prefix
- Route to network
- Binding (lease)

Neighbor Discovery

- Neighbor discovery (ND) protocol
- Replaces ARP on IPv4
- Tracks and discovers other IPv6 hosts
- Auto-configures address
- Uses ICMPv6 protocol

Neighbor Discovery

- Has 5 message types:
  - Router solicitation (type 133)
  - Router advertisement (type 134)
  - Neighbor solicitation (type 135)
  - Neighbor advertisement (type 136)
  - Redirect (type 137)

Link Local

- 1st step is to generate link local (LL) address

fe80:: + **Interface ID (Modified EUI-64)**

- 2nd: perform ‘neighbor solicitation’

A: This is my IPv6 address, is this in use? What’s your MAC address?

- 3rd: ‘neighbor advertisement’

B: Yes, I’m using this address. My MAC is 12:34:56:78:90:12

- If nobody answers, host uses generated LL address

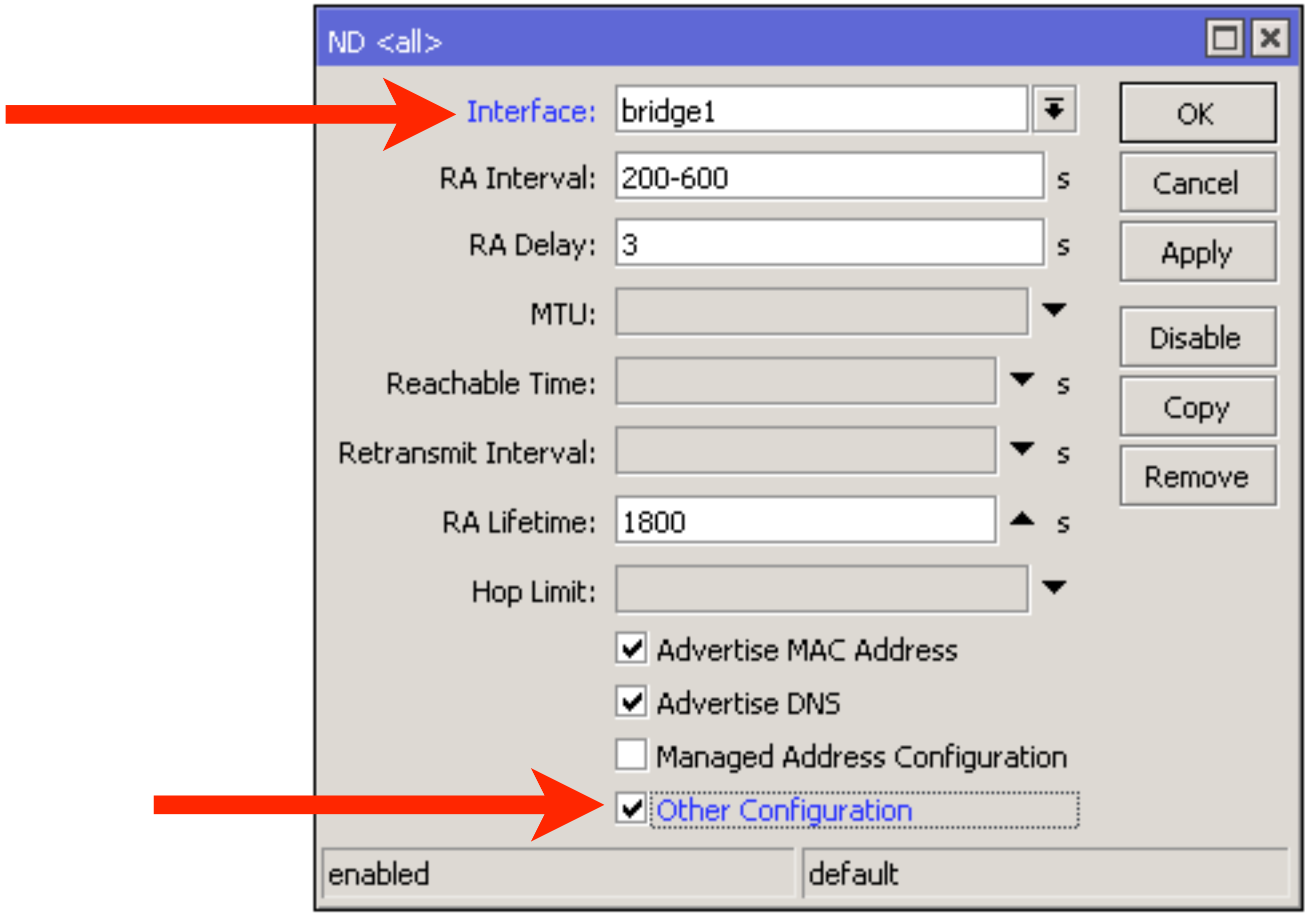
SLAAC

- Stateless address autoconfiguration
- Uses router solicitation and router advertisement messages
- Asks for a router
- Receives the address of the router and IP configuration

DHCPv6 (Stateless)

- If necessary additional configuration can be obtained (for example static routes)
- It is done by DHCPv6
- To configure open IPv6 → ND

DHCPv6 (Stateless)



ND <all>

Interface: bridge1

RA Interval: 200-600 s

RA Delay: 3 s

MTU:

Reachable Time: s

Retransmit Interval: s

RA Lifetime: 1800 s

Hop Limit:

☑ Advertise MAC Address

☑ Advertise DNS

□ Managed Address Configuration

☑ Other Configuration

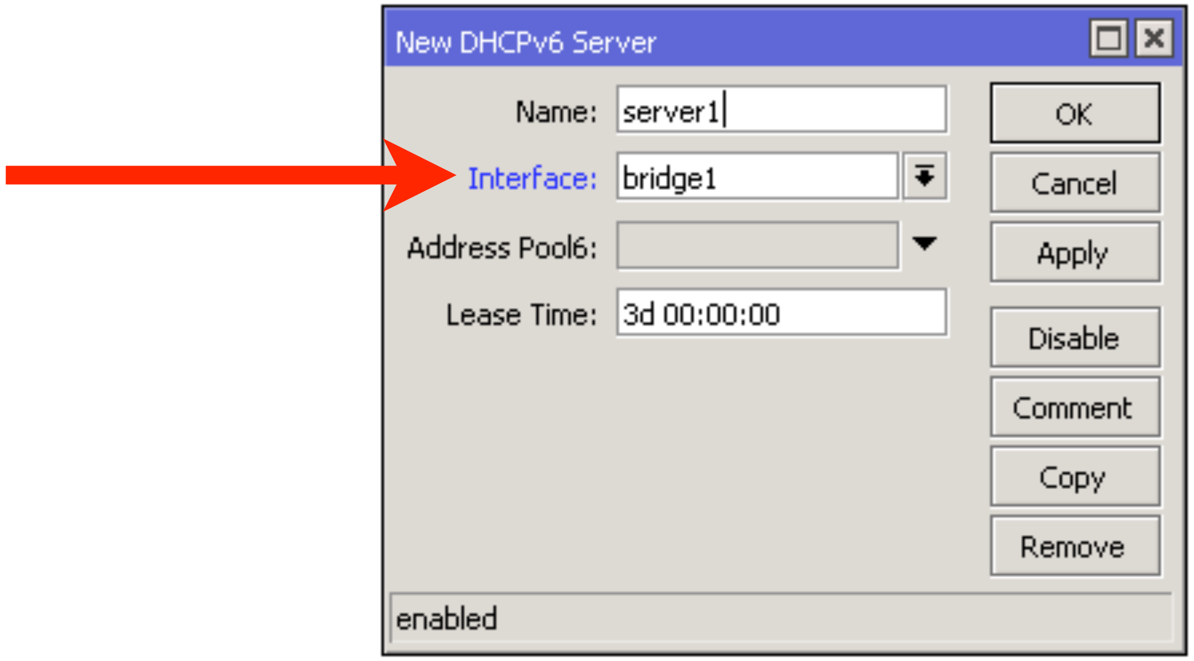
enabled

default

IPv6 → ND → ‘edit’

- Configure required interfaces and enable “Other Configuration”

DHCPv6 (Stateless)



New DHCPv6 Server

Name: server1

Interface: bridge1

Address Pool6:

Lease Time: 3d 00:00:00

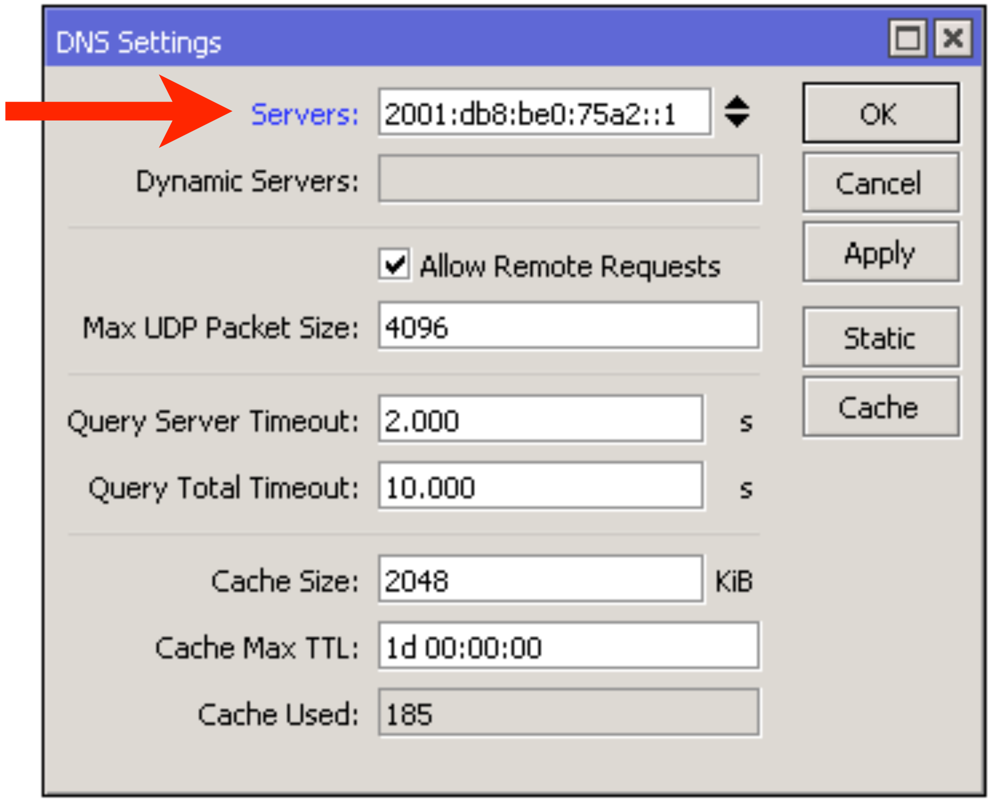
enabled

IPv6 → DHCPv6 → ‘+’

- Add new DHCP server on an interface

DHCPv6 (Stateless)

- Note: For MS Windows clients it is necessary to configure DHCPv6 in order to obtain DNS configuration
- Make sure, that IPv6 DNS server is configured in IP → DNS

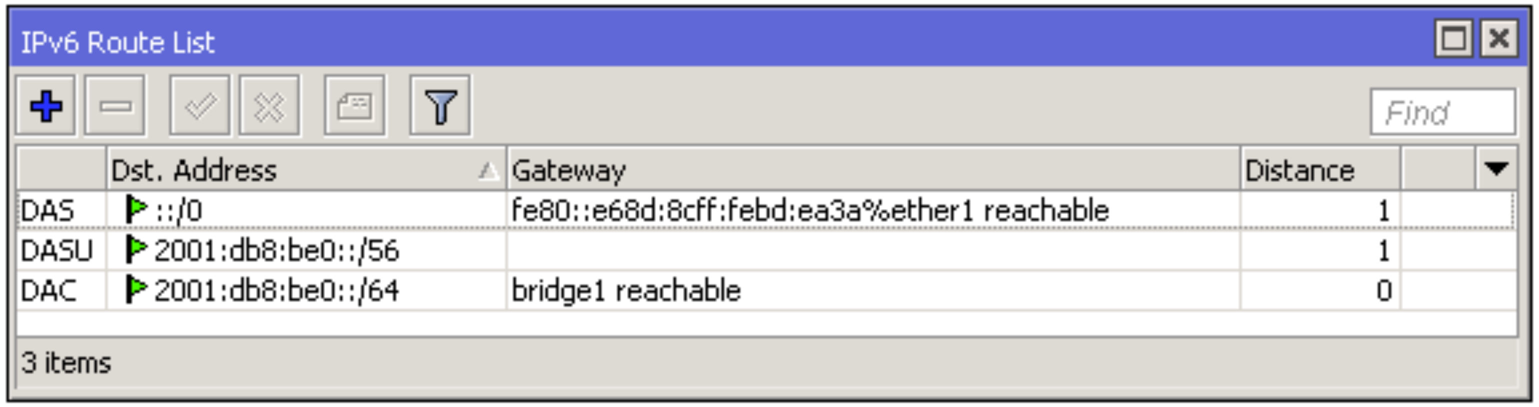


Servers: 2001:db8:be0:75a2::1
Dynamic Servers:
☑ Allow Remote Requests
Max UDP Packet Size: 4096
Query Server Timeout: 2.000 s
Query Total Timeout: 10.000 s
Cache Size: 2048 KiB
Cache Max TTL: 1d 00:00:00
Cache Used: 185

IP → DNS

IPv6 Routing

- Works similar like IPv4 classless routing
- Subnet size can be arbitrary
- SLAAC works only with /64 prefixes



	Dst. Address	Gateway	Distance
DAS	::/0	fe80::e68d:8cff:febd:ea3a%ether1 reachable	1
DASU	2001:db8:be0::/56		1
DAC	2001:db8:be0::/64	bridge1 reachable	0

3 items

IPv6 → Routes

IPv6 Routing

	IPv6	IPv4
Default gateway	0:0:0:0:0:0:0:0/0	0.0.0.0/0
	::/0	
	2000::/3	

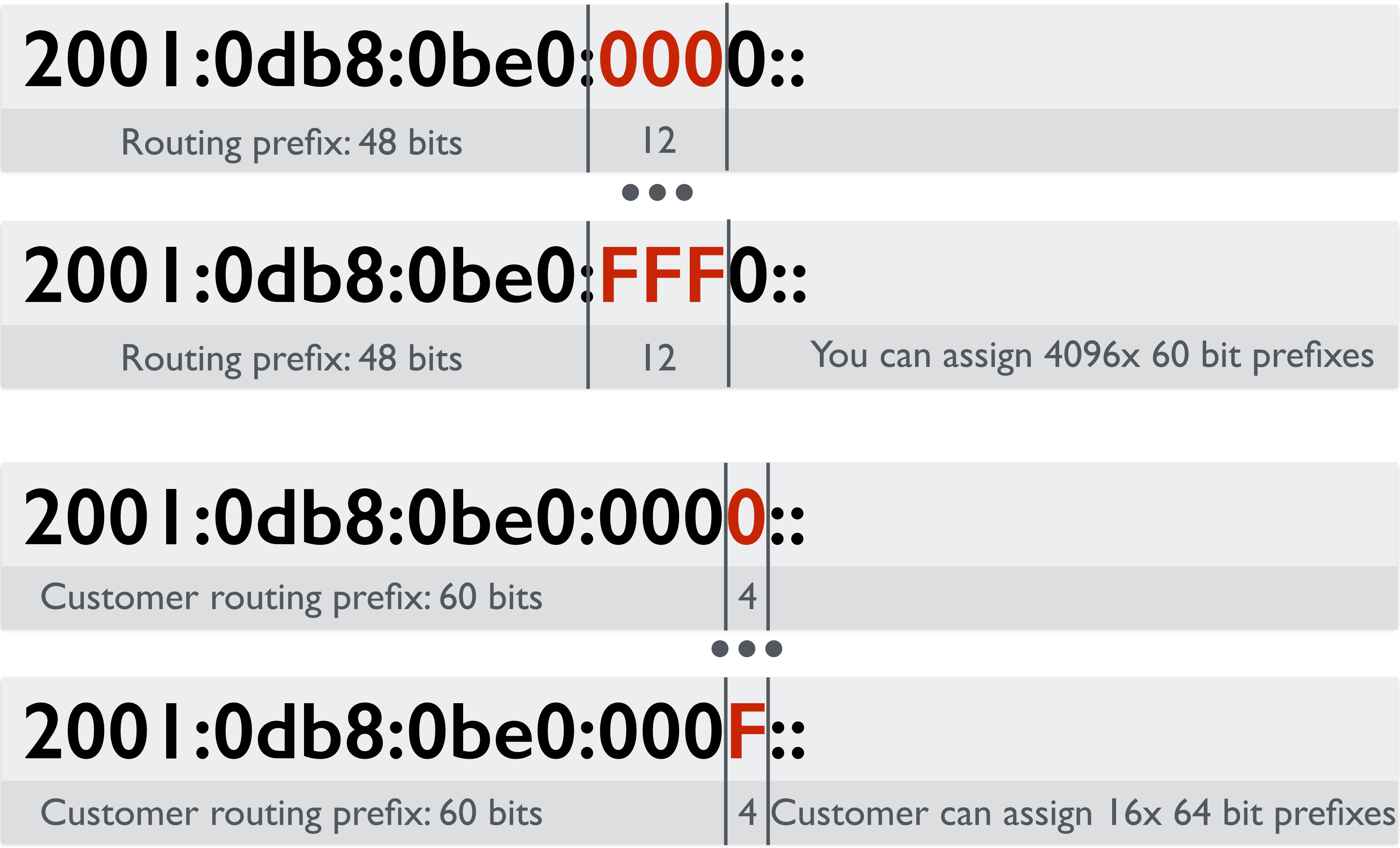
- Several ways how to write default gateway

MTCIPv6E

IPv6 Subnetting

- You have been assigned /48 block
- You're planning to assign /60 to your customers
- 2^{12} = 4096 /60 subnets
- Each of your customers will have 16x /64 prefixes for their devices

IPv6 Subnetting



2001:0db8:0be0:0000::

Routing prefix: 48 bits | 12

• • •

2001:0db8:0be0:FFF0::

Routing prefix: 48 bits | 12 | You can assign 4096x 60 bit prefixes

2001:0db8:0be0:0000::

Customer routing prefix: 60 bits | 4

• • •

2001:0db8:0be0:000F::

Customer routing prefix: 60 bits | 4 | Customer can assign 16x 64 bit prefixes

IPv6

- It is possible to split /64 prefix even further
- SLAAC requires /64 prefix length
- If the prefix is split beyond /64 will have to use DHCPv6 or static configuration
- Simpler devices might not support DHCPv6 (only SLAAC)

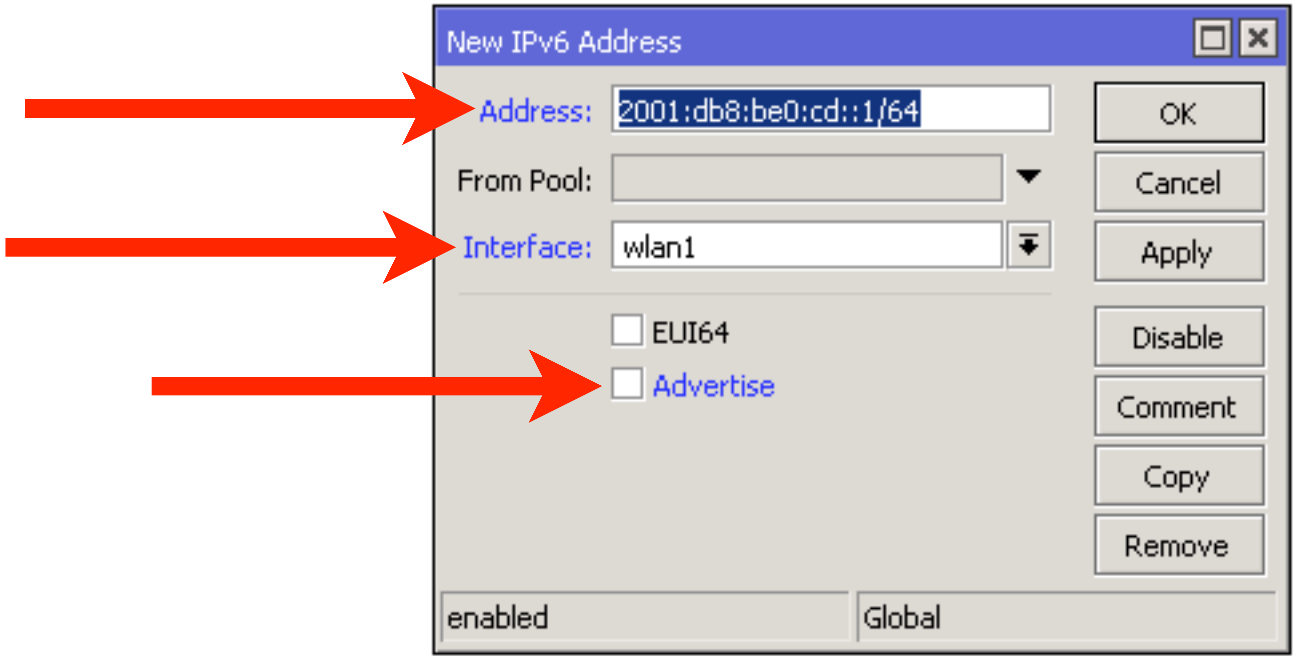
LAB

Configure IPv6

- The trainer now will give you an IPv6 address
- Configure it on your router's external interface (the same that already has public IPv4 address)
- Uncheck 'Advertise'
- From your router try to ping trainer's router IPv6 address

LAB

Configure IPv6



New IPv6 Address

Address: 2001:db8:be0:cd::1/64

From Pool:

Interface: wlan1

□ EUI64

□ Advertise

enabled

Global

IPv6 → Addresses → ‘+’

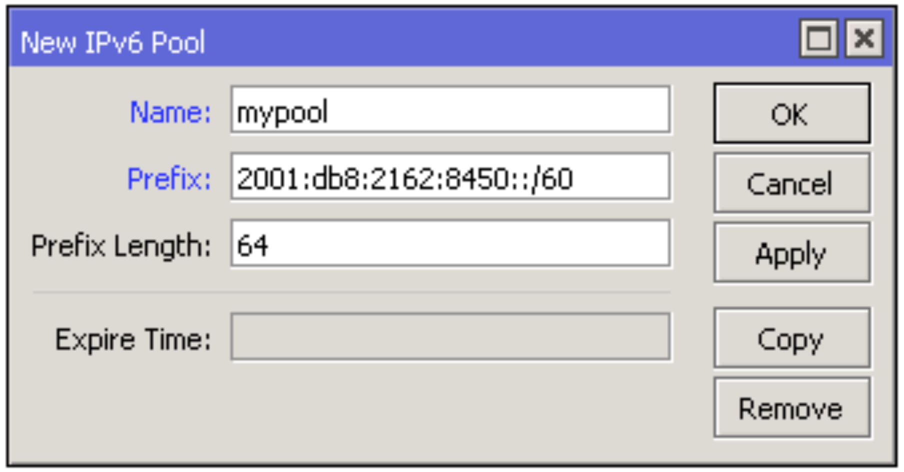
Configure IPv6

LAB

- The trainer now will give you an IPv6 prefix which to use for your clients
- Add it as an IPv6 pool
- Add an IP address on the bridge interface from the pool
- Enable IPv6 on your laptop
- It should receive an IPv6 prefix via SLAAC

Configure IPv6

- For example, the prefix is
  - 2001:db8:2162:8450::/60
  - Your laptop and other devices will receive /64 prefix



New IPv6 Pool

Name: mypool

Prefix: 2001:db8:2162:8450::/60

Prefix Length: 64

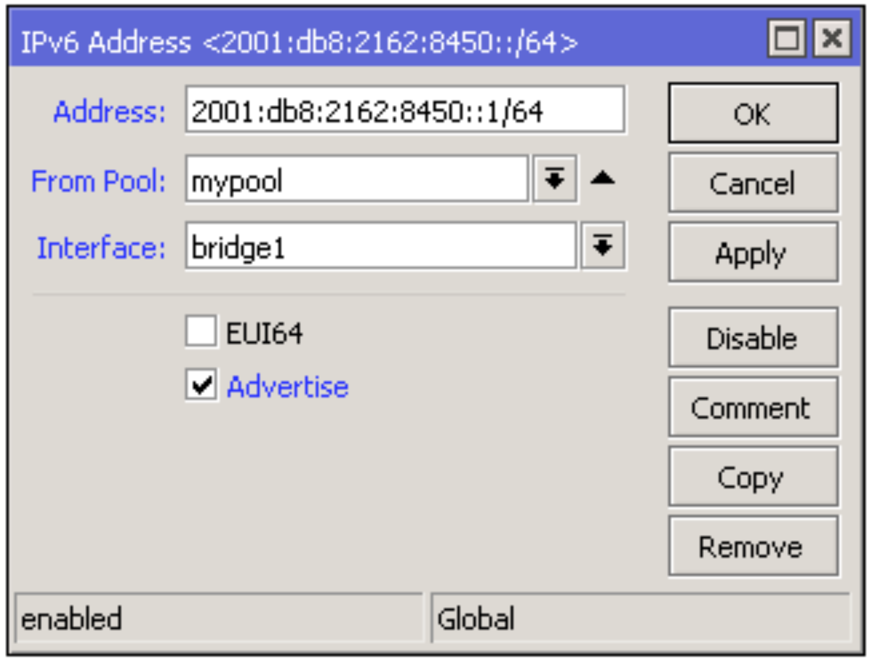
Expire Time:

IPv6 → Pool → ‘+’

LAB

Configure IPv6

- Choose an IP address from the pool, for example 2001:db8:2162:8450::1/64
- Configure it on the bridge interface
- Enable 'Advertise'


IPv6 Address <2001:db8:2162:8450::/64>

Address: 2001:db8:2162:8450::1/64

From Pool: mypool

Interface: bridge1

☐ EUI64

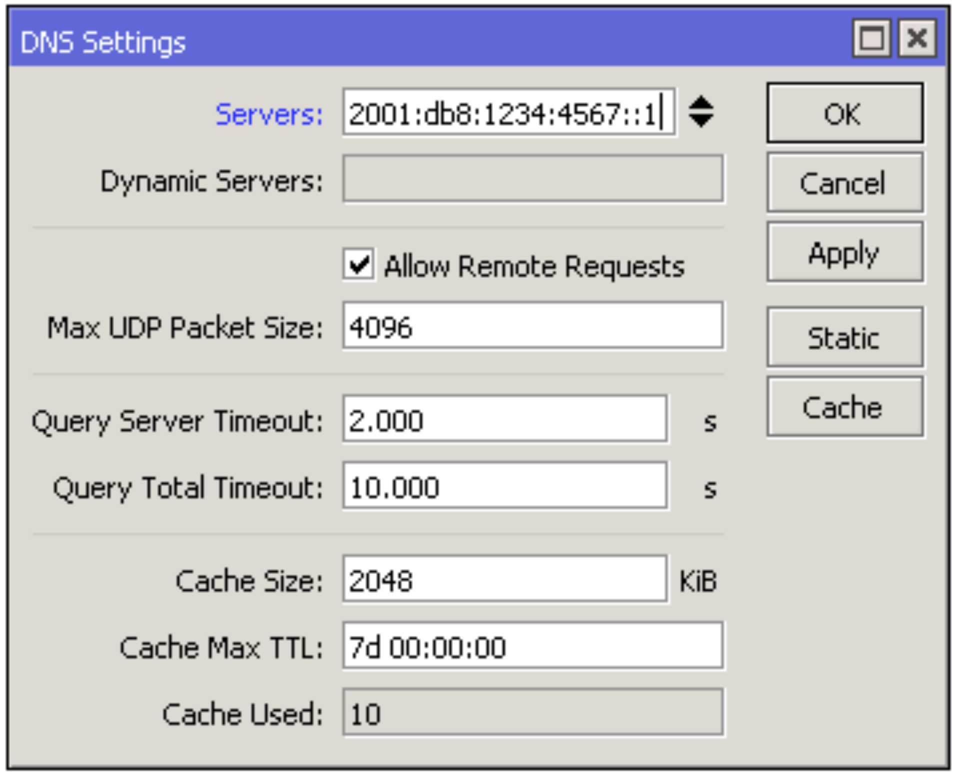
☑ Advertise

enabled | Global

IPv6 → Addresses → '+'

Configure IPv6

- The trainer now will give you an IPv6 address of the DNS server to use



DNS Settings

Servers: 2001:db8:1234:4567::1

Dynamic Servers:

☑ Allow Remote Requests

Max UDP Packet Size: 4096

Query Server Timeout: 2.000 s

Query Total Timeout: 10.000 s

Cache Size: 2048 KiB

Cache Max TTL: 7d 00:00:00

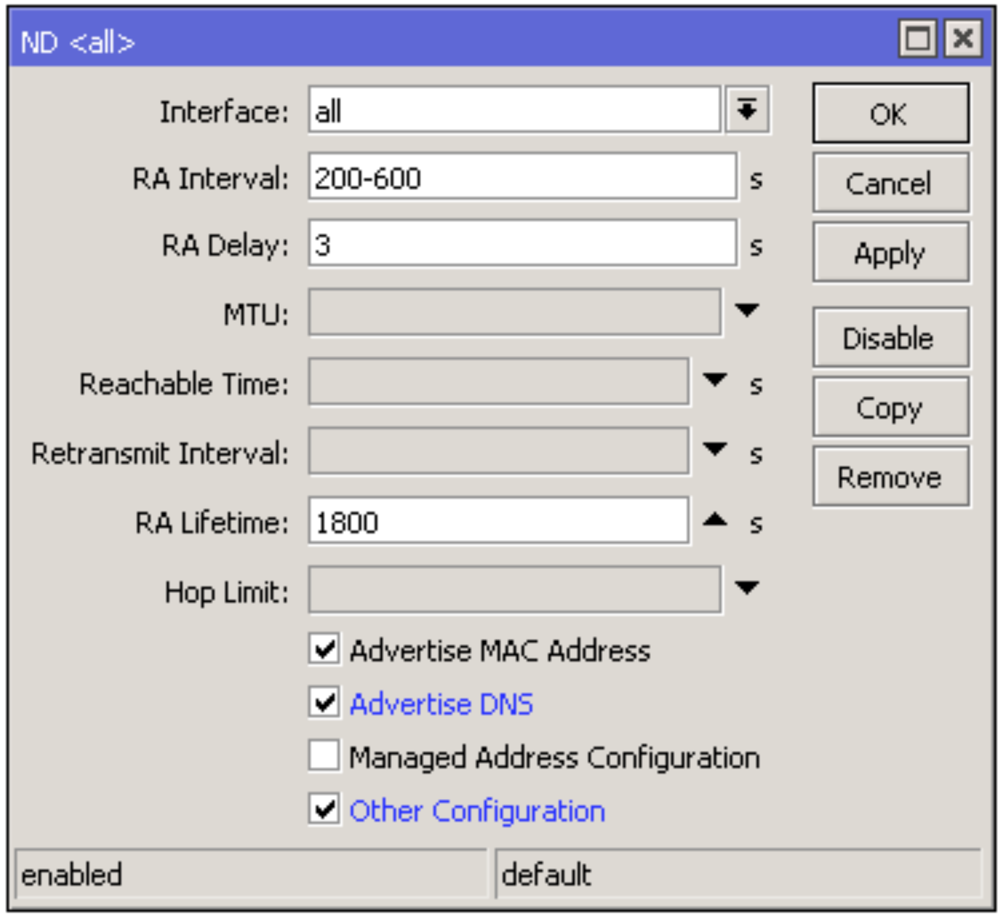
Cache Used: 10

IP → DNS

Configure IPv6

LAB

- Enable “Advertise DNS” in IPv6 → ND
- Linux and macOS should already have fully working IPv6
- If you’re using Windows, enable “Other configuration” flag



ND <all>

Interface: all

RA Interval: 200-600 s

RA Delay: 3 s

MTU:

Reachable Time: s

Retransmit Interval: s

RA Lifetime: 1800 s

Hop Limit:

☑ Advertise MAC Address

☑ Advertise DNS

☐ Managed Address Configuration

☑ Other Configuration

enabled | default

IPv6 → ND → ‘edit’

Configure IPv6

LAB

- Enable IPv6 on your laptop
- Try to ping the router’s IP address from your laptop (using ping6 command)
- Try to ping www.mikrotik.com IPv6 address (2a02:610:7501:1000::2)

MikroTik
MTCIPv6E

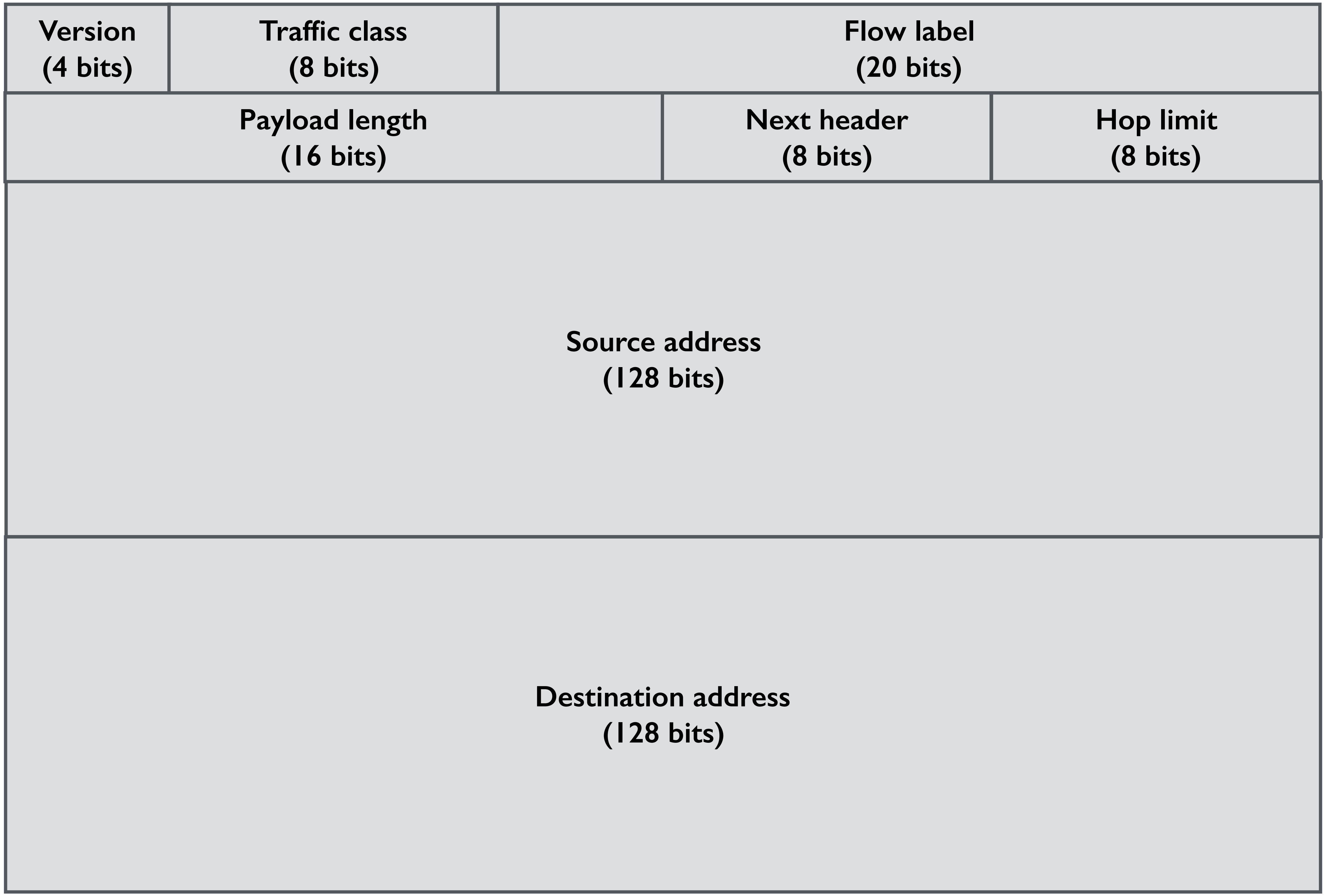
Module 2 Summary

Certified IPv6 Engineer (MTCIPv6E)

Module 3

IPv6 Packet

IPv6 Header



Version
(4 bits)

Traffic class
(8 bits)

Flow label
(20 bits)

Payload length
(16 bits)

Next header
(8 bits)

Hop limit
(8 bits)

Source address
(128 bits)

Destination address
(128 bits)

IPv6 Header

- **Version** - always contains '6' (0110 in binary)
- **Traffic class** - holds 2 values.
  - 6 most significant bits - to classify packets for QoS
  - 2 remaining bits - for Explicit Congestion Notification (ECN) where supported

IPv6 Header

- **Flow label** - used to maintain packet sequence
- **Payload length** - Length of the IPv6 payload, i.e., the rest of the packet following this IPv6 header, in octets
- **Next header** - Identifies the type of header immediately following the IPv6 header

IPv6 Header

- **Hop limit** - Decremented by 1 by each router that forwards the packet. The packet is discarded if hop limit is 0
- **Source address** - address of the originator of the packet
- **Destination address** - address of the intended recipient of the packet

IPv6 Header

- Length: fixed size 40 bytes (320 bits)
- Field count: 8
- Simplified in comparison to IPv4

Next Header Field

- IPv6 header has fixed size
- Optional information is encoded in separate extension headers
- Situated between the IPv6 and the upper-layer headers
- Each Next Header is identified by a distinct value

Next Header Field

- IPv6 packet may carry zero, one, or more extension headers

Extension Header	Value
Hop-by-Hop Options	0
Fragment	44
Routing (Type 0)	43
Destination Options	60
Authentication	51
Encapsulating Security Payload	50

Fragmentation

- Performed only by source nodes
- Fragment header is identified by a Next Header value of 44
- For every packet the source node generates an identification value
- ID must be different than any other fragmented packet sent recently with the same Src and Dst Address

Fragmentation

- The packet consists of "unfragmentable" and "fragmentable" parts
- Unfragmentable = IPv6 header + extenstion headers that must be processed by routers en route to the destination
- Fragmentable = the rest of the packet

Path MTU

- Path MTU (PMTU) is the largest packet size that can traverse between a source and destination without fragmentation
- IPv6 requires MTU 1280 bytes or greater
    - IPv4 requires MTU 68 bytes

Path MTU Discovery

- PMTU discovery is a technique for determining the path MTU between two IP hosts
- To discover and take advantage of PMTUs greater than 1280, it is strongly recommended to implement PMTU discovery
- For packets that are larger than PMTU fragmentation is used

Module 3
Summary

Certified IPv6 Engineer (MTCIPv6E)

Module 4

IPv6 Security

ICMPv6

- ICMPv6 is an integral part of IPv6
- It is used to report errors encountered in processing packets, and to perform other functions, such as diagnostics
- There are 2 ICMPv6 message classes - error (types 0-127) and information (types 128-255)

ICMPv6

Type	Meaning	Class
1	Destination Unreachable	Error
3	Time Exceeded	
128	Echo Request	Information
129	Echo Reply	

ICMPv6 Message Types (example)

Neighbor Discovery

- NDP uses 5 different ICMPv6 packet types:
  - Router solicitation (type 133)
  - Router advertisement (type 134)
  - Neighbor solicitation (type 135)
  - Neighbor advertisement (type 136)
  - Redirect (type 137)

Neighbor Discovery

- Neighbor Discovery makes use of a number of different special addresses including:
  - Link-local scope address to reach all nodes (multicast address) - FF02::1
  - Link-local scope address to reach all routers (multicast address) - FF02::2
  - And others, for more info see - IPv6 Multicast Address Space Registry

Router Solicitation

- Hosts send Router Solicitations in order to prompt routers to generate Router Advertisements quickly rather than at their next scheduled time
- It is sent to all-routers multicast address

Router Solicitation

- Source - IP address assigned to the sending interface
- Or the unspecified address (::/128) if no address is assigned
- Destination - typically the all-routers multicast address

Router Advertisement

- Routers advertise their presence periodically, or in response to a Router Solicitation message
- A host receives Router Advertisements from all routers, building a list of default routers
- Various internet and link parameters are advertised such as prefixes, address configuration, MTU, etc.

Router Advertisement

- Facilitates centralized administration of critical parameters, that can be set on routers and automatically propagated to all attached hosts
- Allow routers to inform hosts how to perform address autoconfiguration

Router Advertisement

- Routers can specify whether hosts should use DHCPv6 and/or autonomous (stateless) address configuration
- Contains - source, link-local address assigned to the interface from which this message is sent

Router Advertisement

- Destination, typically the Source Address of an invoking Router Solicitation or the all-nodes multicast address
- M: 1-bit "Managed address configuration" flag
- O: 1-bit "Other configuration" flag

Neighbor Solicitation

- Nodes accomplish address resolution by multicasting a Neighbor Solicitation, that asks the target node to return its link-layer address
- To verify that a neighbor is still reachable
- The target returns its link-layer address in a unicast Neighbor Advertisement message

Neighbor Solicitation

- A single request-response pair of packets is sufficient for both to resolve each other's link-layer addresses
- Neighbor Solicitation is also used for Duplicate Address Detection

Neighbor Solicitation

- Contains - source, either an address assigned to the interface from which this message is sent or (if Duplicate Address Detection is in progress) the unspecified address
- Destination, either the solicited-node multicast address corresponding to the target address, or the target address

Neighbor Advertisement

- A response to a Neighbor Solicitation message
- A node may also send unsolicited Neighbor Advertisements in order to (unreliably) propagate new information quickly
- E.g. to announce a link-layer address change

Neighbor Advertisement

- Source: an address assigned to the interface from which the advertisement is sent
- Destination: the Source Address of an invoking Neighbor Solicitation or the all-nodes multicast address

Redirect

- Used by routers to inform hosts of a better first hop for a destination
- Hosts can also be informed by a redirect that the destination is in fact a neighbor
- Separate address resolution is not needed upon receiving a redirect

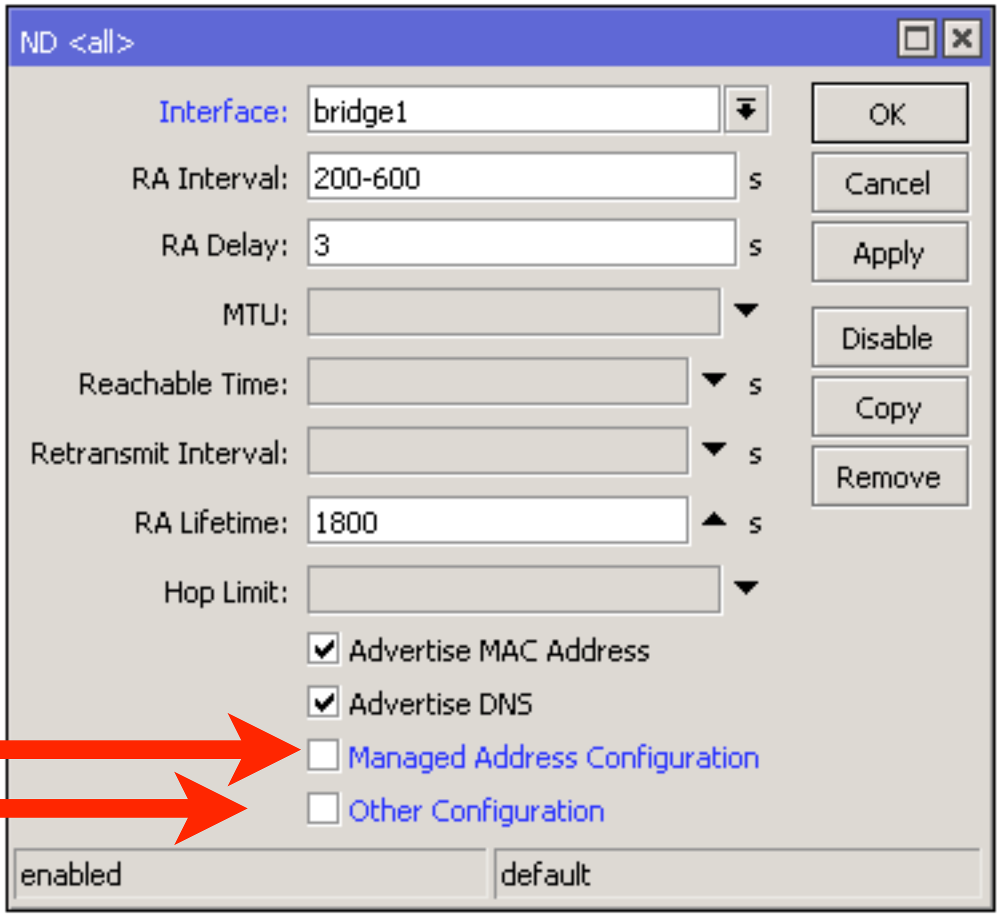
Managed Address Configuration

- Router Advertisement 1-bit M flag
- When set, it indicates that addresses are available via DHCPv6
- If the M flag is set, the O flag is redundant and can be ignored because DHCPv6 will return all available configuration information
- SLAAC will not be used

Other Configuration

- Router Advertisement 1-bit O flag
- When set, it indicates that other configuration information is available via DHCPv6
- E.g. DNS-related information (necessary for Windows clients)
- If neither M nor O flags are set, this indicates that no information is available via DHCPv6

M and O Flags



Interface: bridge1

RA Interval: 200-600 s

RA Delay: 3 s

MTU:

Reachable Time: s

Retransmit Interval: s

RA Lifetime: 1800 s

Hop Limit:

☑ Advertise MAC Address

☑ Advertise DNS

□ Managed Address Configuration

□ Other Configuration

enabled | default

M flag

O flag

IPv6 → ND → ‘edit’

MikroTik MTCIPv6E

Duplicate Address Detection (DAD)

- Using Neighbor Solicitation a node can determine whether or not an address it wishes to use is already in use
- DAD sends a message with an unspecified source address targeting its own "tentative" address

Duplicate Address Detection (DAD)

- Such messages trigger nodes already using the address to respond with a multicast Neighbor Advertisement indicating that the address is in use
- If no response is received, the node uses the chosen address

Neighbor Unreachability Detection (NUD)

- Communication to or through a neighbor may fail for numerous reasons at any time, including hardware failure, hot-swap of an interface card, etc.
- NUD detects the failure of a neighbor or the failure of the forward path to the neighbor

Neighbor Unreachability Detection (NUD)

- NUD uses confirmation from two sources
- When possible, upper-layer protocols provide a positive confirmation that a connection is making "forward progress"

Neighbor Unreachability Detection (NUD)

- When positive confirmation is not forthcoming, a node sends unicast Neighbor Solicitation messages that solicit Neighbor Advertisements as reachability confirmation from the next hop
- If node address changes NUD ensures that all nodes will reliably discover the new address

Multicast Listener Discovery (MLD)

- MLDv2 is a translation of the IGMPv3 protocol for IPv6 semantics
- It is used by an IPv6 router to discover multicast listeners (nodes that wish to receive multicast packets) on directly attached links
- To discover which multicast addresses are of interest to those neighboring nodes

MLD

- The purpose of MLD is to enable each multicast router to learn, which multicast addresses and which sources have interested listeners
- Specifies multicast address listeners and multicast routers
- A node can subscribe to certain multicast messages

MLD

- One router becomes elected as the Querier
- It will gather and maintain information about listeners and their subscriptions
- If the router fails another router on the same subnet takes over the role

SEND

- If not secured, NDP is vulnerable to various attacks
- SEcure Neighbor Discovery (SEND) is a proposed standard which helps to mitigate possible threats
- For more info see RFC3971

Special Addresses Lab

LAB

- Login to your router
- Open terminal and try to ping following IP addresses:
  - FF02::1 (all nodes)
  - FF02::2 (all routers)
- Observe the output

Temporary Addresses

- Addresses generated using SLAAC contain an embedded interface identifier, which remains constant over time
- When a fixed identifier is used in multiple contexts, it becomes possible to correlate seemingly unrelated activity using this identifier

Temporary Addresses

- For a "road warrior" who has Internet connectivity both at home and at the office, the interface identifier contained within the address remains the same
- Privacy Extensions for SLAAC in IPv6 (RFC4941) suggests improvements to this behavior

Temporary Addresses

- There are various implementations
- macOS and Windows10 generate new temporary IPv6 address every 24 hours
- Linux may create new temporary address for each new SSL/TLS connection

LAB

Temporary Addresses

- Find out the temporary address(es) of your computer
- If you're using Linux/macOS, open terminal and use command `ifconfig`
- For Windows - `ipconfig`

Firewall

- RouterOS IPv6 → Firewall is similar with IP → Firewall
- RouterOS IPv6 Firewall implements same Filter and Mangle rules as with IPv4
- As well as Address Lists

Firewall

- By default RouterOS IPv6 firewall does not have any filter rules
- Protect your router from outside

LAB

Firewall

- Create following IPv6 Firewall rules:
  - Accept input for established and related packets (all interfaces)
  - Accept ICMPv6 from link local (LL) IP addresses (ff80::/10)
  - Accept ICMPv6 to link local (LL) IP addresses (ff80::/10)

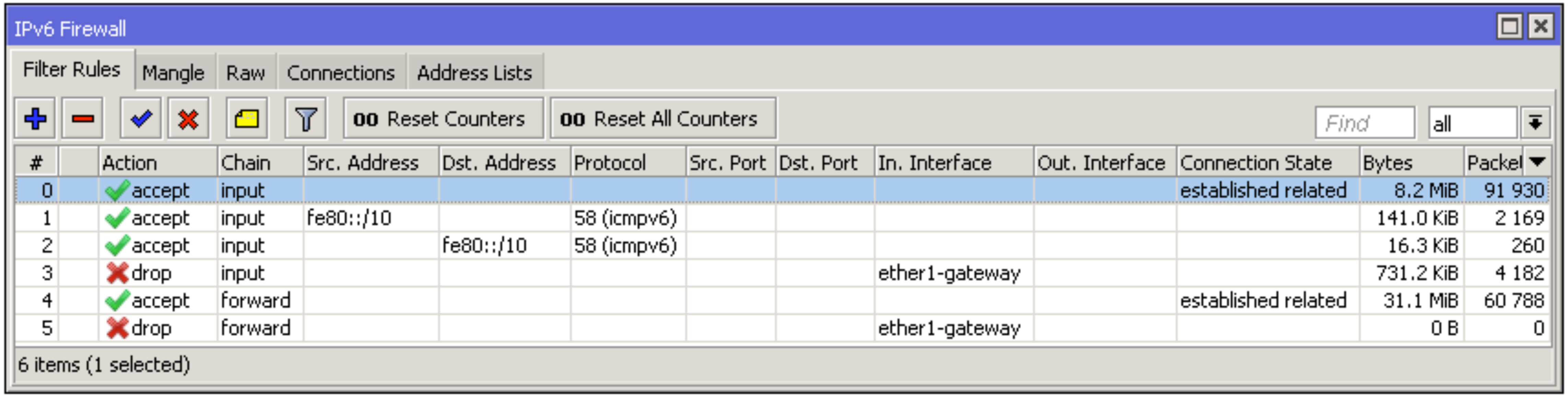
LAB

Firewall

- Create following IPv6 Firewall rules:
  - Drop input for everything else on external interface
  - Accept forward for established and related packets (all interfaces)
  - Drop forward for all traffic coming in through external interface

LAB

Firewall



#		Action	Chain	Src. Address	Dst. Address	Protocol	Src. Port	Dst. Port	In. Interface	Out. Interface	Connection State	Bytes	Packet
0		✓ accept	input								established related	8.2 MiB	91 930
1		✓ accept	input	fe80::/10		58 (icmpv6)						141.0 KiB	2 169
2		✓ accept	input		fe80::/10	58 (icmpv6)						16.3 KiB	260
3		✗ drop	input						ether1-gateway			731.2 KiB	4 182
4		✓ accept	forward								established related	31.1 MiB	60 788
5		✗ drop	forward						ether1-gateway			0 B	0

6 items (1 selected)

IPv6 → Firewall → Filter Rules

NAT

- There's no IPv6 → Firewall → NAT menu
- No need for NAT
  - There are plenty IPv6 addresses available
- One should not confuse NAT box with firewall - it does not provide security in itself
- See RFC5902: IAB Thoughts on IPv6 NAT

IPsec

- Internet Protocol Security (IPsec) - a set of protocols to support secure communication at the IP layer
- Originally developed for IPv6, later backported also to IPv4
- Provides encryption to the IP protocol
- Can be used both with IPv4 and IPv6

IPsec

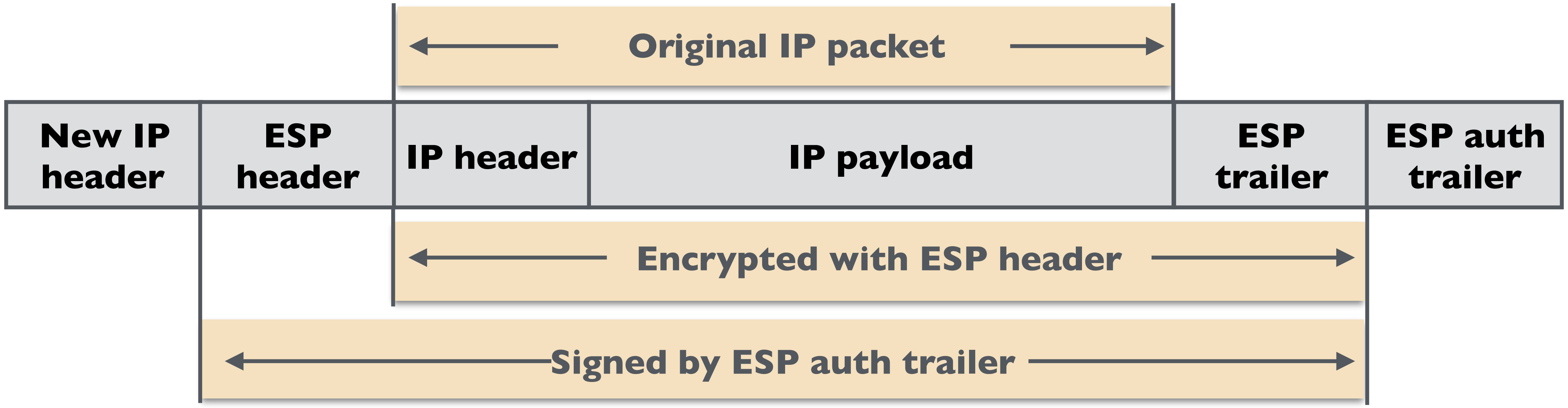
- Multiple approaches can be used to implement IPsec:
    - Header only encryption (AH)
    - Data only encryption (ESP)
    - Header and data encryption (AH+ESP)
- ESP (packet data encryption) is the most widely used, the other two are used rarely

IPsec

- Can be configured to operate in two different modes:
  - Transport
  - Tunnel
- Both can be used to encrypt IPv6 traffic

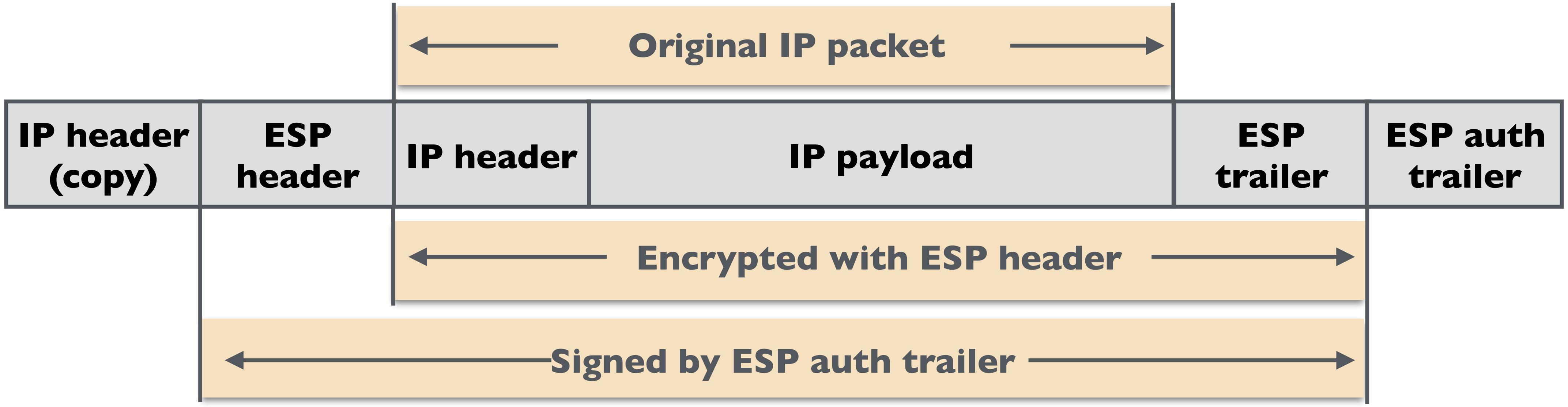
Tunnel Mode

- The original packet is wrapped, encrypted, a new IP header is added and the packet is sent to the other side of the tunnel



Transport Mode

- The data of the packet is encrypted, but the header is sent in open clear text, IP header is copied to the front



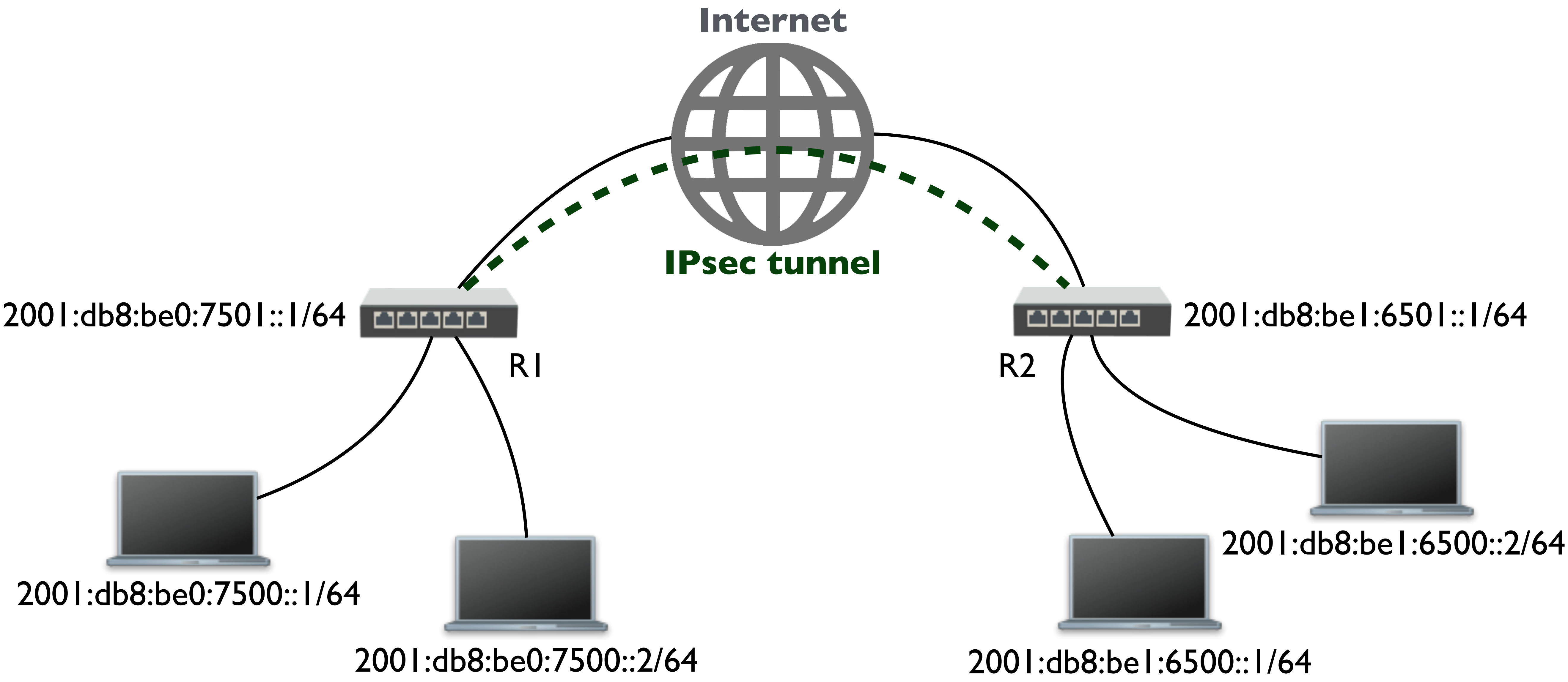
IPsec

- IPv6 Node Requirements (RFC6434) states that all IPv6 nodes SHOULD support IPsec

SHOULD - means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course

IPsec Tunnel Mode

Example



IPsec Tunnel Mode

Example

- IPsec peer config
- R1

```
/ip ipsec peer add address=2001:db8:be1:6501::1 port=500
auth-method=pre-shared-key secret="test"
```

- R2

```
/ip ipsec peer add address=2001:db8:be0:7501::1 port=500
auth-method=pre-shared-key secret="test"
```

IPsec Tunnel Mode

Example

- IPsec default proposal on both routers

```
/ip ipsec proposal print
 0  * name="default" auth-algorithms=sha1 enc-
algorithms=aes-256-cbc,aes-192-cbc,aes-128-cbc
lifetime=30m pfs-group=modp1024
```

IPsec Tunnel Mode

Example

- IPsec policy config
- R1

```
/ip ipsec policy
add src-address=2001:db8:be0:7500::/64 src-port=any dst-
address=2001:db8:be1:6500::/64 dst-port=any \
sa-src-address=2001:db8:be0:7501::1 sa-dst-
address=2001:db8:be1:6501::1 \
tunnel=yes action=encrypt proposal=default
```

IPsec Tunnel Mode

Example

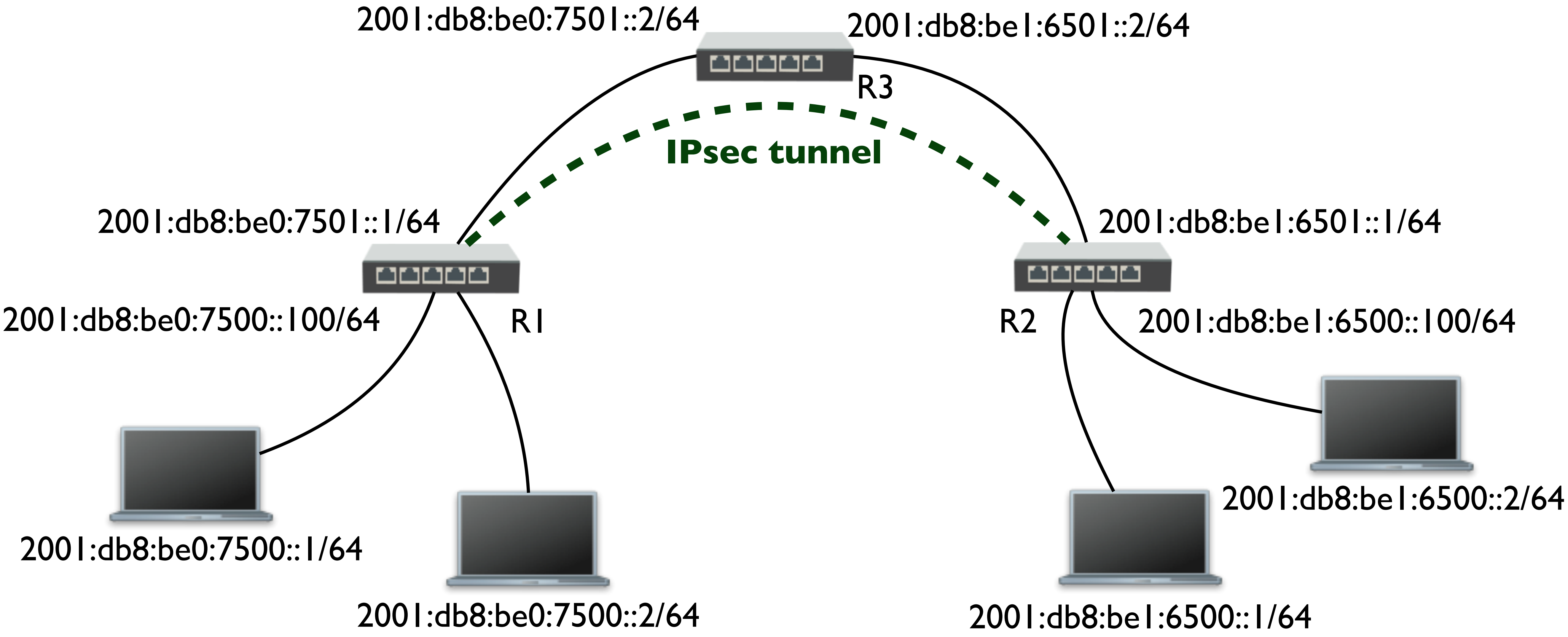
- IPsec policy config
- R2

```
/ip ipsec policy
add src-address=2001:db8:be1:6500::/64 src-port=any dst-
address=2001:db8:be0:7500::/64 dst-port=any \
sa-src-address=2001:db8:be1:6501::1 sa-dst-
address=2001:db8:be0:7501::1 \
tunnel=yes action=encrypt proposal=default
```

- All traffic between subnets will be encrypted
- For more info see IPsec manual page

Optional LAB

IPsec LAB



2001:db8:be0:7501::2/64

2001:db8:be1:6501::2/64

R3

IPsec tunnel

2001:db8:be0:7501::1/64

2001:db8:be1:6501::1/64

2001:db8:be0:7500::100/64

R1

R2

2001:db8:be1:6500::100/64

2001:db8:be1:6500::2/64

2001:db8:be0:7500::1/64

2001:db8:be0:7500::2/64

2001:db8:be1:6500::1/64

Module 4 Summary

Certified IPv6 Engineer (MTCIPv6E)

Module 5

Transition Mechanisms

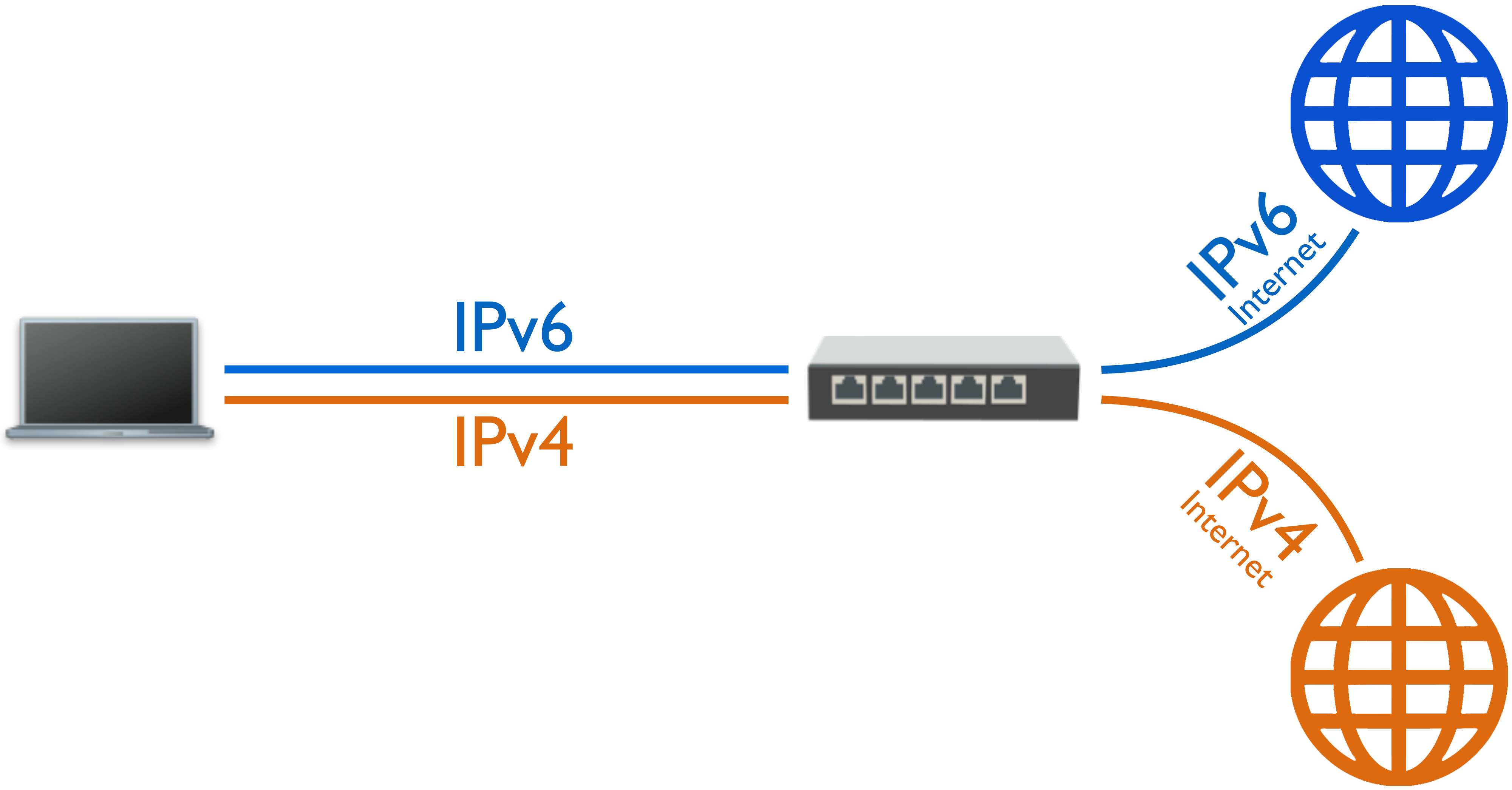
Transition Mechanisms

- Dual stack
- 6to4
- 6RD
- Teredo
- DS-lite (Dual stack lite)

Dual Stack

- Fully functional IPv4 and IPv6 work side by side
- The most recommended way of implementing IPv6
- Also endorsed by RIPE

Dual Stack



End-user device (host) has both IPv4 and IPv6 connectivity

Transition Mechanisms

- If for some reason dual stack is not possible, there are other options

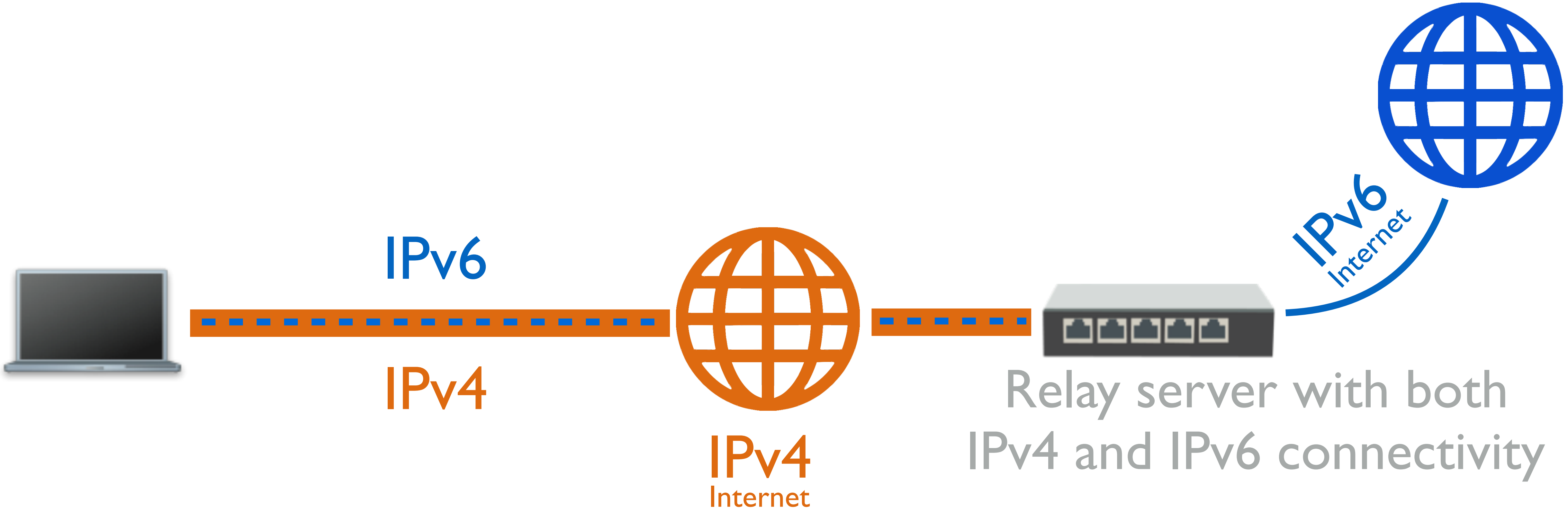
6to4

- Allows IPv6 packets to be transmitted over an IPv4 network
- A 6to4 relay server with native IPv6 connectivity needs to be configured on the other end
- Intended only as a transition mechanism, not as a permanent solution

6to4

- IPv6 packets are encapsulated in IPv4 packets
- Delivered to a 6to4 relay via IPv4 network
- Decapsulated and sent forward as IPv6 packets

6to4



Relay server with both IPv4 and IPv6 connectivity

6to4

- Ready to use services offer 6to4 tunnels free of charge
- E.g. Hurricane Electric, SixXS
- Can setup your own

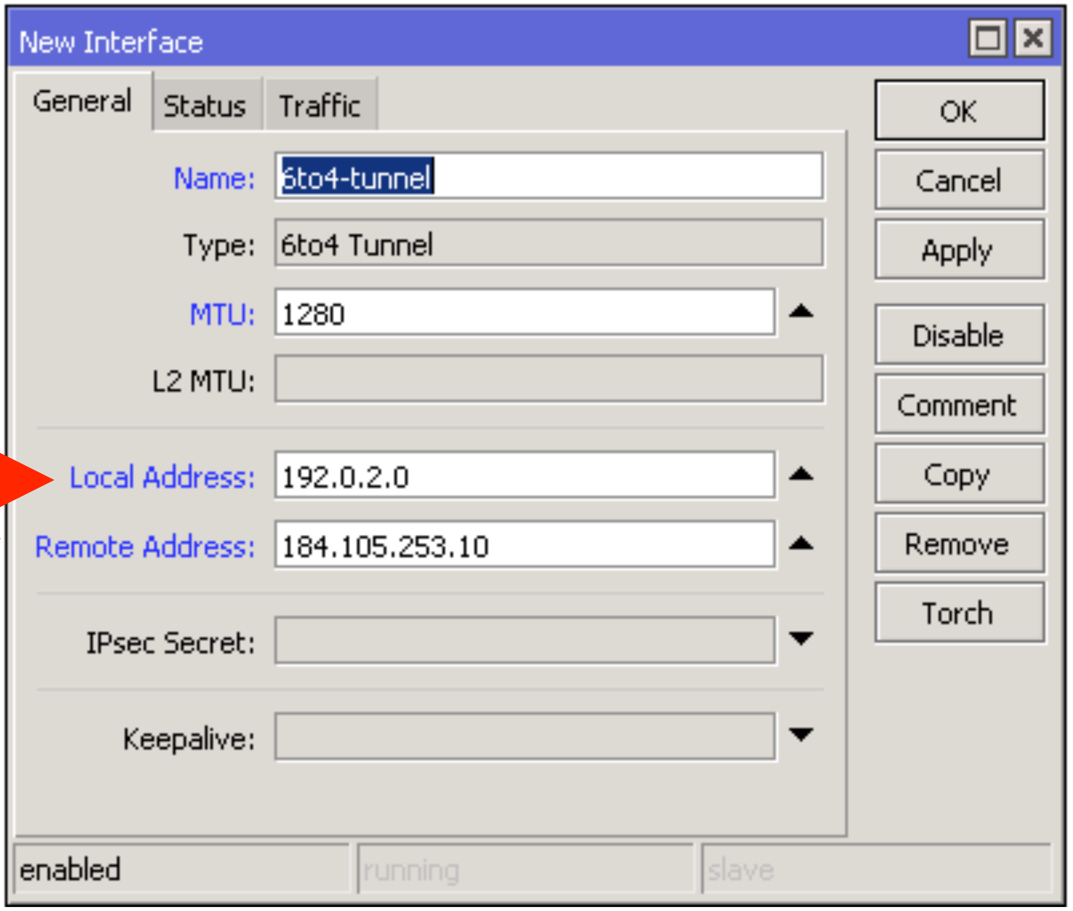
6to4

- Hurricane Electric (tunnelbroker.net) provides a 6to4 service with ready to use configuration for RouterOS
- Additional information how to get IPv6 connectivity can be found on wiki.mikrotik.com

6to4

- RouterOS 6to4 interface is used to set up the tunnel
- Local and remote public IPv4 addresses have to be entered
- 6to4 uses encapsulation, the MTU has to be changed to a smaller one

6to4



Name: 6to4-tunnel

Type: 6to4 Tunnel

MTU: 1280

L2 MTU:

Local Address: 192.0.2.0

Remote Address: 184.105.253.10

IPsec Secret:

Keepalive:

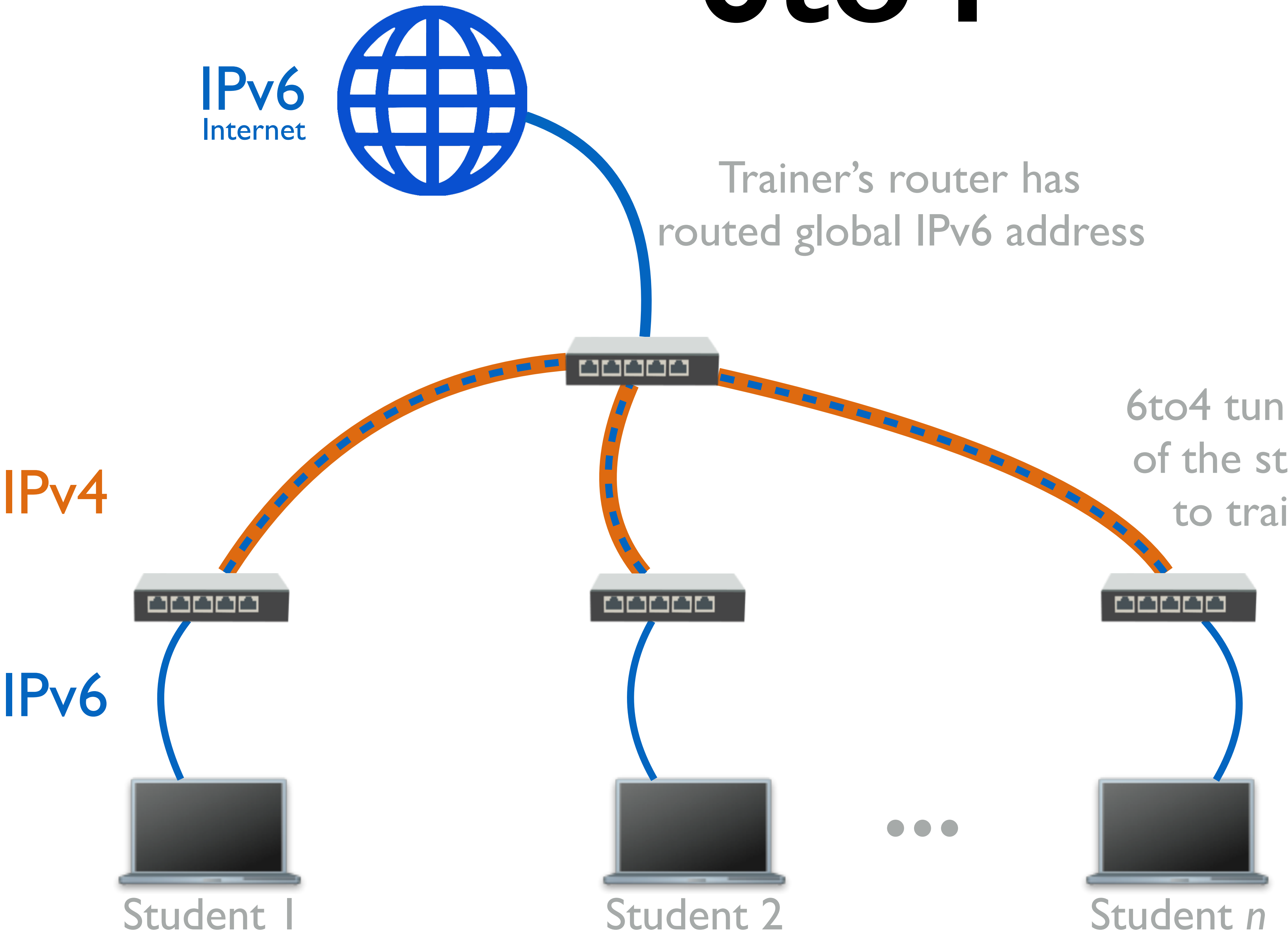
enabled | running | slave

Your public IP

Relay server IP

Interfaces → '+' → 6to4 Tunnel

6to4



Trainer's router has routed global IPv6 address

6to4 tunnels from each of the student's router to trainer's router

Student laptops have full IPv6 connectivity

Optional LAB

6to4

Trainer Configuration on trainer’s router

Student Configuration on student’s router

Optional LAB

6to4

- Trainer's router has been assigned a routed IPv6 prefix
  - Depending on the class size /60 might do, /56 should always be more than enough
- Decide how are you going to assign IPv4 and IPv6 addresses to student router's
- Create 6to4 tunnels from your router to each of student's routers (via IPv4)

Trainer

Optional LAB

6to4

- Assign each student IPv4 address which will be used to create a 6to4 tunnel back to your router
- Assign IPv6 ULAs to your end of tunnels, assign each student their 6to4 endpoint IPv6 address
- Create routes to student IPv6 prefixes through 6to4 interfaces

Trainer

Optional LAB

6to4

- The trainer will give you:
  - An IPv4 address that will be used to create a 6to4 tunnel
  - An IPv6 ULA that will be used for 6to4 interface
  - An IPv6 prefix which will be used to assign IP addresses to your devices via SLAAC
  - IPv6 address to use for the default route

Optional LAB

6to4

- Assign IPv4 address an interface which is connected to the trainer's router
- Create a 6to4 tunnel to the IP which the trainer gave you
- Assign IPv6 ULA to the 6to4 interface
- Create IPv6 pool with the assigned prefix

Optional LAB

6to4

- Add global IPv6 address to the local interface from the prefix, that the trainer gave to you, set advertise = yes
- Make sure that there is at least one reachable DNS server in IP → DNS
- Add default IPv6 (::/0) via the trainer’s 6to4 interface address

Optional LAB

6to4

- When done, open ipv6.mikrotik.com in your browser
- The end result should be that your laptop has full IPv6 connectivity via IPv4 network using 6to4 tunnel which encapsulates IPv6 packets into IPv4 packets

Optional LAB

6to4

- The trainer will give you a public IPv4 address
- Configure it on the router
- Register yourself on tunnelbroker.net
- Create a new regular tunnel (choose a destination close to you)
- Configure the tunnel on your router

Optional LAB

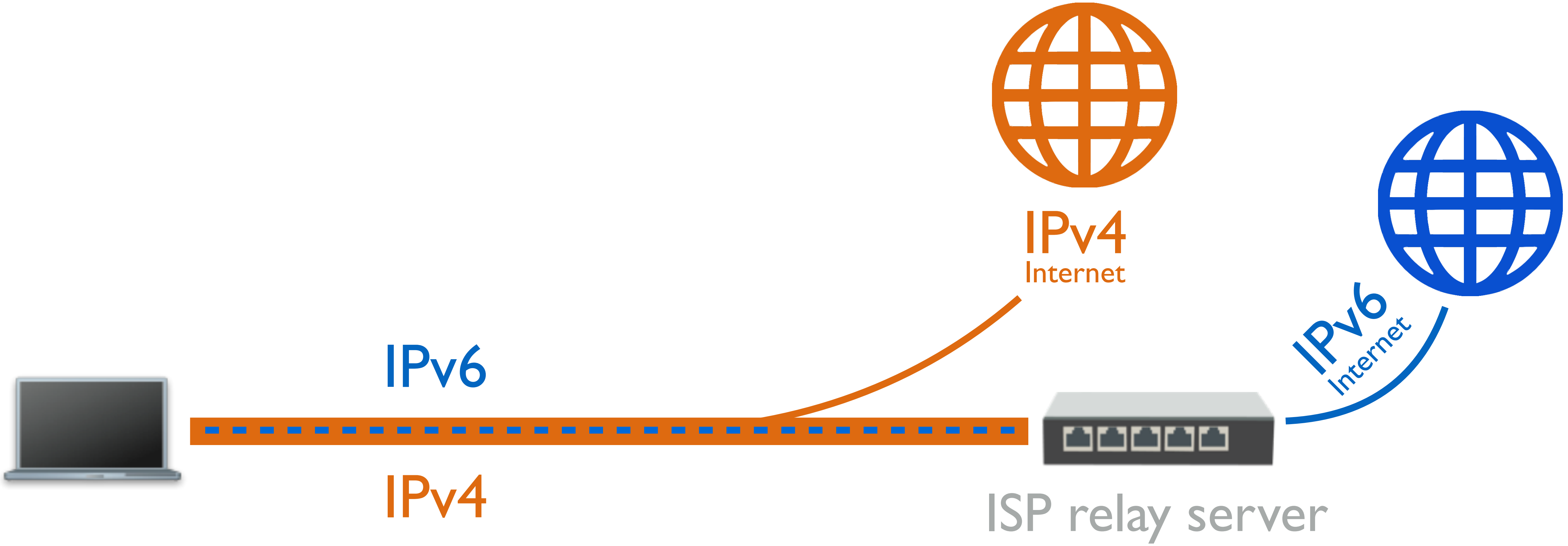
6to4

- Tunnelbroker website provides a script for RouterOS which can be used to set up the tunnel
- For more info see Tunnelbroker example on wiki.mikrotik.com
- When done, open ipv6.mikrotik.com in your browser

6RD

- IPv6 Rapid Deployment is 6to4 derivative
- IPv6 relay is controlled by your ISP
- From client to ISP is IPv4 network only
- On the client side additional software is needed to encapsulate IPv6 into IPv4 packets
- Described in RFC5569

6RD



Teredo

- Teredo encapsulates IPv6 traffic into IPv4 UDP packets
- The traffic is sent through IPv4 Internet
- Unlike 6to4, Teredo works behind an IPv4 NAT
- Uses Teredo prefix 2001::/32

Teredo

- Can only provide a single IPv6 address per tunnel endpoint
- Cannot be used to distribute addresses to multiple hosts like 6to4
- Developed by Microsoft
- Described in RFC4380

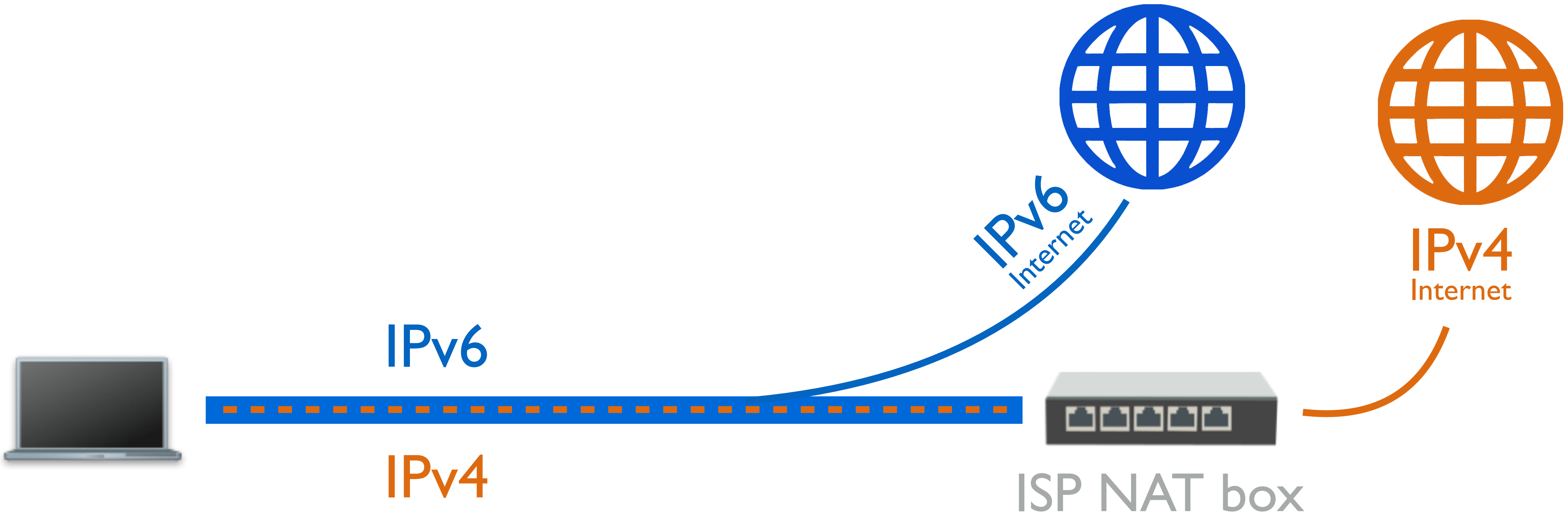
DS-lite

- Dual stack lite
- IPv6 only links are used between the ISP and the client
- Client has native IPv6 connectivity
- When and IPv4 packet needs to be sent, it is encapsulated into an IPv6 packet

DS-lite

- Sent to the ISP’s NAT box which decapsulates and forwards it as IPv4 traffic
- NAT is centralized at the ISP level
- Clients use private IPv4 addresses (e.g. 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16)
- ISP → Client network is IPv6 only

DS-lite



Module 5
Summary

Certified IPv6 Engineer (MTCIPv6E)

Module 6

Interoperability

IPv6 Pool

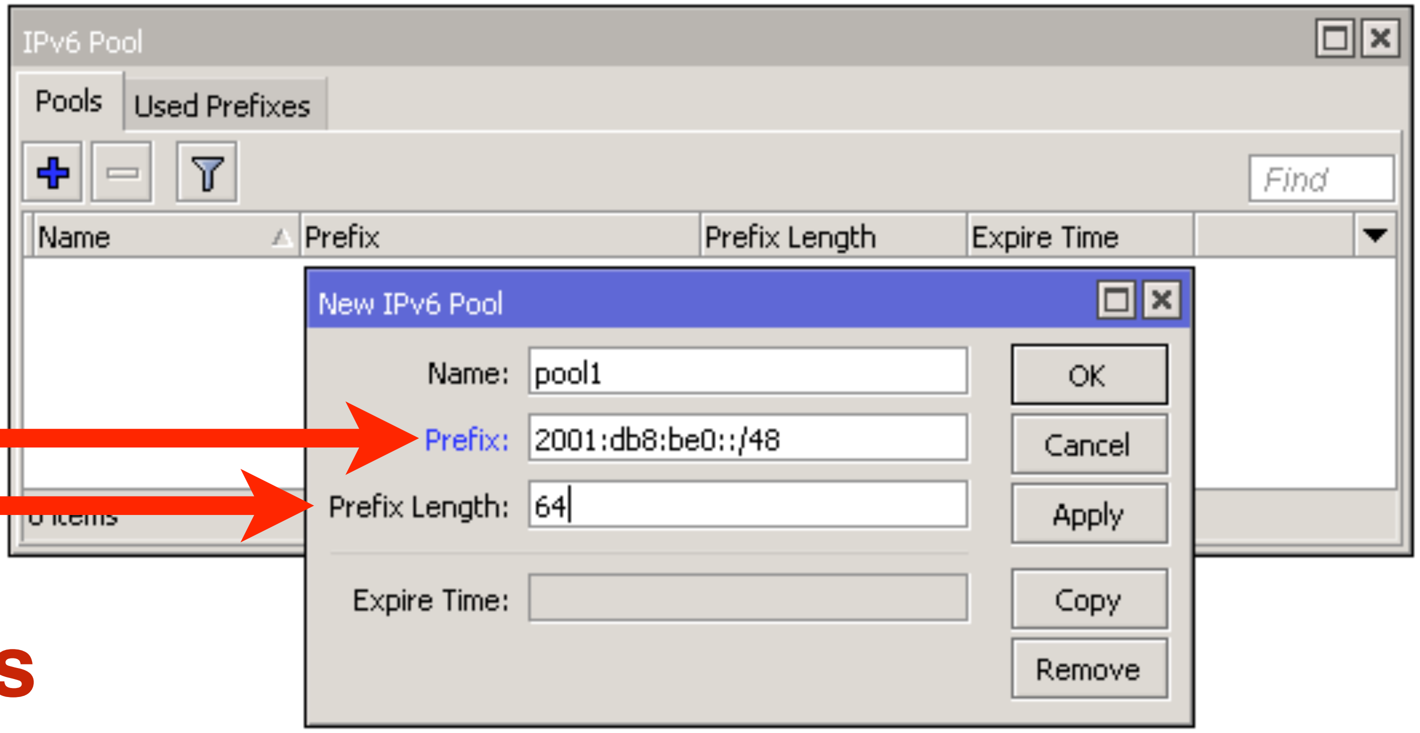
- Define range of IPv6 addresses that is used for SLAAC, DHCPv6 and PPP servers
- Groups IPv6 addresses for further usage
- A single configuration point for all features that assign IPv6 addresses to clients

MikroTik
MTCIPv6E

IPv6 Pool

Prefix that has been assigned to this router

Prefix that will be given to the clients



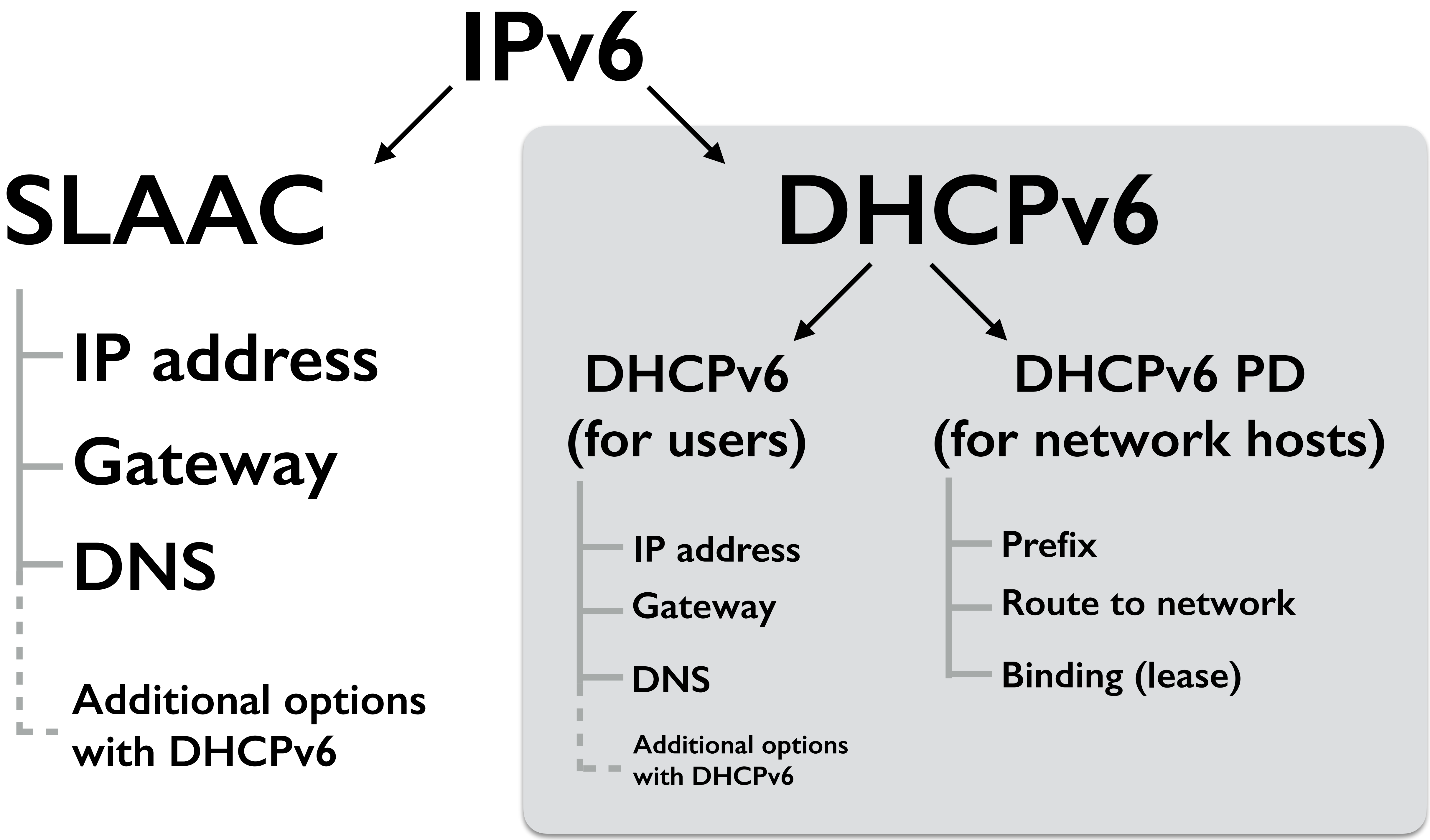
Name: pool1

Prefix: 2001:db8:be0::/48

Prefix Length: 64

Expire Time:

IPv6 → Pool→ '+'



IPv6

SLAAC

- IP address
- Gateway
- DNS
- Additional options with DHCPv6

DHCPv6

DHCPv6 (for users)

- IP address
- Gateway
- DNS
- Additional options with DHCPv6

DHCPv6 PD (for network hosts)

- Prefix
- Route to network
- Binding (lease)

DHCPv6 PD Client

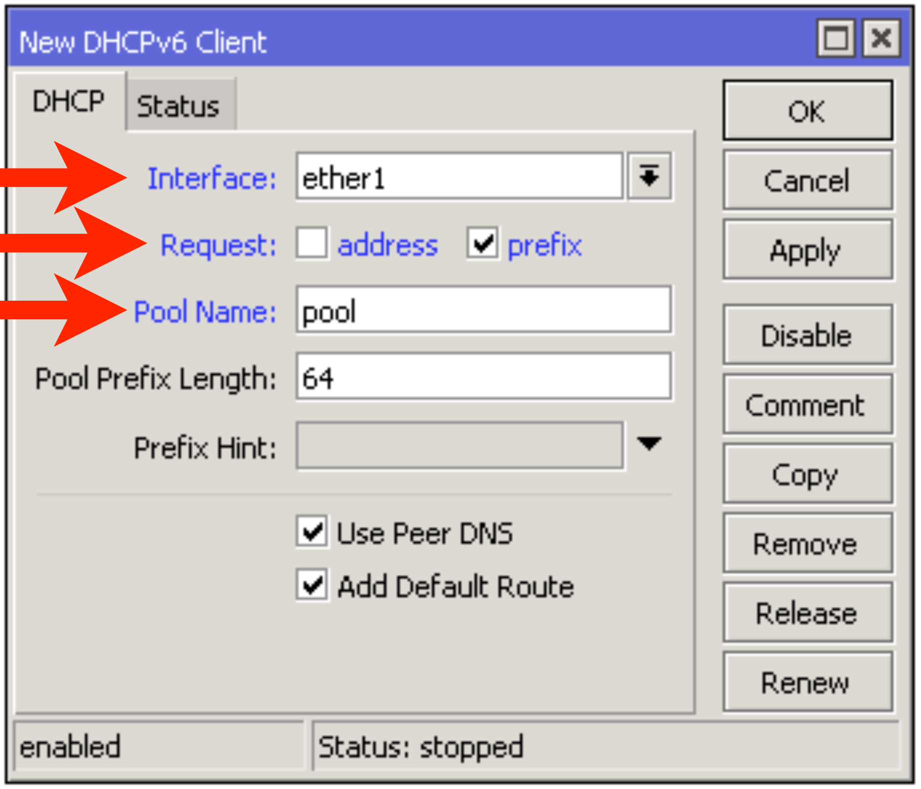
- For acquiring IPv6 prefix from a DHCPv6 PD server
- PD client sets route to the DHCPv6 PD server
- Afterwards the router can subdivide the acquired prefix and hand out to it’s clients

DHCPv6 PD Client

Interface on which to listen

Request a prefix

Pool name that will be created



New DHCPv6 Client

Interface: ether1

Request: □ address ☑ prefix

Pool Name: pool

Pool Prefix Length: 64

Prefix Hint:

☑ Use Peer DNS

☑ Add Default Route

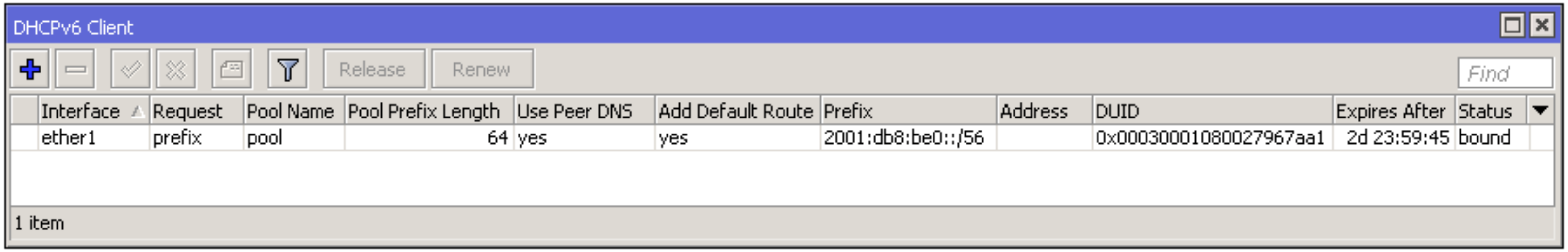
enabled

Status: stopped

IPv6 → DHCP Client→ ‘+’

MikroTik MTCIPv6E

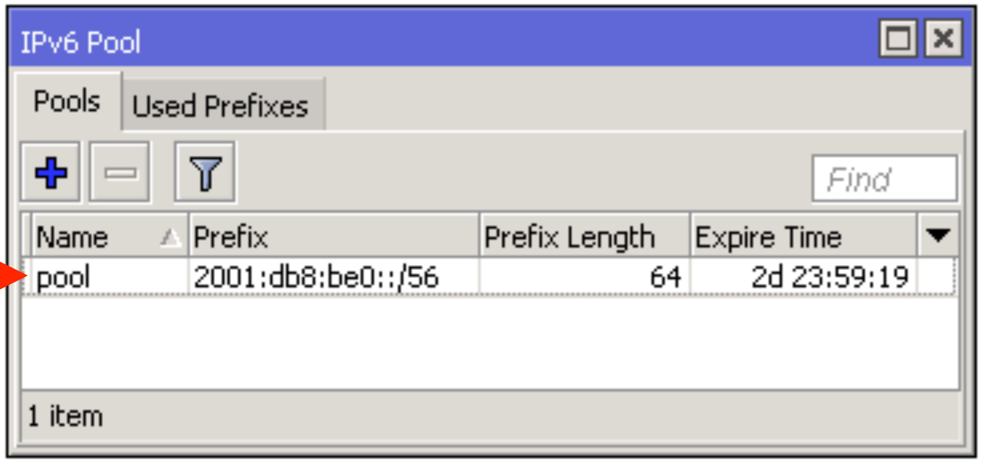
DHCPv6 PD Client



Interface	Request	Pool Name	Pool Prefix Length	Use Peer DNS	Add Default Route	Prefix	Address	DUID	Expires After	Status
ether1	prefix	pool	64	yes	yes	2001:db8:be0::/56		0x00030001080027967aa1	2d 23:59:45	bound

1 item

IPv6 → DHCP Client



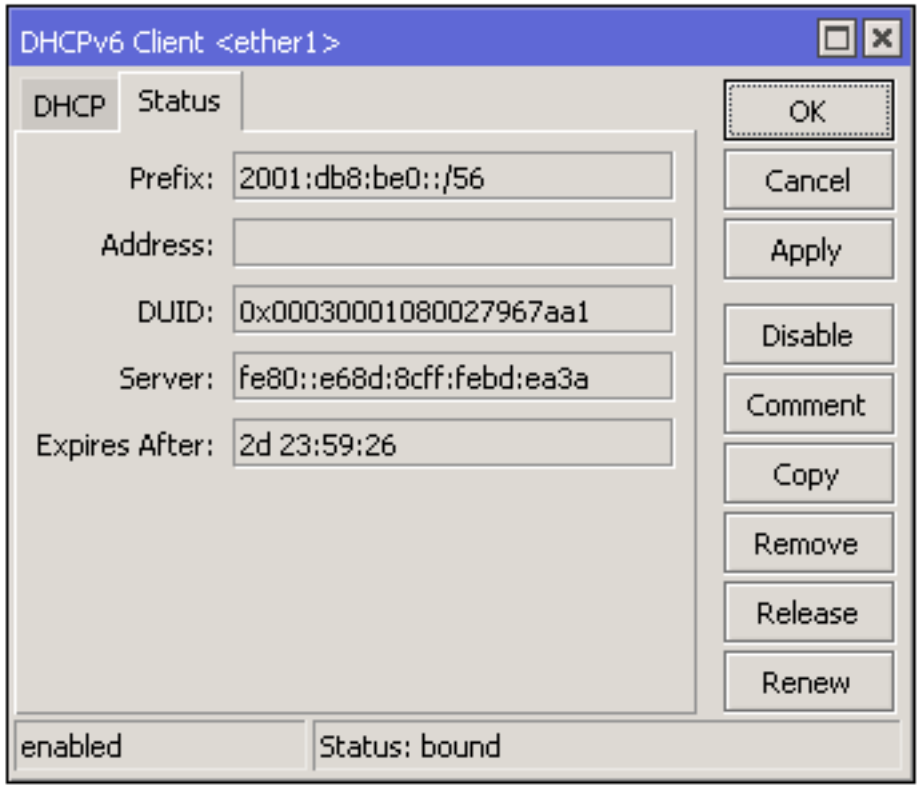
Name	Prefix	Prefix Length	Expire Time
pool	2001:db8:be0::/56	64	2d 23:59:19

1 item

Pool is created automatically by the PD Client

IPv6 → Pool

DHCPv6 PD Client



Prefix: 2001:db8:be0::/56

Address:

DUID: 0x00030001080027967aa1

Server: fe80::e68d:8cff:febd:ea3a

Expires After: 2d 23:59:26

enabled

Status: bound

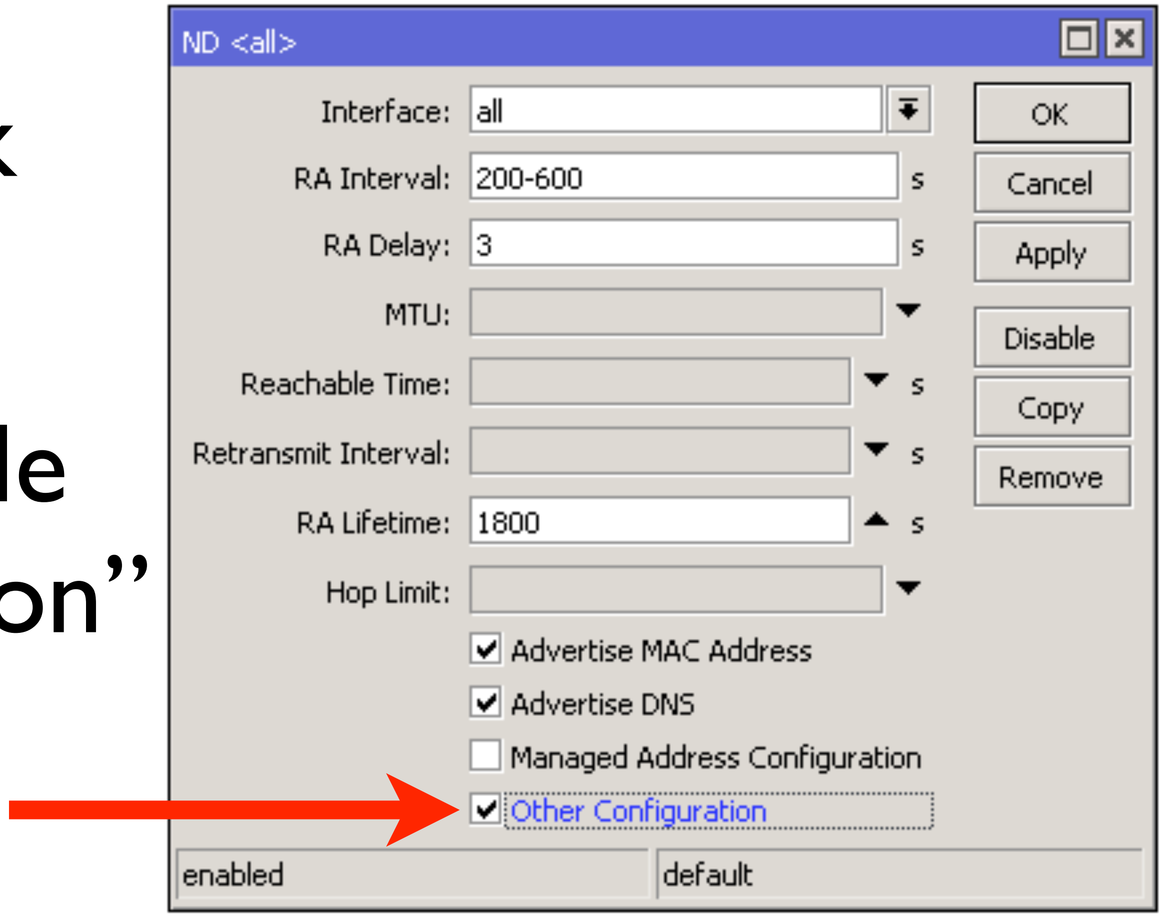
IPv6 → DHCP Client

DHCP unique identifier

- DHCP unique identifier (DUID). Each DHCP client and server has exactly one DUID
- DHCP servers use DUIDs to identify clients for the selection of configuration parameters
- DHCP clients use DUIDs to identify a server in messages where a server needs to be identified.

DHCPv6 PD Server

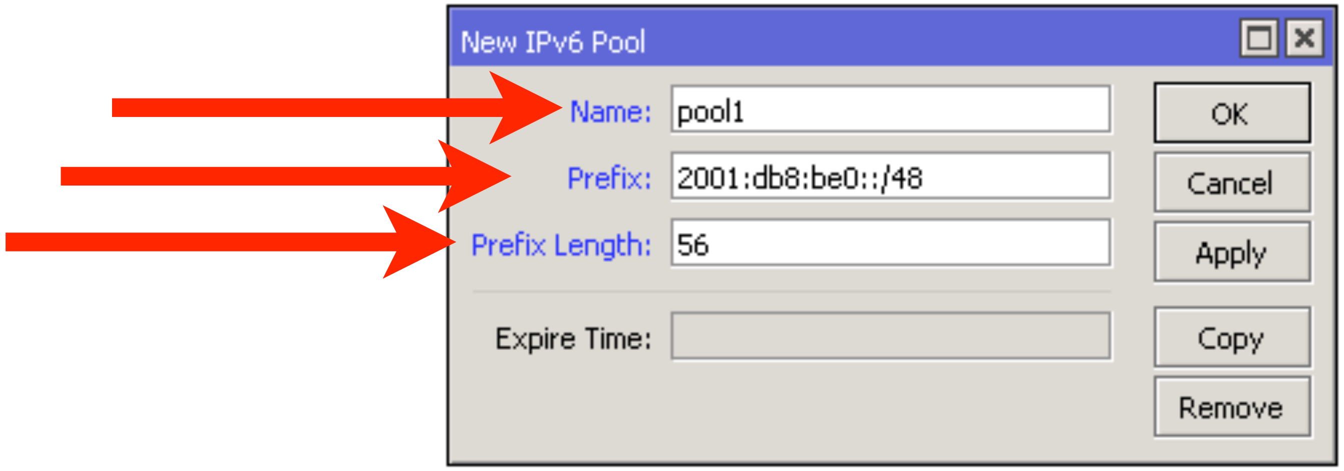
- DHCPv6 PD (prefix delegation)
- It is used to assign prefixes to network hosts (e.g. routers)
- To configure - enable “Other Configuration” in IPv6 → ND



Interface: all
RA Interval: 200-600 s
RA Delay: 3 s
MTU:
Reachable Time: s
Retransmit Interval: s
RA Lifetime: 1800 s
Hop Limit:
☑ Advertise MAC Address
☑ Advertise DNS
□ Managed Address Configuration
☑ Other Configuration
enabled default

IPv6 → ND → ‘all’

DHCPv6 PD Server



Name: pool1

Prefix: 2001:db8:be0::/48

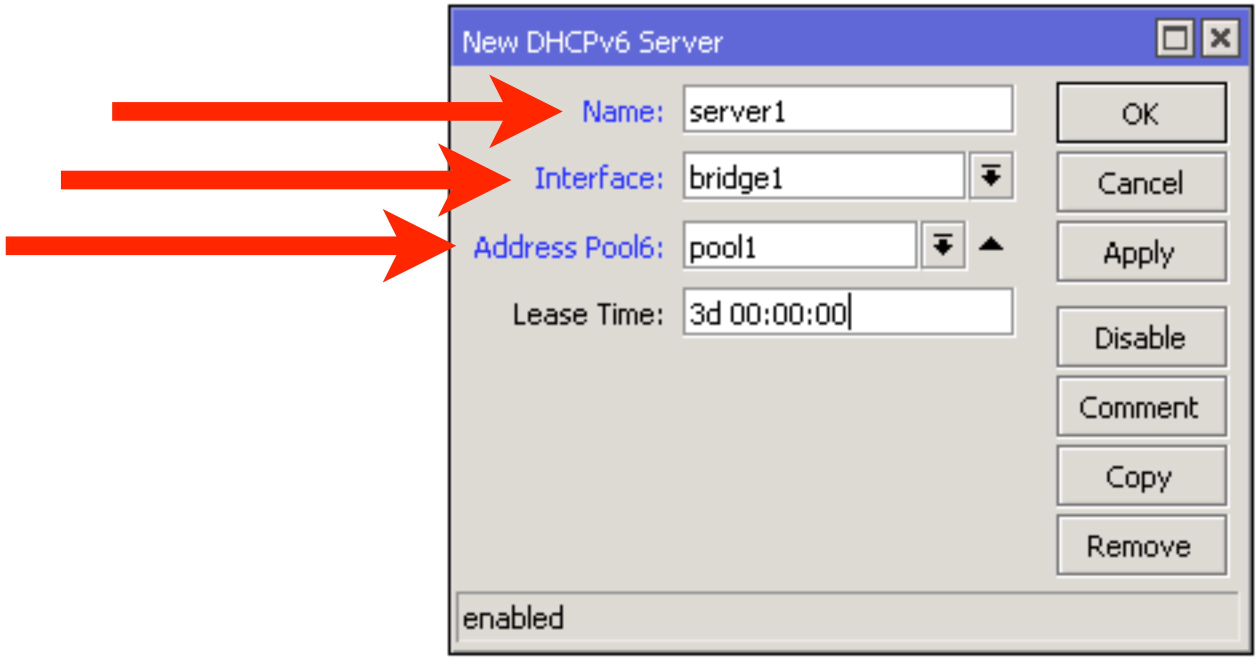
Prefix Length: 56

Expire Time:

IPv6 → Pool → ‘+’

- Add IPv6 address pool from which prefixes will be assigned
- Specify assigned prefix length

DHCPv6 PD Server

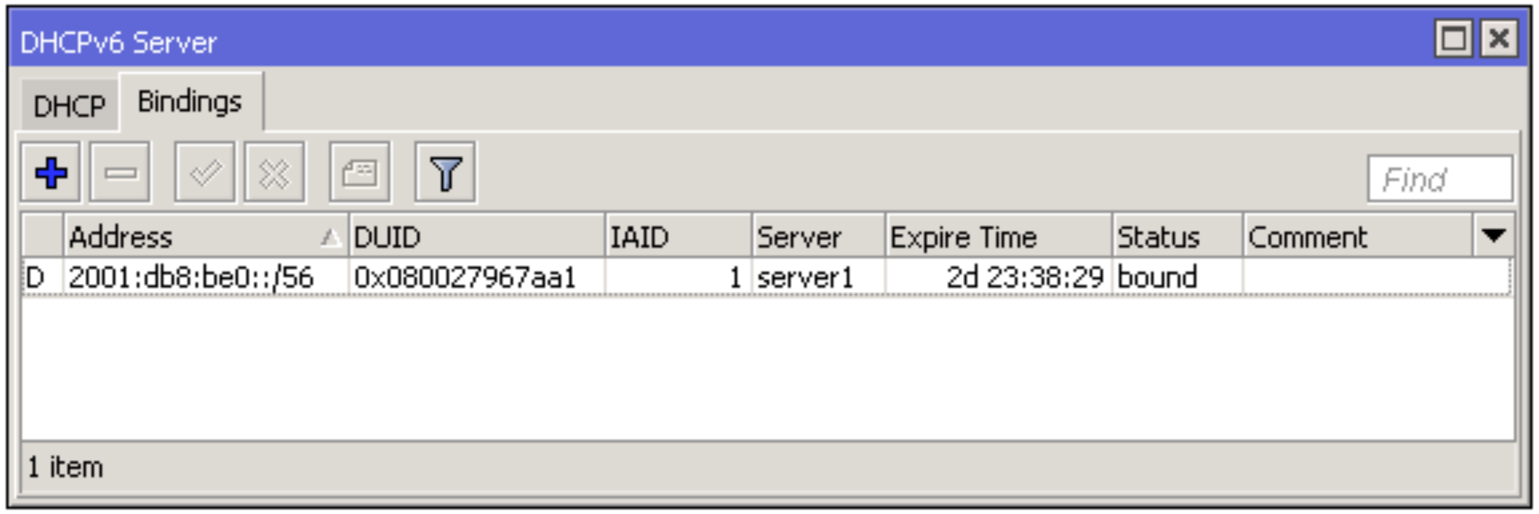


Name: server1
Interface: bridge1
Address Pool6: pool1
Lease Time: 3d 00:00:00
enabled

IPv6 → DHCPv6 → ‘+’

- Add new DHCP server on an interface
- Configure address pool from which addresses will be assigned

DHCPv6 PD Server



	Address	DUID	IAID	Server	Expire Time	Status	Comment
D	2001:db8:be0::/56	0x080027967aa1	1	server1	2d 23:38:29	bound	

1 item

IPv6 → DHCP Server → Bindings

- Assigned prefixes can be observed in bindings menu

MikroTik MTCIPv6E

DHCPv6 Client

- For acquiring IPv6 address from a DHCPv6 server
- Client can set default route to the DHCPv6 server
- Acquires DNS, NTP and other information

LAB

DHCPv6 PD

- Trainer will now configure DHCPv6 PD server on his router
- It will issue /60 prefixes
- Configure DHCPv6 PD client on your router
- Assign /64 prefix to your laptop via SLAAC

IPv6 Tunnels

- Currently RouterOS supports following IPv6 tunnels
  - IPIPv6
  - EoIPv6
  - GRE6
- Work in a similar way as IPv4 counterparts

LAB

IPIPv6

- Pair up with another student
- Create an IPIPv6 tunnel between your routers
  - On R1, set source address R1 public address, destination R2 public address
  - On R2, set source address R2 public address, destination R1 public address

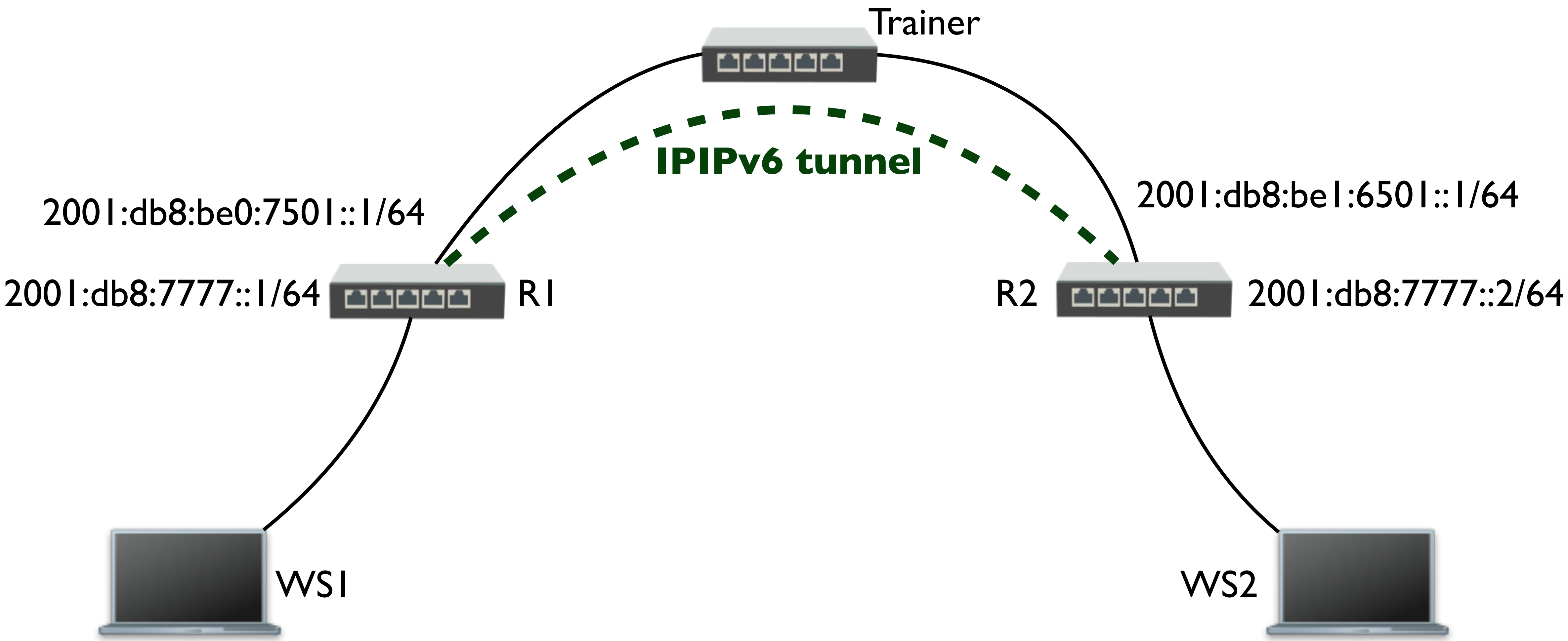
IPIPv6

LAB

- Assign arbitrary IPv6 addresses on R1 and R2 IPIPv6 tunnel interfaces
- Both from the same subnet, e.g.
  - 2001:db8:7777::1/64 (R1)
  - 2001:db8:7777::2/64 (R2)
- Ping tunnel addresses from your routers
- Observe the IPIPv6 interface traffic counters

IPIPv6

LAB



MikroTik
MTCIPv6E

LAB

IPIPv6

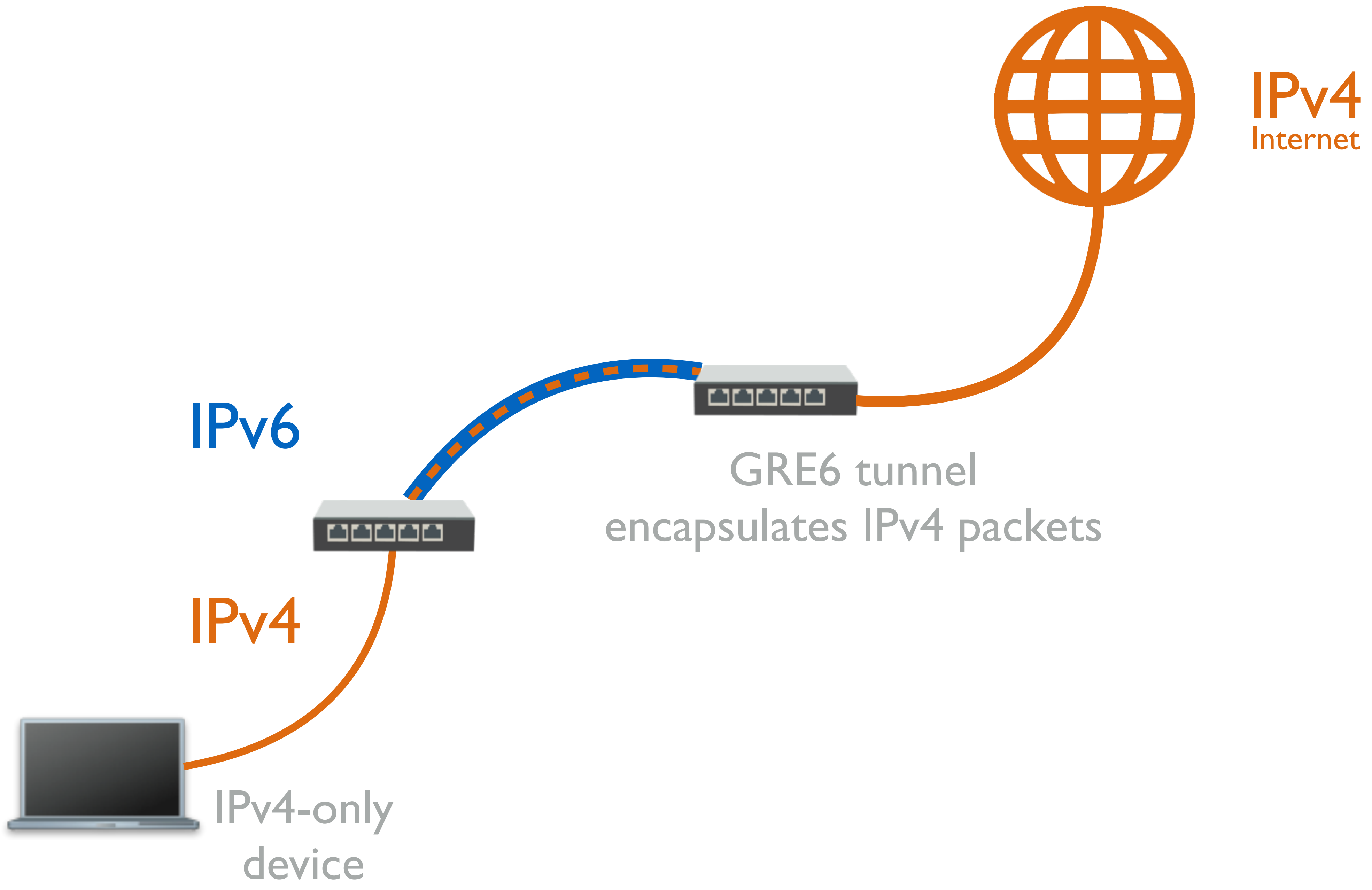
- Add IPsec secret on the IPIPv6 tunnel interface on both routers (the same secret phrase)
- Observe the IP → IPsec menu
- Now the IPIPv6 tunnel is encrypted

Optional LAB

IPIPv6

- Add static routes on R1 and R2 routers to your internal networks through the IPIPv6 tunnel
- Ping between laptops (WS1 and WS2)
- Now the communication between your laptops is going through the encrypted IPIPv6 tunnel

GRE6

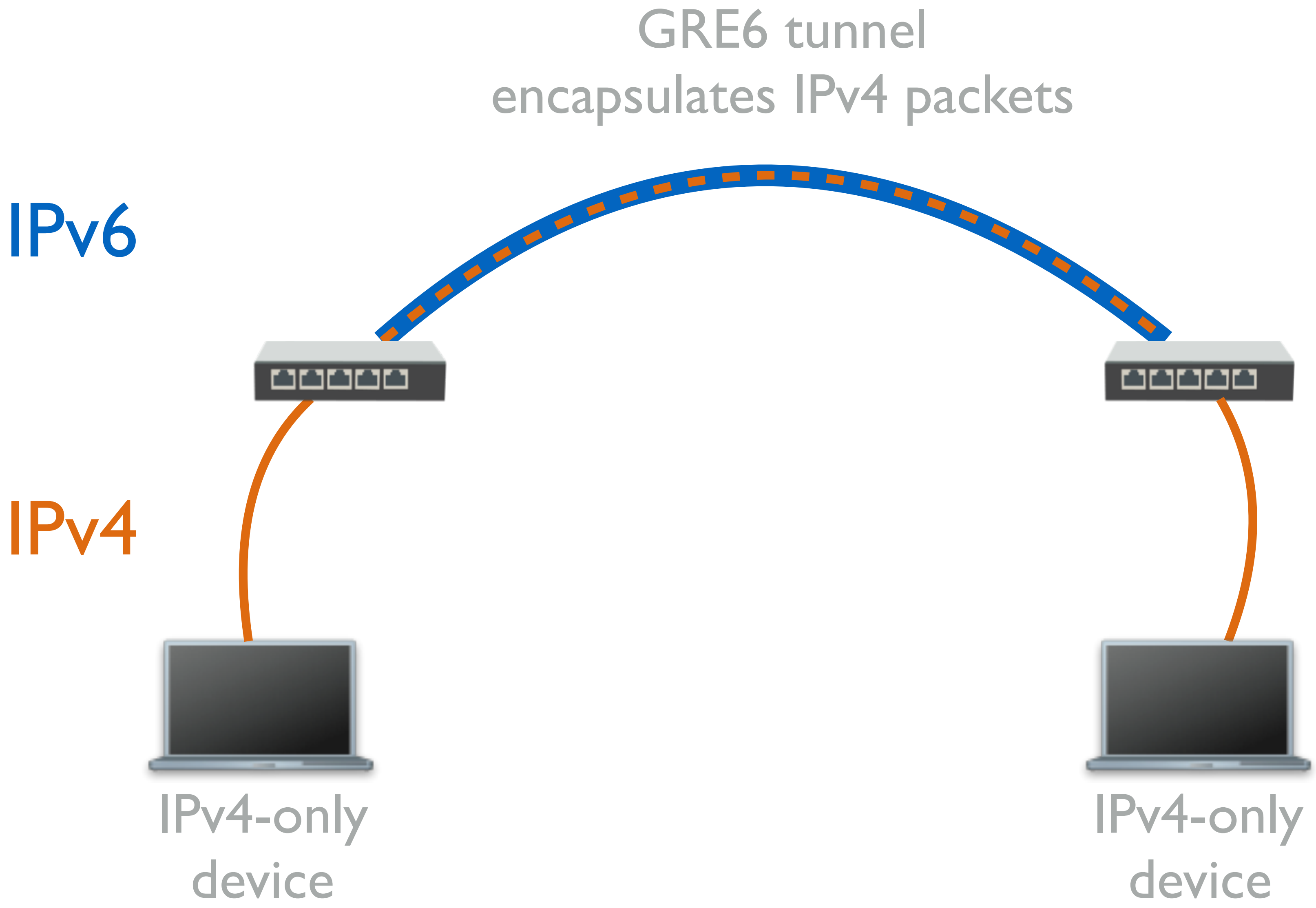

IPv4
Internet
IPv6
IPv4
GRE6 tunnel
encapsulates IPv4 packets
IPv4-only
device

GRE6

- In cases when you have IPv6-only network, but need to provide access to the Internet to a device which only supports IPv4
- IPv6 tunnels can be used to encapsulate IPv4 packets into IPv6 and tunnel them to a router which has IPv4 connectivity
- For example: GRE6 tunnel

Optional LAB

GRE6



Optional LAB

GRE6

- Pair up with another student
- Both create a GRE6 tunnel to the other's router
- Agree on IPv4 addresses you're going to use inside the tunnel and on your laptops
- If necessary create masquerade rules, bridge interfaces or create static routes accordingly

Optional LAB

GRE6

- Disable IPv6 on your laptops
- Set IPv4 addresses on your laptops - either manually or using DHCP
- Ping each others laptop IPv4 addresses
- The connection between your routers is IPv6-only, but now for backwards compatibility you have IPv4 connectivity

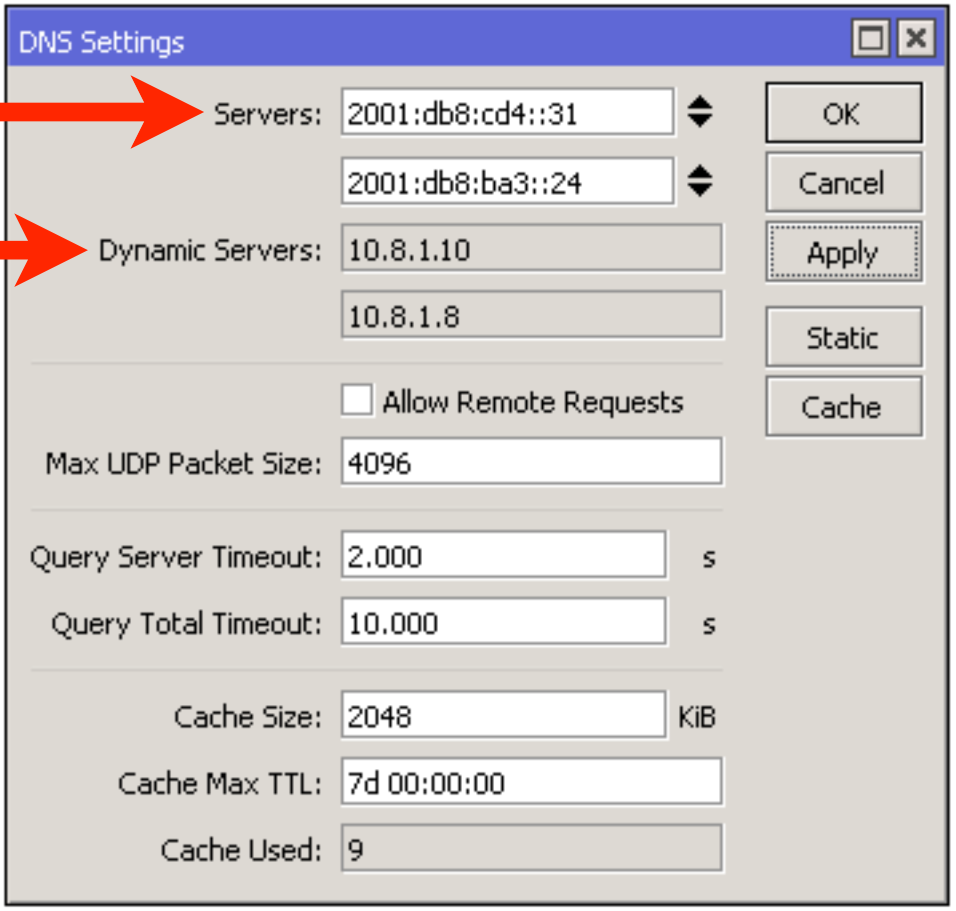
IP Version Agnostic

- IP → DNS supports both IPv4 and IPv6 addresses
- Both for DNS servers and static entries

IP DNS

IPv6 DNS servers

Dynamically acquired IPv4 DNS servers



DNS Settings

Servers: 2001:db8:cd4::31

2001:db8:ba3::24

Dynamic Servers: 10.8.1.10

10.8.1.8

□ Allow Remote Requests

Max UDP Packet Size: 4096

Query Server Timeout: 2.000 s

Query Total Timeout: 10.000 s

Cache Size: 2048 KiB

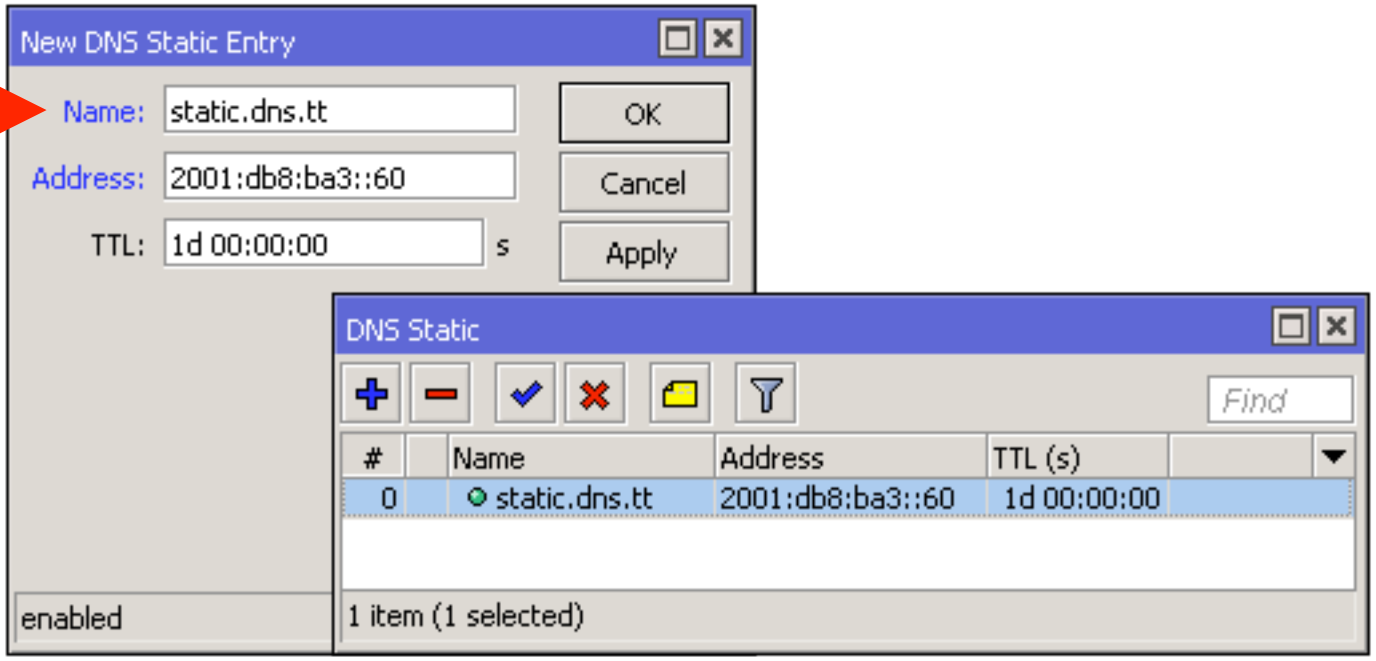
Cache Max TTL: 7d 00:00:00

Cache Used: 9

IP → DNS

Static DNS

New static IPv6 DNS entry (AAAA)



New DNS Static Entry

Name: static.dns.tt

Address: 2001:db8:ba3::60

TTL: 1d 00:00:00 s

enabled

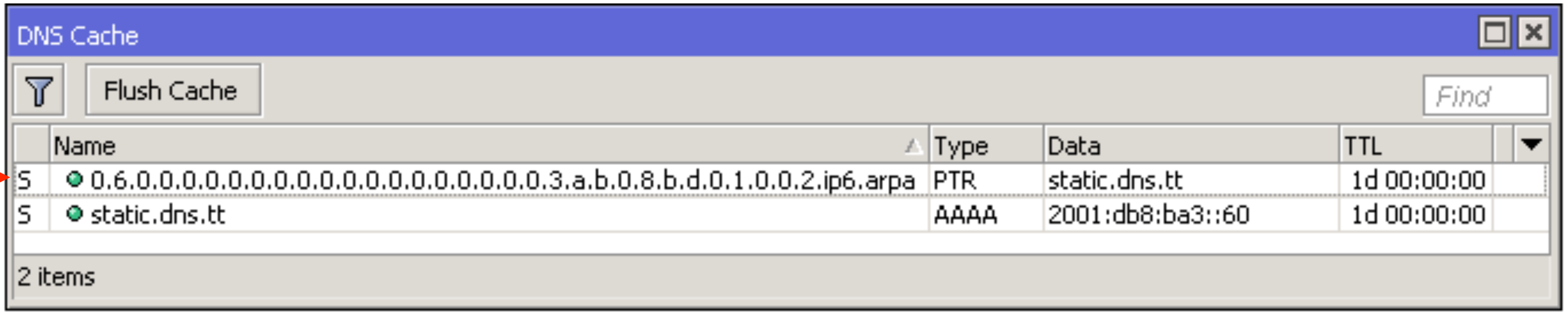
DNS Static

#	Name	Address	TTL (s)
0	static.dns.tt	2001:db8:ba3::60	1d 00:00:00

1 item (1 selected)

IP → DNS → Static

IPv6 reverse DNS entry



DNS Cache

	Name	Type	Data	TTL
S	0.6.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.3.a.b.0.8.b.d.0.1.0.0.2.ip6.arpa	PTR	static.dns.tt	1d 00:00:00
S	static.dns.tt	AAAA	2001:db8:ba3::60	1d 00:00:00

2 items

IP → DNS → Cache

MikroTik MTCIPv6E

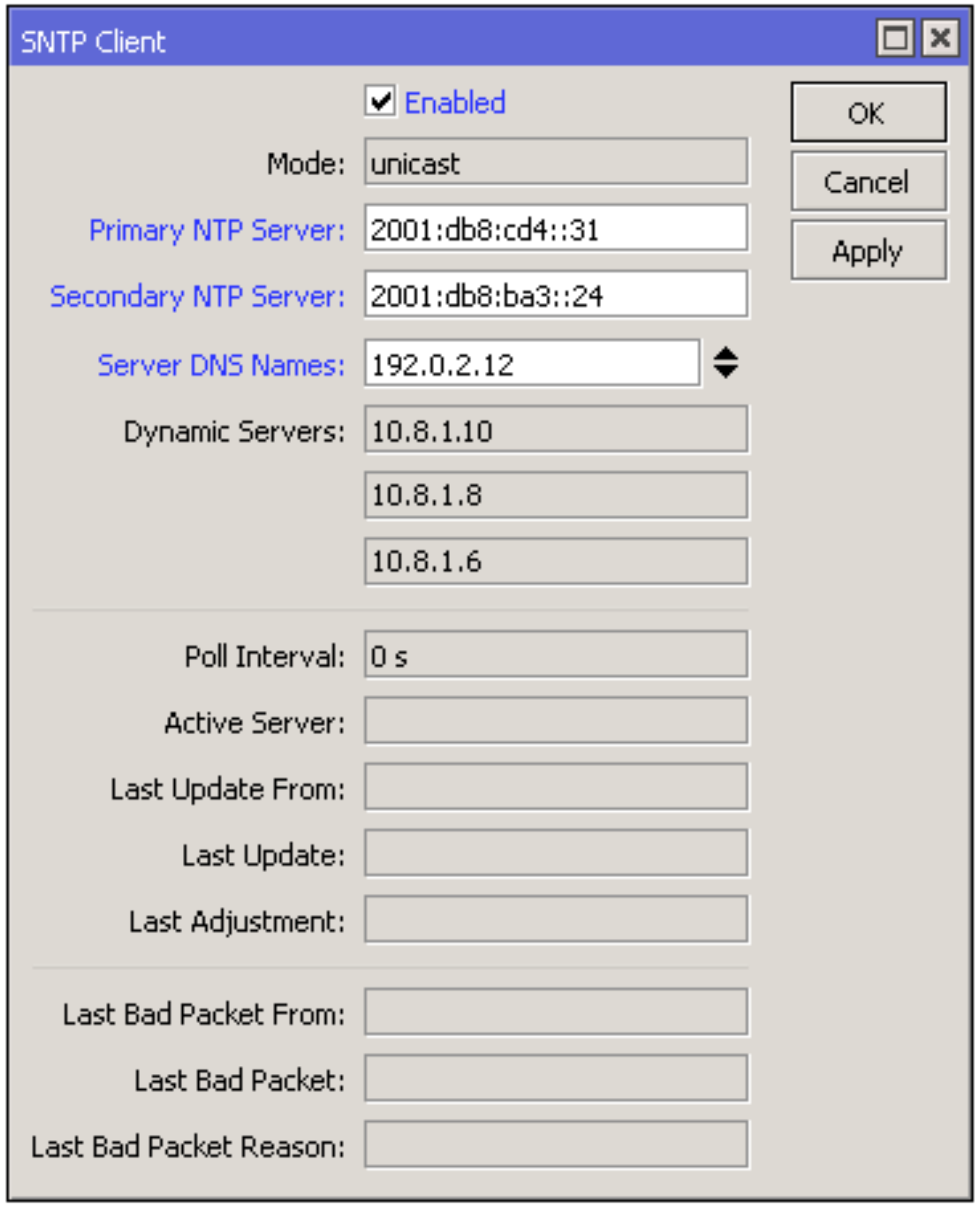
IPv6 Reverse DNS

- Entry consists or 32 values separated by dots
- Zeros are not omitted
- ip6.arpa. is added at the end

AAAA	2001:db8:3:4:5:6:7:8
PTR	8.0.0.0.7.0.0.0.6.0.0.0.5.0.0.0.4.0.0.0.3.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

NTP

- NTP client supports both IPv4 and IPv6 addresses



☑ Enabled

Mode:	unicast
Primary NTP Server:	2001:db8:cd4::31
Secondary NTP Server:	2001:db8:ba3::24
Server DNS Names:	192.0.2.12
Dynamic Servers:	10.8.1.10
	10.8.1.8
	10.8.1.6
Poll Interval:	0 s
Active Server:	
Last Update From:	
Last Update:	
Last Adjustment:	
Last Bad Packet From:	
Last Bad Packet:	
Last Bad Packet Reason:	

System → SNTP Client

PPP IPv6 Support

- PPP supports prefix delegation (PD) to PPP clients
- Use PPP Profile DHCPv6 PD Pool option to specify pools that will be assigned to clients
- If a RouterOS device is a client, a DHCPv6 PD client must be configured on PPP client interface

Optional LAB

PPP IPv6 Support

- Pair up with another student
- Decide who will create the server part and who the client part

PPPoE server Server part configuration

PPPoE client Client part configuration

Optional LAB

PPP IPv6 Support

- To configure PPPoE server to assign IPv6 prefix to a RouterOS client following steps have to be done:

1. Create IP Pool from which prefixes will be assigned
2. Create a PPP profile which will be used for IPv6
3. Create a PPPoE server using the profile created in previous step

Optional LAB

PPP IPv6 Support

- To configure RouterOS PPPoE client to receive IPv6 prefix following steps have to be done:

4. Create a PPPoE client

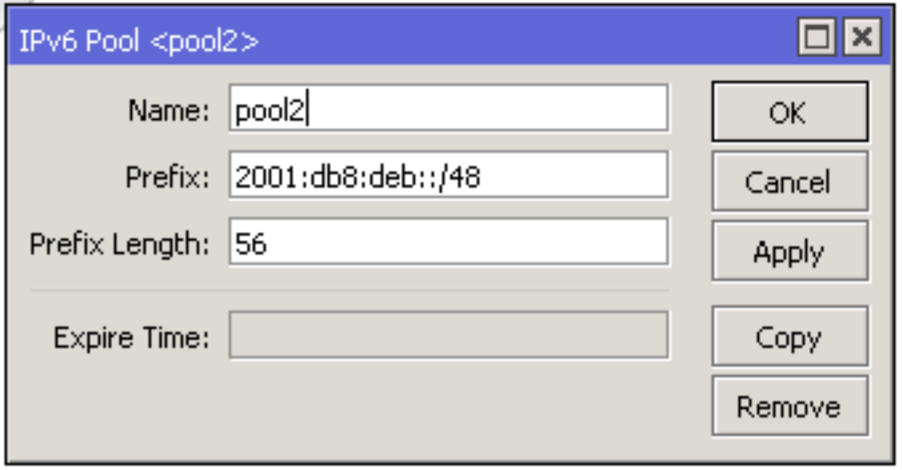
5. Configure IPv6 DHCP PD client on the PPPoE client interface

Optional LAB

PPP IPv6 Support

- To configure PPPoE server to assign IPv6 prefix to a RouterOS client following steps have to be done:

1


IPv6 Pool <pool2>

Name: pool2

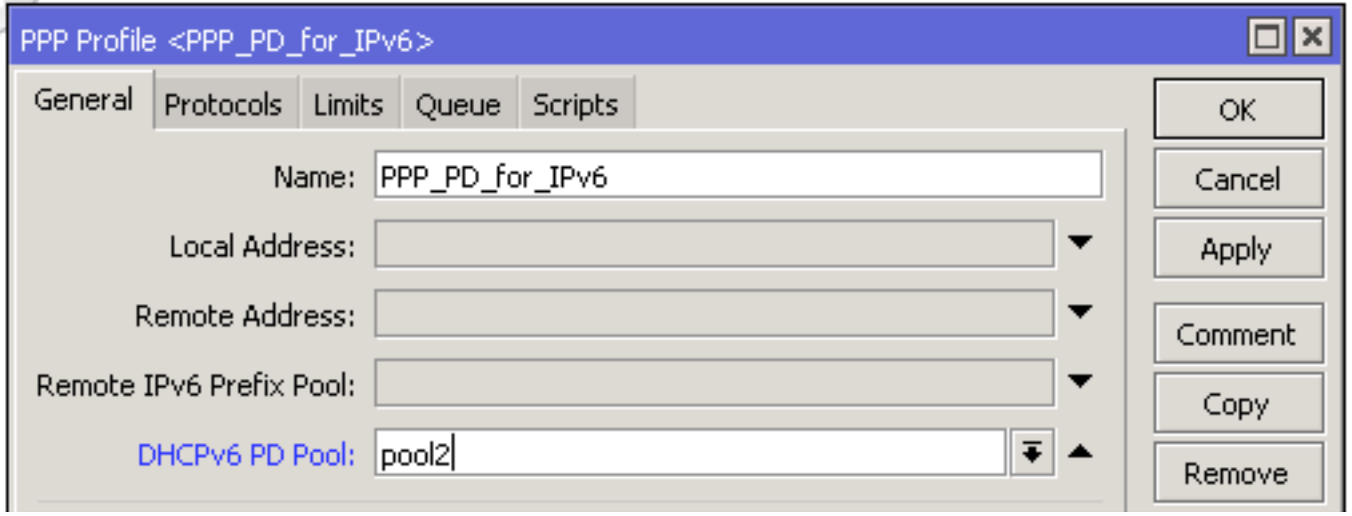
Prefix: 2001:db8:deb::/48

Prefix Length: 56

Expire Time:

IPv6 → Pool → '+'

2


PPP Profile <PPP_PD_for_IPv6>

Name: PPP_PD_for_IPv6

Local Address:

Remote Address:

Remote IPv6 Prefix Pool:

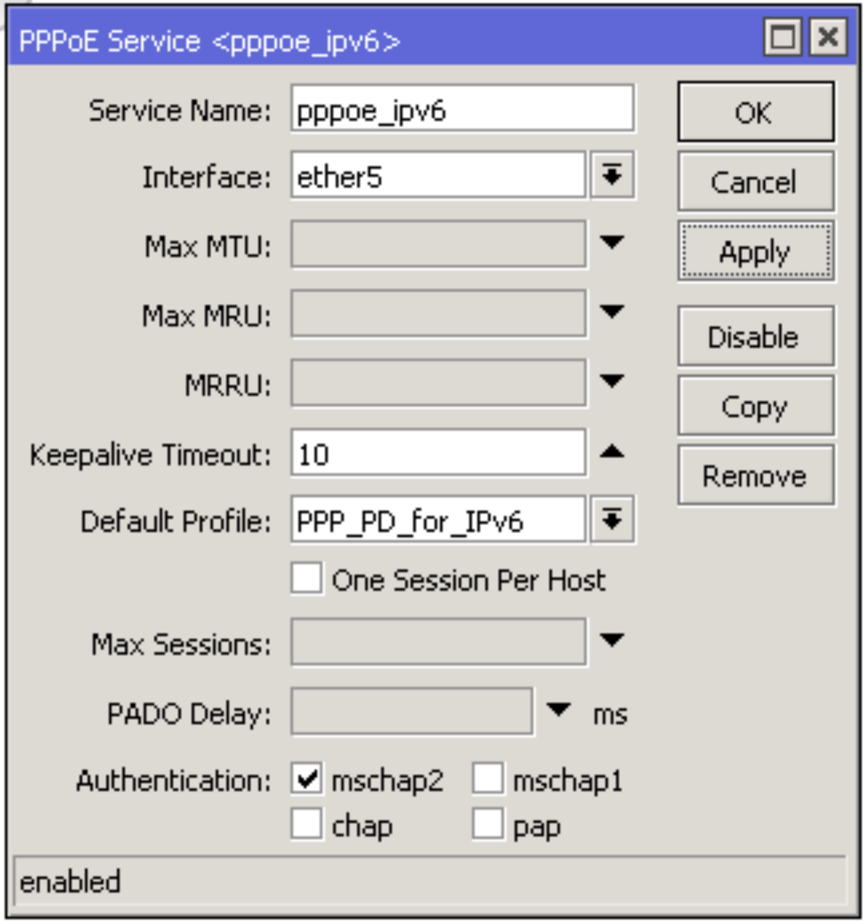
DHCPv6 PD Pool: pool2

PPP → Profiles → '+'

Optional LAB

PPP IPv6 Support

3



PPPoE Service <pppoe_ipv6>

Service Name: pppoe_ipv6

Interface: ether5

Max MTU:

Max MRU:

MRRU:

Keepalive Timeout: 10

Default Profile: PPP_PD_for_IPv6

□ One Session Per Host

Max Sessions:

PADO Delay: ms

Authentication: ☑ mschap2 □ mschap1 □ chap □ pap

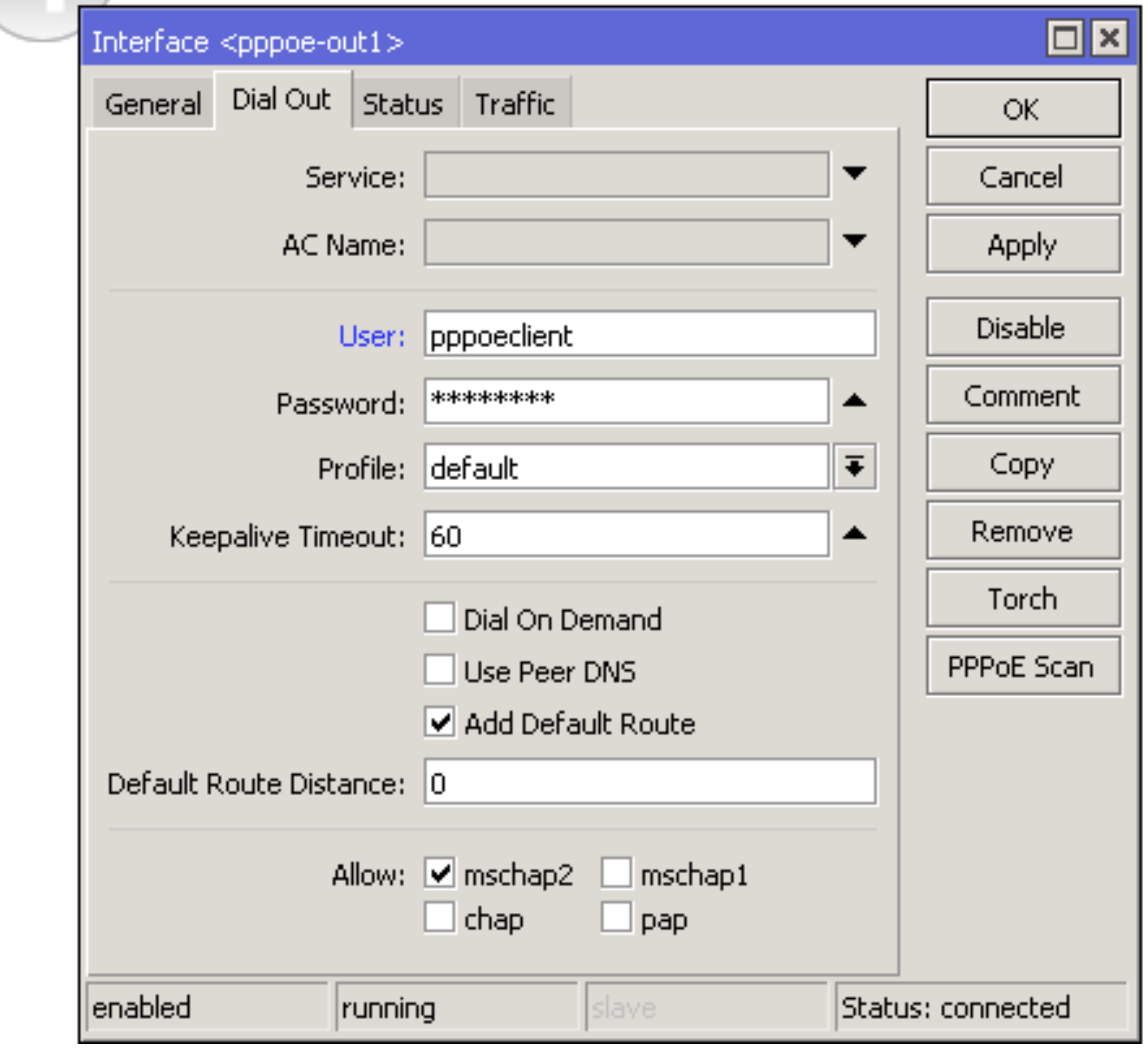
enabled

PPP → PPPoE Servers → '+'

Optional LAB

PPP IPv6 Support

4


Interface <pppoe-out1>

General | Dial Out | Status | Traffic

Service:

AC Name:

User: pppoeclient

Password: ********

Profile: default

Keepalive Timeout: 60

□ Dial On Demand

□ Use Peer DNS

☑ Add Default Route

Default Route Distance: 0

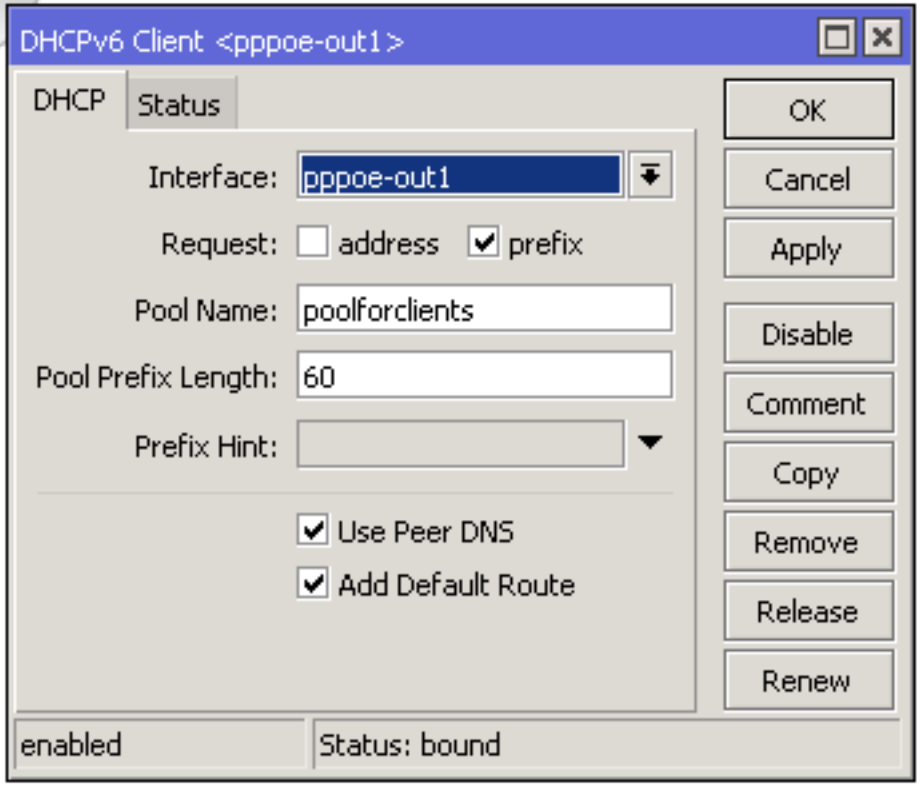
Allow: ☑ mschap2 □ mschap1

□ chap □ pap

enabled | running | slave | Status: connected

PPP → Interface → ‘+’ → PPPoE Client

5


DHCPv6 Client <pppoe-out1>

DHCP | Status

Interface: pppoe-out1

Request: □ address ☑ prefix

Pool Name: poolforclients

Pool Prefix Length: 60

Prefix Hint:

☑ Use Peer DNS

☑ Add Default Route

enabled | Status: bound

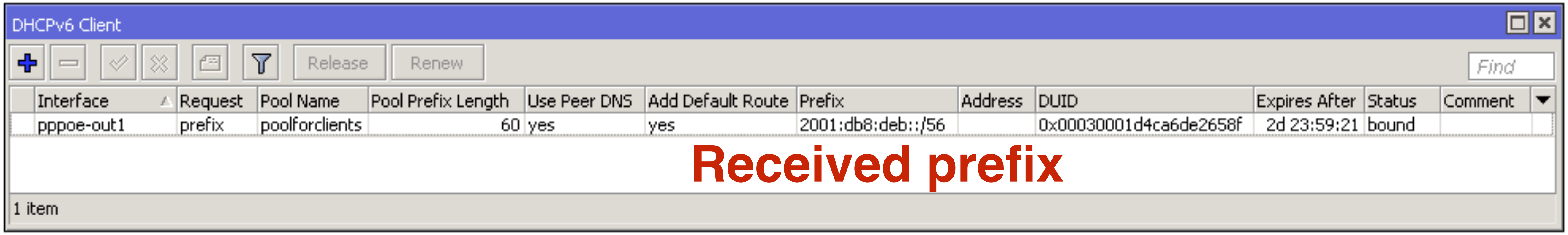
IPv6 → DHCP Client → ‘+’

MikroTik
MTCIPv6E

PPPoE client

Optional LAB

PPP IPv6 Support



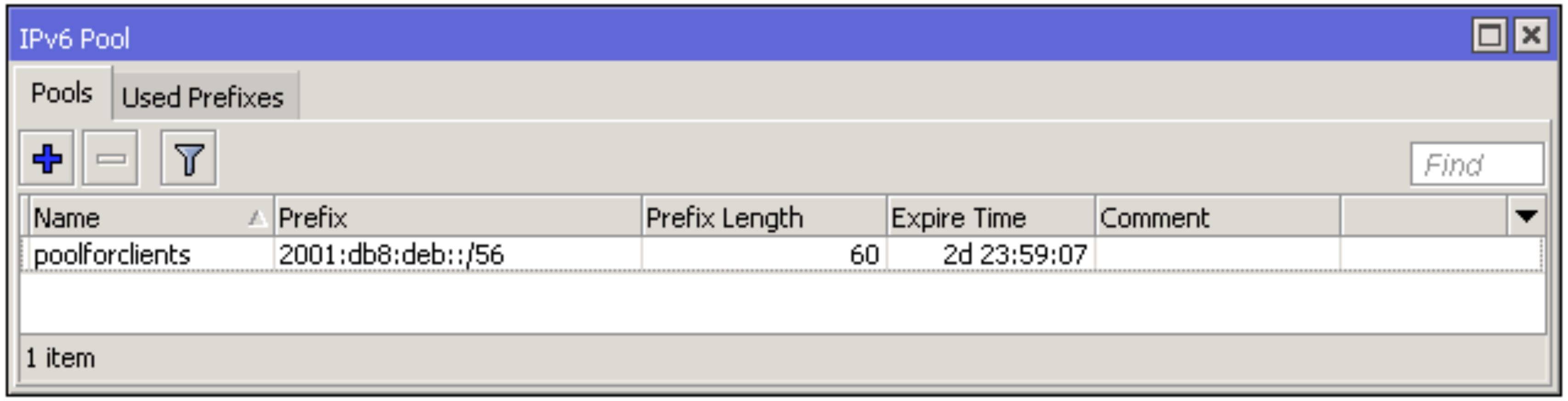
Interface	Request	Pool Name	Pool Prefix Length	Use Peer DNS	Add Default Route	Prefix	Address	DUID	Expires After	Status	Comment
pppoe-out1	prefix	poolforclients	60	yes	yes	2001:db8:deb::/56		0x00030001d4ca6de2658f	2d 23:59:21	bound	

1 item

Received prefix

IPv6 → DHCP Client

New pool from received prefix



Name	Prefix	Prefix Length	Expire Time	Comment
poolforclients	2001:db8:deb::/56	60	2d 23:59:07	

1 item

IPv6 → Pool

Optional LAB

PPP IPv6 Support

- Now the PPPoE client RouterOS can issue prefixes to it’s clients via SLAAC or DHCPv6 PD

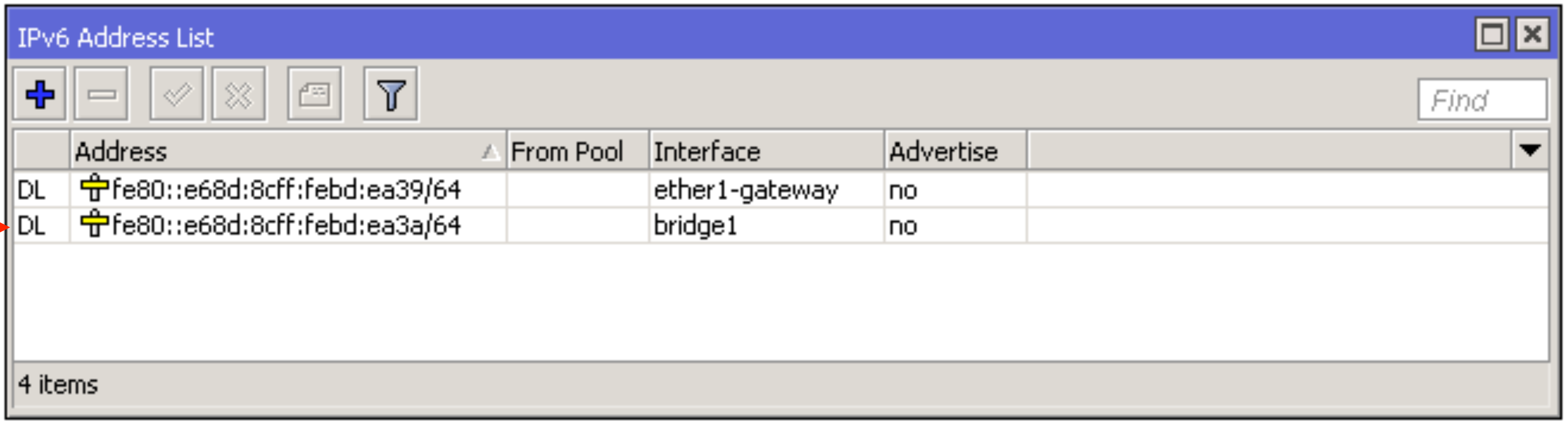
Routing

- IPv6 global routing works similar as in IPv4
- Concepts are the same
- Static and/or dynamic routing can be used
- Dynamic routing protocols such as OSPF (v3), RIP (ng), BGP support IPv6

Routing

- IPv6 link-local addresses can be used to communicate between hosts
- There’s no need for global IPv6 addresses
- Fully functional internal IPv6 network can be created with LL addresses

Routing



	Address	From Pool	Interface	Advertise
DL	fe80::e68d:8cff:febd:ea39/64		ether1-gateway	no
DL	fe80::e68d:8cff:febd:ea3a/64		bridge1	no

4 items

Bridge interface LL address

IPv6 → Addresses

```
$ ping6 fe80::e68d:8cff:febd:ea3a%en6
PING6(56=40+8+8 bytes) fe80::2e0:4cff:fe68:33a%en6 --> fe80::e68d:8cff:febd:ea3a%en6
16 bytes from fe80::e68d:8cff:febd:ea3a%en6, icmp_seq=0 hlim=64 time=0.376 ms
16 bytes from fe80::e68d:8cff:febd:ea3a%en6, icmp_seq=1 hlim=64 time=0.498 ms
16 bytes from fe80::e68d:8cff:febd:ea3a%en6, icmp_seq=2 hlim=64 time=0.502 ms

--- fe80::e68d:8cff:febd:ea3a%en6 ping6 statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.376/0.459/0.502/0.058 ms
```

Ping router’s LL address from macOS. Have to specify interface!

Routing

```
en6: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
	options=4<VLAN_MTU>
	ether 00:e0:4c:68:03:3a
	inet6 fe80::2e0:4cff:fe68:33a%en6 prefixlen 64 scopeid 0x9
	nd6 options=1<PERFORMNUD>
	media: autoselect (1000baseT <full-duplex>)
	status: active
```

Computer LL address

```
[admin@3rd_fl_Kaspars] > /ping fe80::2e0:4cff:fe68:33a interface=bridge1
  SEQ HOST                                     SIZE TTL TIME  STATUS
    0 fe80::2e0:4cff:fe68:33a                    56  64 0ms   echo reply
    1 fe80::2e0:4cff:fe68:33a                    56  64 0ms   echo reply
    2 fe80::2e0:4cff:fe68:33a                    56  64 0ms   echo reply
    sent=3 received=3 packet-loss=0% min-rtt=0ms avg-rtt=0ms max-rtt=0ms
```

Ping from RouterOS to computer’s LL address

Not Yet

- Several of popular RouterOS features which are available for IPv4 are not available using IPv6:
    - NAT
    - HotSpot
    - RADIUS integration
    - Policy routing
    - DHCPv6 server

IPv6 NAT

- NAT was originally used for ease of rerouting traffic in IP networks without renumbering every host
- It has become a popular tool in conserving global IPv4 addresses
- There are 2^{128} IPv6 addresses vs 2^{32} IPv4

IPv6 NAT

- Each IPv6 enabled host can have a global IPv6 address
- In most common cases there’s usually no need for IPv6 NAT
- NAT is not a security feature, firewall is needed also for IPv4

IPv6 NAT

- Companies can apply for Provider Independent (PI) address space
- In case a provider has to be changed, IP’s can remain the same

IPv6 HotSpot

- RouterOS current HotSpot implementation does not support IPv6
- MikroTik is planning to introduce a HotSpot version which will support IPv6
  - No specific timeframe can be given yet

RADIUS Integration

- Currently RouterOS services does not yet fully support RADIUS IPv6 arguments
- MikroTik is planning to implement IPv6 support for RouterOS services using RADIUS
  - No specific timeframe can be given yet

Policy Routing

- Currently RouterOS policy routing does not support IPv6
- MikroTik is planning to implement IPv6 support for policy routing
  - No specific timeframe can be given yet

DHCPv6 server

- Currently RouterOS supports
  - DHCPv6 PD (prefix delegation)
  - SLAAC
- It is not possible to assign custom size prefixes smaller than /64 from RouterOS

Tools

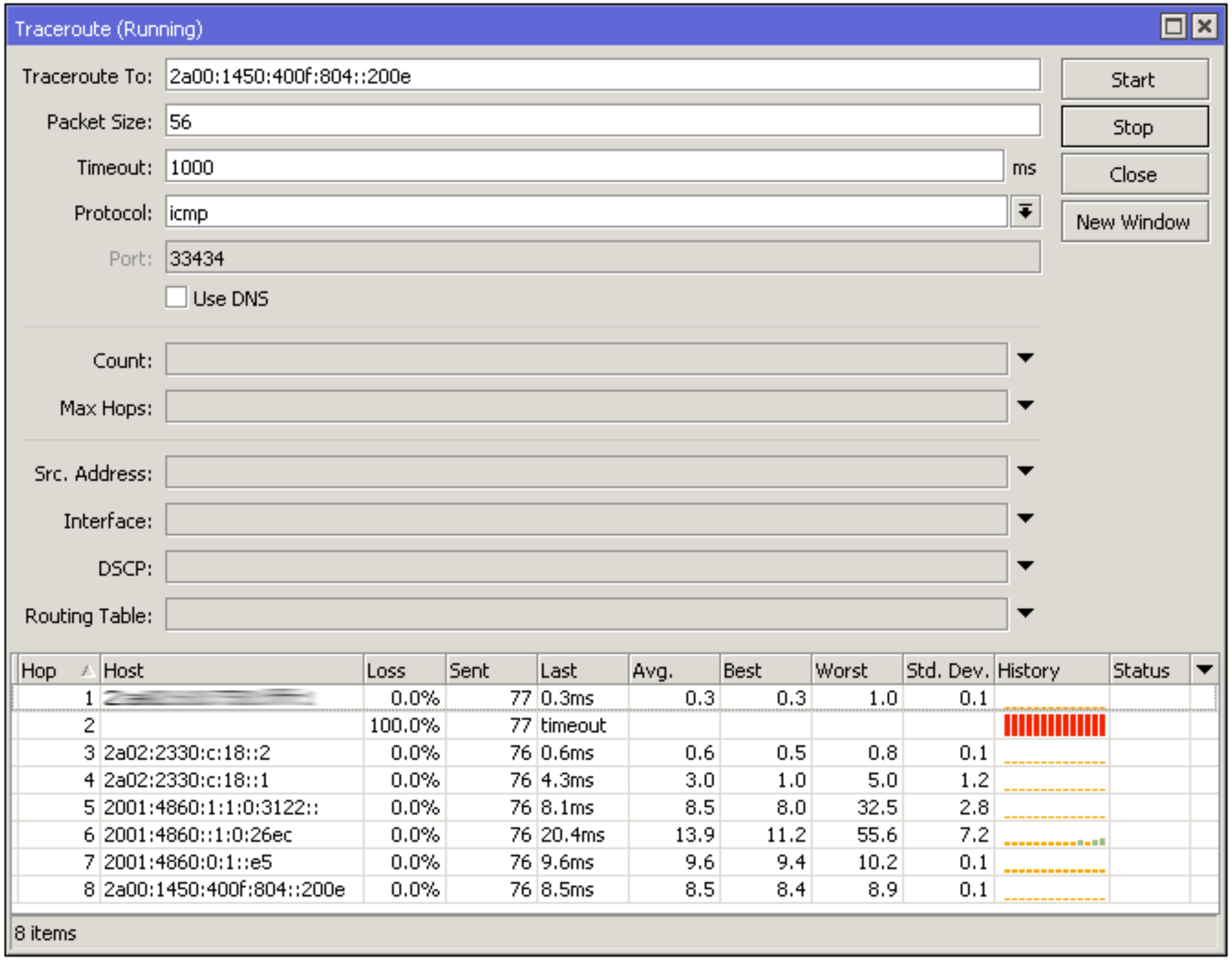
- Most of RouterOS tools support both IPv4 and IPv6 addresses, for example:
  - Ping
  - Traceroute
  - Torch
  - Traffic generator
  - E-mail
  - Netwatch
  - Traffic flow

Ping

```
[admin@MikroTik] > /ping 2a00:1450:400f:807::200e
  SEQ HOST                                     SIZE TTL TIME  STATUS
    0 2a00:1450:400f:807::200e                   56  57 10ms  echo reply
    1 2a00:1450:400f:807::200e                   56  57 9ms   echo reply
    2 2a00:1450:400f:807::200e                   56  57 9ms   echo reply
    sent=3 received=3 packet-loss=0% min-rtt=9ms avg-rtt=9ms max-rtt=10ms
```

- Ping tool supports both IPv4 and IPv6 addresses

Traceroute



Traceroute (Running)

Traceroute To: 2a00:1450:400f:804::200e

Packet Size: 56

Timeout: 1000 ms

Protocol: icmp

Port: 33434

□ Use DNS

Count:

Max Hops:

Src. Address:

Interface:

DSCP:

Routing Table:

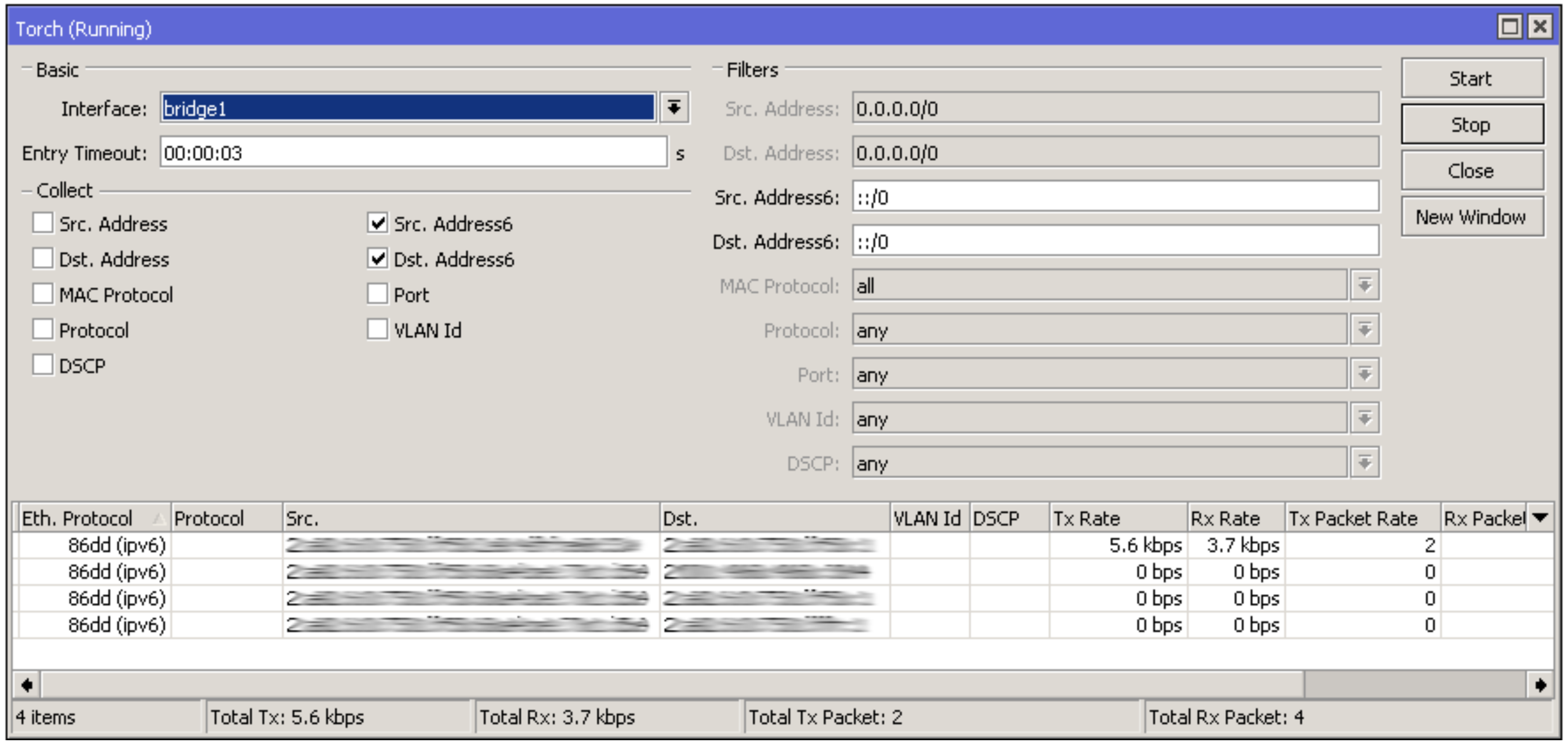
Hop	Host	Loss	Sent	Last	Avg.	Best	Worst	Std. Dev.	History	Status
1	[redacted]	0.0%	77	0.3ms	0.3	0.3	1.0	0.1		
2		100.0%	77	timeout						
3	2a02:2330:c:18::2	0.0%	76	0.6ms	0.6	0.5	0.8	0.1		
4	2a02:2330:c:18::1	0.0%	76	4.3ms	3.0	1.0	5.0	1.2		
5	2001:4860:1:1:0:3122::	0.0%	76	8.1ms	8.5	8.0	32.5	2.8		
6	2001:4860::1:0:26ec	0.0%	76	20.4ms	13.9	11.2	55.6	7.2		
7	2001:4860:0:1::e5	0.0%	76	9.6ms	9.6	9.4	10.2	0.1		
8	2a00:1450:400f:804::200e	0.0%	76	8.5ms	8.5	8.4	8.9	0.1		

8 items

Tools → Traceroute

Torch

- Torch tool supports capturing both IPv4 and IPv6 traffic



Interface: bridge1

Entry Timeout: 00:00:03 s

Collect

☐ Src. Address
☐ Dst. Address
☐ MAC Protocol
☐ Protocol
☐ DSCP
☑ Src. Address6
☑ Dst. Address6
☐ Port
☐ VLAN Id

Src. Address: 0.0.0.0/0

Dst. Address: 0.0.0.0/0

Src. Address6: ::/0

Dst. Address6: ::/0

MAC Protocol: all

Protocol: any

Port: any

VLAN Id: any

DSCP: any

Eth. Protocol	Protocol	Src.	Dst.	VLAN Id	DSCP	Tx Rate	Rx Rate	Tx Packet Rate	Rx Packet
86dd (ipv6)		[redacted]	[redacted]			5.6 kbps	3.7 kbps	2	
86dd (ipv6)		[redacted]	[redacted]			0 bps	0 bps	0	
86dd (ipv6)		[redacted]	[redacted]			0 bps	0 bps	0	
86dd (ipv6)		[redacted]	[redacted]			0 bps	0 bps	0	

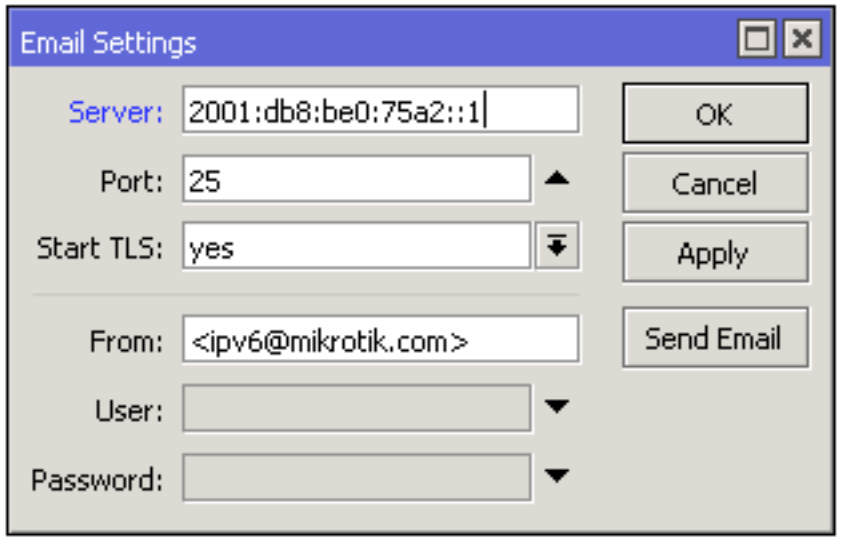
4 items | Total Tx: 5.6 kbps | Total Rx: 3.7 kbps | Total Tx Packet: 2 | Total Rx Packet: 4

Tools → Torch

Traffic Generator

- RouterOS traffic generator supports both IPv4 and IPv6 addresses
- It has several IPv6 specific options, for example:
  - ipv6-next-header
  - ipv6-traffic-class
  - ipv6-flow-label

Email



Server: 2001:db8:be0:75a2::1

Port: 25

Start TLS: yes

From: <ipv6@mikrotik.com>

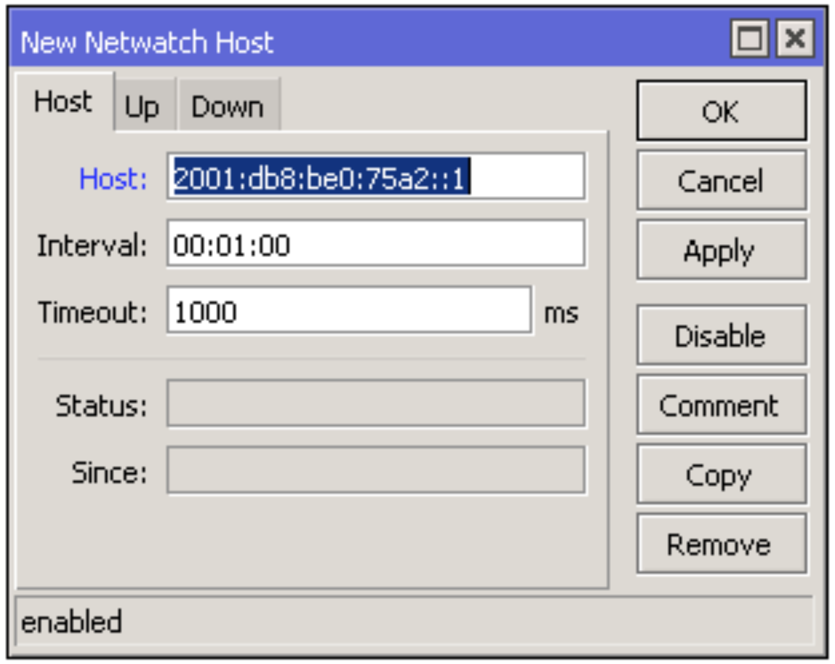
User:

Password:

Tools → Email

- Email tool accepts both IPv4 and IPv6 SMTP address

Netwatch



Host: 2001:db8:be0:75a2::1

Interval: 00:01:00

Timeout: 1000 ms

Status:

Since:

enabled

Tools → Netwatch

- Email tool accepts both IPv4 and IPv6 SMTP address

MikroTik MTCIPv6E

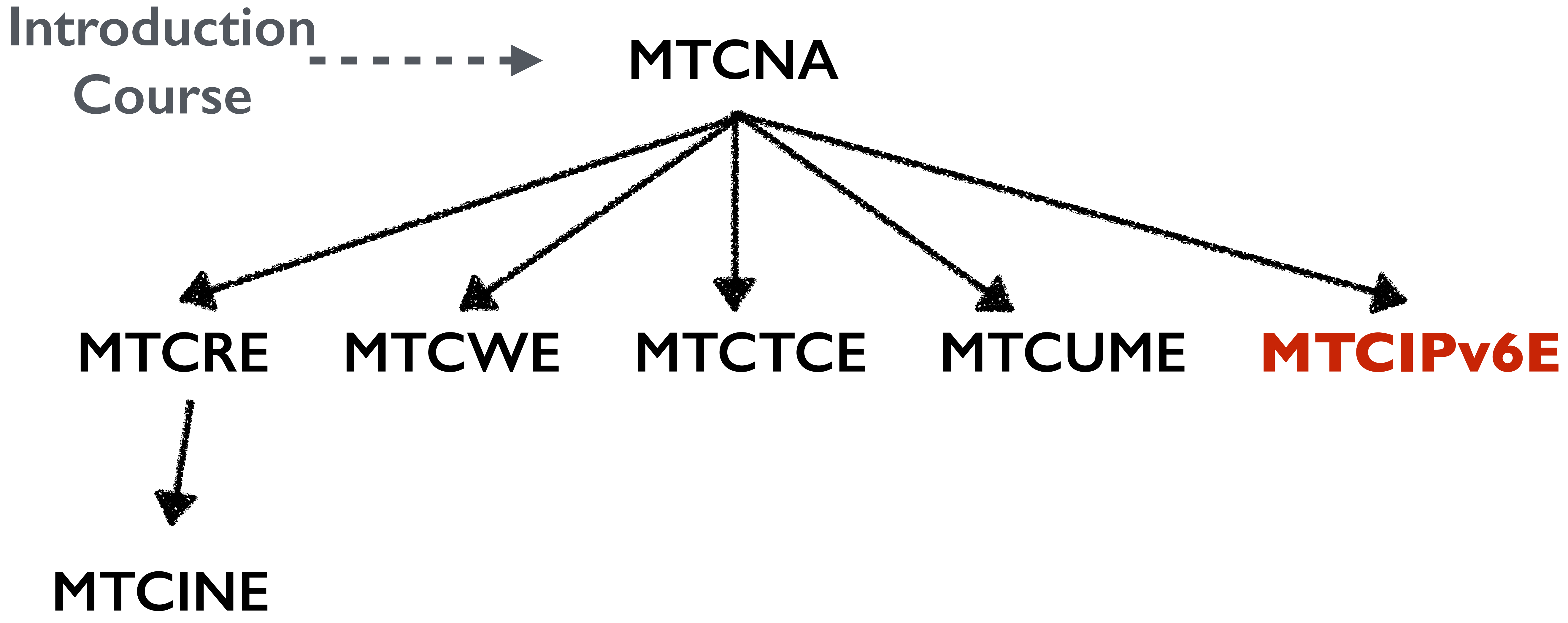
Traffic Flow

- RouterOS traffic flow supports collecting statistics for both IPv4 and IPv6 addresses
- Traffic flow is compatible with Cisco NetFlow
- NetFlow versions 1, 5 and 9 are supported

Module 6 Summary

MTCIPv6E Summary

MikroTik Certified Courses



Introduction Course

MTCNA

MTCRE MTCWE MTCTCE MTCUME MTCIPv6E

MTCINE

For more info see: training.mikrotik.com

Certification Test

- If needed reset router configuration and restore from a backup
- Make sure that you have an access to the www.mikrotik.com training portal
- Login with your account
- Choose **my training sessions**
- Good luck!

MikroTik

Certified IPv6 Engineer (MTCIPv6E)