IPV6 COURSE NOTES JANUAR 2025

CARSTEN STROTMANN, BLUECAT LEARNING TEAM

Version 10.5, 30.01.2025

TABLE OF CONTENT

1. SLIDES AND EXERCISES . . . 1
  1.1. Typographic conventions in this notes . . . 1
  1.2. Extra information: tmux - the terminal multiplexer . . . 1
  1.3. IPv6 Fundamentals . . . 1
  1.4. Exercise 1: IPv6 Addresses . . . 1
    1.4.1. Question 1 . . . 1
    1.4.2. Questions 2 and 3 . . . 2
    1.4.3. Solution: IPv6 Addresses . . . 2
  1.5. Exercise 2: Making connection to the virtual machines . . . 2
  1.6. Exercise 3: IPv6 addresses . . . 2
  1.7. IPv6 Address Configuration . . . 3
  1.8. IPv6 Address Configuration Exercise . . . 3
  1.9. The lab network . . . 3
    1.9.1. Check the IPv6 network configuration on the server and client machines . . . 3
    1.9.2. Link Local communication . . . 4
    1.9.3. Quagga Cheat-Sheet . . . 4
    1.9.4. Configuring an IPv6 Router . . . 4
    1.9.5. Exercise: Inspcting IPv6 traffic . . . 6
    1.9.6. Enable the Privacy Extensions on the client . . . 6
  1.10. DHCPv6 . . . 6
  1.11. DHCPv6 Exercise . . . 7
    1.11.1. DHCPv6 - Relay Agent . . . 7
    1.11.2. The Kea-DHCP-Server . . . 7
    1.11.3. IPv6 DHCP ClientA . . . 9
  1.12. ISC-Kea-DHCPv6 Prefix-Delegation . . . 11
    1.12.1. Prefix Delegation (PD) configuration on a Kea-Server . . . 11
    1.12.2. DHCPv6-PD request from a Client . . . 12
  1.13. ISC-Kea-DHCPv6 Rapid-Commit . . . 13
  1.14. DNS and IPv6 . . . 14
  1.15. DNS and IPv6 exercise . . . 14
    1.15.1. IPv6 in DNS responses . . . 14
    1.15.2. IPv6 support . . . 14
    1.15.3. Send queries towards an IPv6 address of an DNS server . . . 14
    1.15.4. Reverse IPv6 lookup . . . 15
    1.15.5. DNS lookup protocol . . . 15
  1.16. IPv6 NAT . . . 15
  1.17. Happy Eyeballs . . . 15
  1.18. IPv6 Security . . . 15
  1.19. IPv6 Firewall . . . 15
    1.19.1. IPv6 ICMPv6 . . . 15
    1.19.2. References . . . 16
    1.19.3. IPTables ruleset template . . . 16
    1.19.4. nftables host firewall template . . . 17
    1.19.5. IPv6 Firewall exercise . . . 18
    1.19.6. Possible solution . . . 18
  1.20. IPv6 Address Planning . . . 19
  1.21. Routing . . . 19
  1.22. IPv6 Monitoring . . . 19

1.23. IPv4 Sunset 19

CHAPTER 1. SLIDES AND EXERCISES

1.1. TYPOGRAPHIC CONVENTIONS IN THIS NOTES

* `[host]$` indicates a shell of an un-privileged user
* `[host]%` indicates a shell of the root user (the normal bourne shell prompt for `root` is `#`, but that can be confused with comments in configuration files)
* `[host]%` commands for the container host
* `[relay]%` commands for the Router / DHCP relay machine

1.2. EXTRA INFORMATION: TMUX - THE TERMINAL MULTIPLEXER

* We will work with multiple machines (host, router/relay, serverA, serverB, clientA, clientB) in the lab exercises. You can open one terminal session (SSH Session/Browser Tab) per machine, but you can also use the Terminal Multiplexer `tmux` within one terminal session
* start `tmux` by executing the command `tmux`
* attach to a already running tmux session: `[host]% tmux attach`
* Help : CTRL+B - ?
* start a new tmux window : CTRL+B - c
* cancel the current window : CTRL+B - x
* next window : CTRL+B - n
* previous window : CTRL+B - p
* jump to a window : CTRL+B - <n> (0-9)
* Split-Screen (horizontal) : CTRL+B - "
* Split-Screen (vertical) : CTRL+B - %
* switch pane in split screen : CTRL+B - o
* switch pane in split screen : CTRL+B - Cursor-Arrow-Key
* tmux detach session : CTRL+B - d

1.3. IPV6 FUNDAMENTALS

* IPv6 Fundamentals

1.4. EXERCISE 1: IPV6 ADDRESSES

Computers are not required. Answer the following questions.

1.4.1. Question 1

* A router is advertising the prefix `2001:db8:1::/64`
* Write the prefix without leaving out preceding zeros.
  - The answer will look like: `xxxx:xxxx:xxxx:xxxx::/64`

1.4.2. Questions 2 and 3

* A host has the MAC address `08:00:00:00:00:80`
* A router is advertising the prefix `2001:db8:0:a00::/64`

(Write the addresses as compactly as possible. Write the answer into the chat)

* What is the host's link-local address?
* What is the host's global address?

1.4.3. Solution: IPv6 Addresses

* Answer Question 1: `2001:0db8:0001:0000::/64`
* Answer Question 2: `fe80::a00:ff:fe00:80`
* Answer Question 3: `2001:db8::a00:a00:ff:fe00:80`

1.5. EXERCISE 2: MAKING CONNECTION TO THE VIRTUAL MACHINES

* Try to connect to the remote machine via SSH

```
$ ssh user@ipv6-NNN.defaultroutes.org
```

* Become `root` with the `sudo` command
* Inspect the IP address configuration on this virtual machine

```
[host]$ ip a
```

* Try to answer these questions and write the answers to the chat
  - When you send ICMP echo requests to `google.com`, which IP protocol will be used?

```
[host]$ ping google.com
```

* How can you force `ping` to use IPv4 or IPv6 only?

1.6. EXERCISE 3: IPV6 ADDRESSES

* On your virtual machine system (host)
* Inspect the IP address configuration on this virtual machine

```
[host]$ ip a
```

* Try to answer these questions and write the answers to the chat
  - How many IPv6 addresses do you see in the network configuration for `eth0`?
  - Which network "scopes" have these addresses?
  - What is the "valid lifetime" and the "preferred lifetime" of these addresses? Take a guess!
* Start a `tcpdump` capture for IPv6 and write the capture to a pcap file. Keep the capture running until the next exercise (we want to collect IPv6 traffic over approx. one hour)

```
$ sudo tcpdump -i eth0 -w ipv6-one-hour.pcap ip6
```

1.7. IPV6 ADDRESS CONFIGURATION

* IPv6 Address Configuration / The Role of Router in IPv6 / SLAAC

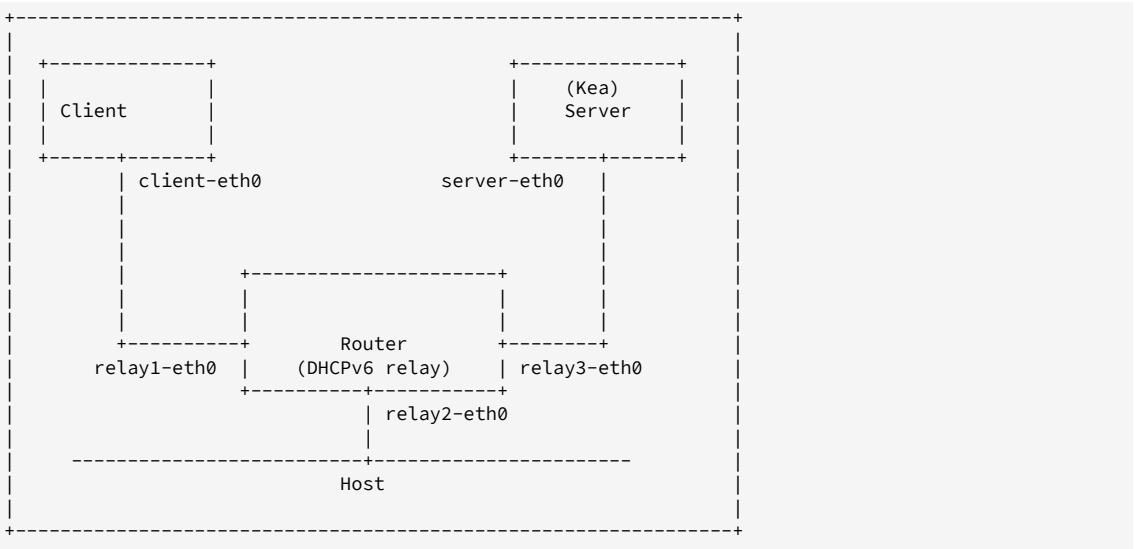
1.8. IPV6 ADDRESS CONFIGURATION EXERCISE

* Login to your virtual machine system
* We use the `lab03` environment. To start the lab containers, execute on the follwing commands on the virtual machine (some error messages when starting the `./run` script are expected and normal)

```
[host]$ sudo -s
[host]% cd /root/lab/lab03
[host]% ./run
```

1.9. THE LAB NETWORK

* The IPv6 lab-network contains 4 machines
  - The host machine (Virtual machine)
  - The router/relay machine (container). The router has network connections to the host, the server and the client.
  - The client machine (container). The client machine is only connected to the network of the router, not to the host or the server. We can enter the client machine through console access
  - The server machine (container). The server machine is only connected to the network of the router, not to the host or the client. We can enter the client machine through console access



1.9.1. Check the IPv6 network configuration on the server and client machines

* In the first shell session (use `tmux` or multiple SSH logins), check the IP configuration. Observe the IPv4 and IPv6 addresses configured on the interfaces

```
[host]$ sudo -s
[host]% /root/bin/enter kea-server
[kea-server]% ip a
```

* In the second shell session (use `tmux` or multiple SSH logins), check the IP configuration of one of the

client containers. Observe the IPv4 and IPv6 addresses configured on the interfaces

```
[host]$ sudo -s
[host]% /root/bin/enter clientA
[clientA]% ip a
```

* In the third shell session (use `tmux` or multiple SSH logins), check the IP configuration of the router/relay container. Observe the IPv4 and IPv6 addresses configured on the interfaces

```
[host]$ sudo -s
[host]% /root/bin/enter relay
[clientA]% ip a
```

1.9.2. Link Local communication

* Note the Link-Local IPv6 addresses of the server and the clientA machines. Write them down.
* Go to the router/relay machine
* Try to send ICMP echo requests from the router/relay machine towards the link-local IPv6 addresses of the clientA and the server container
    - Don't forget to specify the scope-id/zone-id from where the IPv6 traffic should originate (required for all link-local IPv6 communication)

```
[relay]% ping fe80::<address-of-client>%relay1-eth0
PING fe80::<address-of-client>%relay1-eth0(fe80::<address-of-client>%relay1-eth0) 56 data bytes
64 bytes from fe80::<address-of-client>%relay1-eth0: icmp_seq=1 ttl=64 time=0.128 ms
^C
--- fe80::<address-of-client>%relay1-eth0 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.128/0.128/0.128/0.000 ms
```

* Try to ping the link-local IPv6 address of the server machine from the router
* Try to ping the link-local IPv6 address of the server machine from the client
* Which one works, which doesn't? Why? Write your answer to the chat.

1.9.3. Quagga Cheat-Sheet

* Basic commands:
    - `?` at any position show the available commands and arguments to commands
    - `show interface`: current configuration on the interfaces
    - `show running`: dump the running configuration on the screen
    - `write file`: writes current config into `/etc/zebra.conf`
    - `enable`: switch to enable-mode (superuser)
    - `config terminal`: change into configuration mode
    - `show ipv6 forwarding`: prints IPv6 forwarding state
    - `show ipv6 route`: prints the current IPv6 routes

1.9.4. Configuring an IPv6 Router

* So far the machines only have IPv4 and IPv6 link-local addresses. We now want to add routable IPv6 addresses to the network.
    - For this, we will configure a router software on the router machine

- We use *Free Range Routing* (https://frrouting.org), which is an open source routing software solution with a CLI interface similar to Cisco IOS

* IPv6 configuration for the Router/Relay-Container.

```
[host]% /root/bin/enter relay
```

* Start the `zebra` (FRRouting) routing daemon (old project name was `zebra`, hence the name of the command. Then it was called `quagga`, hence the name of the configuration directory. It's a mess.)

```
[relay]% echo "hostname relay" > /etc/quagga/zebra.conf
[relay]% systemctl enable --now zebra
```

* Configure IPv6 on the router using quagga. This will enable the router to send RA for the IPv6 prefix `fd00:100::/64` towards the client and the prefix `2001:db8:100::/64` towards the server container.

```
[relay]% vtysh

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

relay# enable
relay# conf t
relay(config)# interface relay1-eth0
relay(config-if)# ipv6 address fd00:100::1/64
relay(config-if)# ipv6 nd prefix fd00:100::/64 900 300
relay(config-if)# no ipv6 nd suppress-ra
relay(config-if)# no shutdown
relay(config-if)# exit
relay(config)# interface relay3-eth0
relay(config-if)# ipv6 address 2001:db8:100::1/64
relay(config-if)# ipv6 nd prefix 2001:db8:100::/64 900 300
relay(config-if)# no ipv6 nd suppress-ra
relay(config-if)# no shutdown
relay(config-if)# exit
relay(config)# ipv6 forwarding
relay(config)# exit
relay# write
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
[OK]
relay# exit
```

* The network configuration on the router should now look like this

```
[root@relay /]% ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
4: relay1-eth0@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
qlen 1000
    link/ether 1a:83:3f:d9:9d:6a brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.0.2.1/24 scope global relay1-eth0
       valid_lft forever preferred_lft forever
    inet6 fd00:100::1/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::1883:3fff:fed9:9d6a/64 scope link
       valid_lft forever preferred_lft forever
6: relay2-eth0@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
qlen 1000
    link/ether e6:14:15:81:4c:fb brd ff:ff:ff:ff:ff:ff link-netnsid 1
    inet 198.51.100.1/24 scope global relay2-eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::e414:15ff:fe81:4cfb/64 scope link
```

```
       valid_lft forever preferred_lft forever
9: relay3-eth0@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
qlen 1000
    link/ether 9e:8a:98:c2:18:96 brd ff:ff:ff:ff:ff:ff link-netnsid 2
    inet 100.64.0.2/24 scope global relay3-eth0
       valid_lft forever preferred_lft forever
    inet6 2001:db8:100::1/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::9c8a:98ff:fec2:1896/64 scope link
       valid_lft forever preferred_lft forever
```

* Check the IPv6 network interface configuration on the client. The client should now have an unique-local (private) IPv6 address

```
[host]% enter clientA
[clientA]% ip a
```

* Check the IPv6 network interface configuration on the server. The server should now have an global unicast IPv6 address (public IPv6 address)

```
[host]% enter kea-server
[kea-server]% ip a
```

* Check the default routes on both the server and the client container machine

```
[client]% ip -6 route
```

* Can you reach the client from the server or the server from the client? Write your answer into the chat.
* Bonus: Observe the IPv6 traffic between server and client from the router/relay machine

```
[relay]% tcpdump -i relay1-eth0 ip6
```

1.9.5. Exercise: Inspcting IPv6 traffic

* Inspect the traffic dump created yesterday (`ipv6-one-hour.pcap`) with `tshark` (Terminal version of Wireshark)
* Compare the output with the information we've discssed yesterday (IPv6 header, Extension header, Upper-layer payload)

```
# tshark -V -r ipv6-one-hour.pcap
```

1.9.6. Enable the Privacy Extensions on the client

* With this *ad-hoc* configuration of the `net.ipv6.conf.client1-eth0.use_tempaddr` kernel variable we will enable IPv6 privacy addresses. This configuration is not persistent, configure the value into `/etc/sysctl.conf` to have privacy addresses enabled during boot.

```
[clientA]% sysctl -w net.ipv6.conf.client1-eth0.use_tempaddr=2
[clientA]% ip link set dev client1-eth0 down
[clientA]% ip link set dev client1-eth0 up
[clientA]% ip -6 addr show dev client1-eth0
```

1.10. DHCPV6

* DHCPv6

1.11. DHCPV6 EXERCISE

1.11.1. DHCPv6 - Relay Agent

* Enter the router/relay machine
* Enter the router configuration shell and enable the *managed config* (DHCPv6 addresses) and *other config* (DHCPv6 network configuration) in the router advertisements.
  - In our exercise we will start the DHCPv6 client manually, so DHCPv6 would work even without these flags set. But with real world operating systems, the flags must be present to trigger DHCPv6 on the client side.

```
[root@relay /]# vtysh

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

relay# enable
relay# conf t
relay(config)# interface relay1-eth0
relay(config-if)# ipv6 nd managed-config-flag
relay(config-if)# ipv6 nd other-config-flag
relay(config-if)# exit
relay(config)# exit
relay# write
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
[OK]
relay# exit
```

* Start DHCPv6 Relay-Agent `dhcprelay`. Command line parameters
  - `-6` for IPv6
  - `-d` for debug mode (stays in foreground)
  - `-l` for "lower". DHCPv6 are recieved on this network interface, it's the client network side of the router/relay
  - `-u` for "upper". DHCPv6 requests will be send on this interface towards the *All-Server-Multicast* IPv6 address. This is the DHCPv6-Server side of the router/relay. Replace `2001:db8::xxxx` with the full IPv6 Address of the (Kea DHCP) server container.

```
[root@relay /]% dhcrelay -6 -d -l relay1-eth0 \
                  -u 2001:db8:100:xxxx%relay3-eth0
Dropped all unnecessary capabilities.
Internet Systems Consortium DHCP Relay Agent 4.4.2b1
Copyright 2004-2019 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Bound to *:547
Listening on Socket/relay3-eth0
Sending on   Socket/relay3-eth0
Listening on Socket/relay1-eth0
Sending on   Socket/relay1-eth0
Dropped all capabilities.
[...]
```

1.11.2. The Kea-DHCP-Server

* Work in the server container

```
 [host]% /root/bin/enter kea-server
```

* Write the simple DHCPv6 configuration in the file `/etc/kea/kea-dhcp6.conf`. Please make sure to use the IPv6 unicast address of your Kea-DHCPv6 container machine:

```
{
    "Dhcp6": {
        "valid-lifetime": 4000,
        "renew-timer": 1000,
        "rebind-timer": 2000,
        "preferred-lifetime": 3000,
        "interfaces-config": {
            "interfaces": [
                "server-eth0/2001:db8:100::144" # <-- change this
            ]
        },
        "lease-database": {
            "type": "memfile",
            "persist": true,
            "name": "/var/lib/kea/dhcp6.leases"
        },
        "subnet6": [
            {
                "subnet": "fd00:100::/64",
                "pools": [
                    {
                        "pool": "fd00:100::1-fd00:100::ffff"
                    }
                ]
            },
            {
                "subnet": "fd00:200::/64",
                "pools": [
                    {
                        "pool": "fd00:200::1-fd00:200::ffff"
                    }
                ]
            }
        ],
        "loggers": [
            {
                "name": "kea-dhcp6",
                "output_options": [
                    {
                        "output": "/var/log/kea-dhcp6.log"
                    }
                ],
                "severity": "INFO",
                "debuglevel": 0
            }
        ]
    }
}
```

* Test the configuration

```
[kea-server]% kea-dhcp6 -t /etc/kea/kea-dhcp6.conf
2022-06-01 10:21:48.978 INFO  [kea-dhcp6.hosts/169.140437370611840] HOSTS_BACKENDS_REGISTERED the
following host backend types are available: mysql postgresql
2022-06-01 10:21:48.978 INFO  [kea-dhcp6.dhcpsrv/169.140437370611840] DHCPSRV_CFGMGR_USE_UNICAST
listening on unicast address 2001:db8:100:0:58ab:b4ff:fefb:3a68, on interface server-eth0
2022-06-01 10:21:48.978 INFO  [kea-dhcp6.dhcpsrv/169.140437370611840] DHCPSRV_CFGMGR_NEW_SUBNET6 a
new subnet has been added to configuration: fd00:100::/64 with params: t1=1000, t2=2000, preferred-
lifetime=3000, valid-lifetime=4000, rapid-commit is false
2022-06-01 10:21:48.978 INFO  [kea-dhcp6.dhcpsrv/169.140437370611840] DHCPSRV_CFGMGR_NEW_SUBNET6 a
new subnet has been added to configuration: fd00:200::/64 with params: t1=1000, t2=2000, preferred-
lifetime=3000, valid-lifetime=4000, rapid-commit is false
```

* Start the Kea DHCPv6 server daemon using systemd

```
[kea-server]% systemctl start kea-dhcp6
[kea-server]% systemctl status kea-dhcp6
 * kea-dhcp6.service - Kea DHCPv6 Server
   Loaded: loaded (/usr/lib/systemd/system/kea-dhcp6.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2018-12-09 19:53:47 UTC; 4s ago
     Docs: man:kea-dhcp6(8)
 Main PID: 95 (kea-dhcp6)
    Tasks: 1 (limit: 1144)
   Memory: 2.3M
   CGroup: /machine.slice/libpod-
5c5b9d031716ba7b04e2726f7c6f7ef48cdd95d4bbac8c51e7fb591fb7c900c1.scope/system.slice/kea-
dhcp6.service
           \ /usr/sbin/kea-dhcp6 -c /etc/kea/kea-dhcp6.conf

systemd[1]: Started Kea DHCPv6 Server.
kea-dhcp6[95]: 2018-12-09 19:53:47.623 INFO  [kea-dhcp6.dhcp6/95] DHCP6_STARTING Kea DHCPv6 server
version 1.3.0 starting
kea-dhcp6[95]: 2018-12-09 19:53:47.628 INFO  [kea-dhcp6.dhcpsrv/95] DHCPSRV_CFGMGR_ADD_IFACE
listening on interface server-eth0
kea-dhcp6[95]: 2018-12-09 19:53:47.628 INFO  [kea-dhcp6.dhcpsrv/95] DHCPSRV_CFGMGR_NEW_SUBNET4 a new
subnet has been added to co>
kea-dhcp6[95]: 2018-12-09 19:53:47.628 INFO  [kea-dhcp6.dhcp6/95] DHCP6_CONFIG_COMPLETE DHCPv6
server has completed configuratio>
kea-dhcp6[95]: 2018-12-09 19:53:47.629 INFO  [kea-dhcp6.dhcpsrv/95] DHCPSRV_MEMFILE_DB opening
memory file lease database: name=>
kea-dhcp6[95]: 2018-12-09 19:53:47.630 INFO  [kea-dhcp6.dhcpsrv/95] DHCPSRV_MEMFILE_LEASE_FILE_LOAD
loading leases from file /va>
kea-dhcp6[95]: 2018-12-09 19:53:47.632 INFO  [kea-dhcp6.dhcpsrv/95] DHCPSRV_MEMFILE_LFC_SETUP
setting up the Lease File Cleanup >
kea-dhcp6[95]: 2018-12-09 19:53:47.633 INFO  [kea-dhcp6.dhcp6/95] DHCP6_USING_SERVERID server is
using server-id 00:01:00:01:23:>
kea-dhcp6[95]: 2018-12-09 19:53:47.634 INFO  [kea-dhcp6.dhcp6/95] DHCP6_STARTED Kea DHCPv6 server
version 1.3.0 started
```

* Check the DHCPv6 server is listening on the servers unicast address in addition to the link-local (`fe80::/10`) and the Multicast address (`ff02::1:2`):

```
[kea-server]% # lsof -i -n
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
kea-dhcp6 162 root    9u  IPv6  93992      0t0  UDP [2001:db8:100:0:74de:2cff:fe55:144]:dhcpv6-
server
kea-dhcp6 162 root   10u  IPv6  93995      0t0  UDP [fe80::74de:2cff:fe55:144]:dhcpv6-server
kea-dhcp6 162 root   11u  IPv6  93998      0t0  UDP [ff02::1:2]:dhcpv6-server
```

* Check the Kea-DHCPv6 DUID

```
[kea-server]% cat /var/lib/kea/kea-dhcp6-serverid
```

* Monitor the Kea-DHCP6 logfile

```
[kea-server]% tail -f /var/log/kea-dhcp6.log
2018-12-09 20:20:07.033 INFO  [kea-dhcp6.dhcp6/162] DHCP6_STARTED Kea DHCPv6 server version 1.9.4
started
```

1.11.3. IPv6 DHCP ClientA

* work on ClientA

```
[host]% /root/bin/enter clientA
```

* Request for a non-temporary (IA_NA) address (the DHCP client will stay in forground with debug mode `-d`)

```
[clientA]% dhclient -6 -d
```

* Check the output of the command on the client
* Check the output of the DHCPv6 relay

```
Relaying Solicit from fe80::a00:ff:fe00:c0x port 546 going up
Relaying Advertise to fe80::a00:ff:fe00:c0x port 546 down.
Relaying Request from fe80::a00:ff:fe00:c0x port 546 going up
Relaying Reply to fe80::a00:ff:fe00:c0x port 546 down.
```

* Check the log output of the Kea-DHCPv6 server

```
2018-12-09 20:25:20.579 INFO  [kea-dhcp6.leases/162] DHCP6_LEASE_ADVERT
duid=[00:04:39:fe:8c:8b:1e:db:47:fc:a2:1c:23:54:ec:d5:5d:d6], tid=0x331393: lease for address
fd00:100::1 and iaid=2698975002 will be advertised
2018-12-09 20:25:21.594 INFO  [kea-dhcp6.leases/162] DHCP6_LEASE_ALLOC
duid=[00:04:39:fe:8c:8b:1e:db:47:fc:a2:1c:23:54:ec:d5:5d:d6], tid=0xc4e72b: lease for address
fd00:100::1 and
iaid=2698975002 has been allocated
2018-12-09 20:25:22.687 INFO  [kea-dhcp6.leases/162] DHCP6_DECLINE_LEASE Client
duid=[00:04:39:fe:8c:8b:1e:db:47:fc:a2:1c:23:54:ec:d5:5d:d6], tid=0xe04f1d sent DECLINE for address
fd00:100::1 and the server marked it as declined. The lease will be recovered in 86400 seconds.
2018-12-09 20:25:23.334 INFO  [kea-dhcp6.leases/162] DHCP6_LEASE_ADVERT
duid=[00:04:39:fe:8c:8b:1e:db:47:fc:a2:1c:23:54:ec:d5:5d:d6], tid=0x5f88b6: lease for address
fd00:100::2 and iaid=2698975002 will be advertised
2018-12-09 20:25:24.385 INFO  [kea-dhcp6.leases/162] DHCP6_LEASE_ALLOC
duid=[00:04:39:fe:8c:8b:1e:db:47:fc:a2:1c:23:54:ec:d5:5d:d6], tid=0xa61428: lease for address
fd00:100::2 and
iaid=2698975002 has been allocated
```

* Stopp the DHCPv6 client with `CTRL+C`, release the client lease

```
[clientA]% dhclient -6 -r
```

* Compare the lease database on the client with the lease database on the server (`ia-na` is a "non-temporary" address, `ia-ta` would be a "temporary" address)

```
[clientA]% more /var/lib/dhclient/dhclient6.leases
lease6 {
  interface "client1-eth0";
  ia-na a0:df:17:1a {
    starts 1544387573;
    renew 1000;
    rebind 2000;
    iaaddr fd00:100::2 {
      starts 1544387573;
      preferred-life 3000;
      max-life 4000;
    }
  }
  option dhcp6.client-id 0:4:39:fe:8c:8b:1e:db:47:fc:a2:1c:23:54:ec:d5:5d:d6;
  option dhcp6.server-id 0:1:0:1:23:a0:2f:4b:76:de:2c:55:1:44;
}
```

* Server-Lease-File `/var/lib/kea/dhcp6.leases`

```
[kea-server]% less /var/lib/kea/dhcp6.leases
address,duid,valid_lifetime,expire,subnet_id,pref_lifetime,lease_type,iaid,prefix_len,fqdn_fwd,fqdn_
rev,hostname,hwaddr,state
fd00:100::1,00:04:39:fe:8c:8b:1e:db:47:fc:a2:1c:23:54:ec:d5:5d:d6,4000,1544391121,1,3000,0,269897500
2,128,0,0,,,0
fd00:100::1,00,86400,1544473522,1,0,0,2698975002,128,0,0,,,1
fd00:100::2,00:04:39:fe:8c:8b:1e:db:47:fc:a2:1c:23:54:ec:d5:5d:d6,4000,1544391124,1,3000,0,269897500
2,128,0,0,,,0
fd00:100::2,00:04:39:fe:8c:8b:1e:db:47:fc:a2:1c:23:54:ec:d5:5d:d6,0,1544387124,1,0,0,2698975002,128,
0,0,,,0
fd00:100::2,00:04:39:fe:8c:8b:1e:db:47:fc:a2:1c:23:54:ec:d5:5d:d6,4000,1544391573,1,3000,0,269897500
2,128,0,0,,,0
```

* Question: what are there **two** leases in the file (we only have one client)?

1.12. ISC-KEA-DHCPV6 PREFIX-DELEGATION

1.12.1. Prefix Delegation (PD) configuration on a Kea-Server

* Add a prefix delegation subnet to the configuration file `/etc/kea/kea-dhcp6.conf`

```
[...]
        "subnet6": [
            {
                "subnet": "fd00:100::/32",
                "pools": [
                    {
                        "pool": "fd00:100::1-fd00:100::ffff"
                    }
                ],
                "pd-pools": [
                    {
                        "prefix": "fd00:100:10::",
                        "prefix-len": 48,
                        "delegated-len": 56,
                        "excluded-prefix": "fd00:100:10::",
                        "excluded-prefix-len": 64
                    }
                ]
            },
[...]
```

* Complete configuration file with prefix delegation

```
{
    "Dhcp6": {
        "valid-lifetime": 4000,
        "renew-timer": 1000,
        "rebind-timer": 2000,
        "preferred-lifetime": 3000,
        "interfaces-config": {
            "interfaces": [
                "server-eth0/2001:db8:100::144" # <-- change this
            ]
        },
        "lease-database": {
            "type": "memfile",
            "persist": true,
            "name": "/var/lib/kea/dhcp6.leases"
        },
        "subnet6": [
            {
                "subnet": "fd00:100::/64",
                "pools": [
                    {
                        "pool": "fd00:100::1-fd00:100::ffff"
                    }
                ],
                "pd-pools": [
                    {
                        "prefix": "fd00:100:10::",
                        "prefix-len": 48,
                        "delegated-len": 56,
                        "excluded-prefix": "fd00:100:10::",
                        "excluded-prefix-len": 64
                    }
                ]
            },
            {
```

```
                "subnet": "fd00:200::/64",
                "pools": [
                    {
                        "pool": "fd00:200::1-fd00:200::ffff"
                    }
                ]
            }
        ],
        "loggers": [
            {
                "name": "kea-dhcp6",
                "output_options": [
                    {
                        "output": "/var/log/kea-dhcp6.log"
                    }
                ],
                "severity": "INFO",
                "debuglevel": 0
            }
        ]
    }
}
```

* Test the configuration `kea-dhcp6 -t /etc/kea/kea-dhcp6.conf`
* Restart the Kea DHCP Server `systemctl restart kea-dhcp6`

1.12.2. DHCPv6-PD request from a Client

* Request an IPv6 network with a prefix delegation (`-P`) request

```
[clientA]% dhclient -d -6 -P client1-eth0
Internet Systems Consortium DHCP Client 4.3.6
Copyright 2004-2017 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on Socket/client1-eth0
Sending on   Socket/client1-eth0
PRC: Soliciting for leases (INIT).
XMT: Forming Solicit, 0 ms elapsed.
XMT:  X-- IA_PD a0:df:17:1a
XMT:  | X-- Request renew in  +3600
XMT:  | X-- Request rebind in +5400
XMT: Solicit on client1-eth0, interval 1030ms.
RCV: Advertise message on client1-eth0 from fe80::3c1d:4ff:fe2e:950a.
RCV:  X-- IA_PD a0:df:17:1a
RCV:  | X-- starts 1544388880
RCV:  | X-- t1 - renew  +1000
RCV:  | X-- t2 - rebind +2000
RCV:  | X-- [Options]
RCV:  | | X-- IAPREFIX fd00:100:10::/56
RCV:  | | | X-- Preferred lifetime 3000.
RCV:  | | | X-- Max lifetime 4000.
RCV:  X-- Server ID: 00:01:00:01:23:a0:2f:4b:76:de:2c:55:01:44
RCV:  Advertisement recorded.P
RC: Selecting best advertised lease.
PRC: Considering best lease.
PRC:  X-- Initial candidate 00:01:00:01:23:a0:2f:4b:76:de:2c:55:01:44 (s: 10103, p: 0).
XMT: Forming Request, 0 ms elapsed.
XMT:  X-- IA_PD a0:df:17:1a
XMT:  | X-- Requested renew  +3600
XMT:  | X-- Requested rebind +5400
XMT:  | | X-- IAPREFIX fd00:100:10::/56
XMT:  | | | X-- Preferred lifetime +7200
XMT:  | | | X-- Max lifetime +7500
XMT:  V IA_PD appended.
```

```
XMT: Request on client1-eth0, interval 1090ms.
RCV: Reply message on client1-eth0 from fe80::3c1d:4ff:fe2e:950a.
RCV:  X-- IA_PD a0:df:17:1a
RCV:  | X-- starts 1544388881
RCV:  | X-- t1 - renew  +1000
RCV:  | X-- t2 - rebind +2000
RCV:  | X-- [Options]
RCV:  | | X-- IAPREFIX fd00:100:10::/56
RCV:  | | | X-- Preferred lifetime 3000.
RCV:  | | | X-- Max lifetime 4000.
RCV:  X-- Server ID: 00:01:00:01:23:a0:2f:4b:76:de:2c:55:01:44
PRC: Bound to lease 00:01:00:01:23:a0:2f:4b:76:de:2c:55:01:44.
Prefix BOUND6 old= new=fd00:100:10::/56
PRC: Renewal event scheduled in 998 seconds, to run for 1000 seconds.
PRC: Depreference scheduled in 1690 seconds.
PRC: Expiration scheduled in 2690 seconds.
```

1.13. ISC-KEA-DHCPV6 RAPID-COMMIT

Enable "rapid commit" on the Kea-DHCP6 Server

* On the Kea-DHCP6 server configuration file, enable rapid commit on a subnet:

```
[...]
        "subnet6": [
            {
                "subnet": "fd00:100::/64",
                "rapid-commit": true,
                "pools": [
                    {
                        "pool": "fd00:100::1-fd00:100::ffff"
                    }
                ],
[...]
```

* Test the configuration and reload the DHCPv6 service
* Monitor the Kea server logfile and the relay-agent output

Request rapid commit from a Client

* On machine `clientA`, create the file `/etc/dhcp/dhclient6.conf` with the rapid commit option:

```
send dhcp6.rapid-commit;
```

* Remove the old DHCPv6 lease database on the client

```
[clientA]% rm /var/lib/dhclient/dhclient6.leases
```

* Request a DHCPv6 lease with rapid commit

```
[clientA]%  dhclient -6 -d -cf /etc/dhcp/dhclient6.conf
Internet Systems Consortium DHCP Client 4.3.6
Copyright 2004-2017 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on Socket/client1-eth0
Sending on   Socket/client1-eth0
Created duid "\000\0049\376\214\213\036\333G\374\242\034#T\354\325]\326".
PRC: Soliciting for leases (INIT).
XMT: Forming Solicit, 0 ms elapsed.
XMT:  X-- IA_NA a0:df:17:1a
XMT:  | X-- Request renew in  +3600
```

```
XMT:  | X-- Request rebind in +5400
XMT: Solicit on client1-eth0, interval 1080ms.
RCV: Reply message on client1-eth0 from fe80::3c1d:4ff:fe2e:950a.
RCV:  X-- IA_NA a0:df:17:1a
RCV:  | X-- starts 1544389491
RCV:  | X-- t1 - renew  +1000
RCV:  | X-- t2 - rebind +2000
RCV:  | X-- [Options]
RCV:  | | X-- IAADDR fd00:100::2
RCV:  | | | X-- Preferred lifetime 3000.
RCV:  | | | X-- Max lifetime 4000.
RCV:  X-- Server ID: 00:01:00:01:23:a0:2f:4b:76:de:2c:55:01:44
PRC: Bound to lease 00:01:00:01:23:a0:2f:4b:76:de:2c:55:01:44.
PRC: Renewal event scheduled in 999 seconds, to run for 1000 seconds.
PRC: Depreference scheduled in 2999 seconds.
PRC: Expiration scheduled in 3999 seconds.
```

* Observe the different messages on the relay agent and on the Kea DHCP server

1.14. DNS AND IPV6

* DNS and IPv6

1.15. DNS AND IPV6 EXERCISE

1.15.1. IPv6 in DNS responses

* Work on the host VM
* Query the name server entries for the root zone from one root name server

```
[host]$ dig @l.root-servers.net ns .
```

* Observe the number of IPv4 and IPv6 addresses shown in the additional section
* Try the same query against the F-Root-Server `f.root-servers.net`
* Try the same queries (L-Root and F-Root) again, but this time disable the EDNS extension to the DNS protocol that allows large DNS responses with `+noedns`. This will limit the response size over UDP to 512 byte
  - Try the queries over the IPv6 protocol (switch `-6`, should be default) and over the IPv4 protocol (switch `-4`). What are there differences in the DNS answer?

1.15.2. IPv6 support

* Try to query the IPv6 address of popular websites (`google.com`, `twitter.com` (`x.com`), `facebook.com`, `tiktok.com`, `instagram.com`): which websites are IPv6 enabled?

```
[host]% dig AAAA <domain-name>
```

* Can you tell just from the IPv6 address of `instagram.com` to which other service this website belongs?

1.15.3. Send queries towards an IPv6 address of an DNS server

* Find the IPv6 address(es) of `dns.google.com`.
  - Send a DNS query for `ipv6.defaultroutes.org` to one of these addresses. Does the website have an IPv6 address?

```
[host]% dig @<ipv6-address-of-google-public-dns> \
```

```
ipv6.defaultroutes.org AAAA
```

1.15.4. Reverse IPv6 lookup

* Who "owns" the IPv6 address `2001:420:80:1:c:15c0:d06:f00d`?

```
[host]% dig -x 2001:420:80:1:c:15c0:d06:f00d
```

* Is there a hint of the owner in the IPv6 address?

1.15.5. DNS lookup protocol

* Which IP protocol (IPv6 or IPv4) is used when specifying the target of a DNS query by domain name?

```
[host]% dig @ns1.cisco.com. cisco.com.
```

* What is the maximum allowed answer size over UDP for the server `ns1.cisco.com.` (see the `udp` setting in the EDNS OPT pseudosection)? Is this a good configuration?

1.16. IPV6 NAT

* IPv6 and NAT

1.17. HAPPY EYEBALLS

* Happy Eyeballs

1.18. IPV6 SECURITY

* IPv6 Security

1.19. IPV6 FIREWALL

1.19.1. IPv6 ICMPv6

* RFC 4890: "Recommendations for Filtering ICMPv6 Messages in Firewalls" discusses firewall rules for ICMPv6 https://datatracker.ietf.org/doc/html/rfc4890
* Required ICMPv6 message types
  - Destination Unreachable (Type 1) - All codes
  - Packet Too Big (Type 2)
  - Time Exceeded (Type 3) - Code 0 only
  - Parameter Problem (Type 4) - Codes 1 and 2 only
  - Echo request / Echo reply (needed for Teredo)
* Useful ICMPv6 message types
  - Time Exceeded (Type 3) - Code 1
  - Parameter Problem (Type 4) - Code 0
  - Mobile IPv6 (only when in use)
    + Home Agent Address Discovery Request (Type 144)
    + Home Agent Address Discovery Reply (Type 145)

        + Mobile Prefix Solicitation (Type 146)
        + Mobile Prefix Advertisement (Type 147)
* Filter by Firewall policy (permit only if there are good reasons)
    - Node Information Query (Type 139)
    - Node Information Response (Type 140)
    - Router Renumbering (Type 138)
    - ICMPv6 exerimental message types (100, 101, 200, and 201)
    - ICMPv6 extension message types (Types 127 and 255)
* Filter/Block everything else
    - Seamoby Experimental (Type 150)
    - Unallocated Error messages (Types 5-99 inclusive and 102-126 inclusive)
    - Unallocated Informational messages (Types 154-199 inclusive and 202-254 inclusive)

1.19.2. References

* RIPE Labs - Local Packet Filtering with IPv6
* RFC 9099 - Operational Security Considerations for IPv6 Networks
* Recommendations on the Filtering of IPv6 Packets Containing IPv6 Extension Headers at Transit Routers

1.19.3. IPTables ruleset template

```
#!/bin/sh

# default policy
ip6tables -P INPUT DROP
ip6tables -P OUTPUT DROP
ip6tables -P FORWARD DROP

# remove existing ruleset (Firewall Reset)
ip6tables -F
ip6tables -F -t nat

# permit loopback interface
ip6tables -A INPUT  -i lo -j ACCEPT
ip6tables -A OUTPUT -o lo -j ACCEPT

# block source routing (Routing Header 0, Source Routing)
ip6tables -A INPUT -m rt --rt-type 0 -j DROP
ip6tables -A FORWARD -m rt --rt-type 0 -j DROP
ip6tables -A OUTPUT -m rt --rt-type 0 -j DROP

# allow existing incoming connections
ip6tables -A INPUT  -i enp0s3 -m state --state RELATED,ESTABLISHED -j ACCEPT

# allow outgoing connections from the client segment
ip6tables -A INPUT  -i enp0s8 -m state --state RELATED,ESTABLISHED -j ACCEPT

# allow existing / established connections
ip6tables -A OUTPUT  -m state --state RELATED,ESTABLISHED -j ACCEPT
ip6tables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

# Allow connections outgoing from the firewall
ip6tables -A OUTPUT  -m state --state NEW -j ACCEPT

# allow incoming link-local multicast
```

```
ip6tables -A INPUT -s ff02::/16 -j ACCEPT

# allow incoming link-local unicast
ip6tables -A INPUT -s fe80::/10 -j ACCEPT

# ICMPv6
ip6tables -A INPUT -p icmpv6 --icmpv6-type 1 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type 2 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type 3/0 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type 4/1 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type 4/2 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-request -m limit --limit 900/minute -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-reply   -m limit --limit 900/minute -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type neighbour-solicitation -m hl --hl-eq 255 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type neighbour-advertisement -m hl --hl-eq 255 -j ACCEPT

# permit ICMPv6 redirect required for IPv6 router failover
ip6tables -A INPUT -p icmpv6 --icmpv6-type redirect -m hl --hl-eq 255 -j ACCEPT
ip6tables -A INPUT -p icmpv6 -j LOG --log-prefix "ICMPv6:"

# explicit drop
ip6tables -A INPUT -p icmpv6 -j DROP
```

1.19.4. nftables host firewall template

* Example of an combined (IPv4 and IPv6) firewall ruleset for *nftables*

```
# clear ruleset
flush ruleset

# the unspecified IPv6 address (any)
define any = 0::0/0

table inet filter {
        chain input {
                type filter hook input priority 0;

        # set default policy to "drop"
        policy drop;

                # accept any localhost traffic
                iif lo accept

                # accept traffic originated from us
                ct state established,related accept

                # activate the following line to accept common local services
                tcp dport { 22, 80, 443 } ct state new accept

                # NTP multicast
                ip6 daddr ff02::1010 udp dport 123 accept

                # mDNS (avahi)
                ip6 daddr ff02::fb udp dport 5353 accept
                ip  daddr 224.0.0.251 udp dport 5353 accept

                # DHCPv6
                ip6 saddr $any udp dport 546 accept

                # IPP (CUPS)
                tcp dport 631 accept

                # Accept neighbour discovery otherwise IPv6 connectivity breaks.
                ip6 saddr $any icmpv6 type { nd-neighbor-solicit,
                                             nd-router-advert,
                                             nd-neighbor-advert }  accept

                # Accept essential icmpv6
```

```
                ip6 saddr $any icmpv6 type { echo-reply,
                                             echo-request,
                                             packet-too-big,
                                             destination-unreachable,
                                             time-exceeded,
                                             parameter-problem } accept

                # count and drop any other traffic
                counter log prefix "nftables drop: " drop
        }
}
```

1.19.5. IPv6 Firewall exercise

* Work on the VM host machines
* Make sure to be the user `root`
* List the current nftables ruleset

```
[host]% nft list ruleset
```

* Stop the Firewall-Daemon

```
[host]% systemctl stop firewalld
```

* Create a nftables firewall ruleset using the template above. Write the ruleset into `/etc/nftables.rules`
  - Allow https (Port 443 TCP)
  - Allow ssh (Port 22 TCP)
  - Allow DNS (Port 53 UDP/TCP)
  - Allow required ICMPv6
  - Allow established connections
  - Block everything else
* Check the ruleset with `nft -c -f /etc/nftables.rules`
* If no errors are reported, load the ruleset (temporaily) with

```
[host]% nft -f /etc/nftables.rules
```

* List the new ruleset with `nft list ruleset`
* You can inspect the firewall log messages with

```
[host]% journalctl -fk
```

* Does the Web-IU (or SSH) still work? If not, let the trainer know that the machine needs a reset

1.19.6. Possible solution

```
# clear ruleset
flush ruleset

# the unspecified IPv6 address (any)
define any = 0::0/0

table inet filter {
        chain input {
                type filter hook input priority 0;
```

```
                # set default policy to "drop"
                policy drop;

                # accept any localhost traffic
                iif lo accept

                # Allow established communication
                ct state established,related accept

                # DNS, HTTPS, SSH allowed
                tcp dport { 22, 53, 443 } ct state new accept
                udp dport { 53 } ct state new accept

                # Accept neighbour discovery otherwise IPv6 connectivity breaks.
                ip6 saddr $any icmpv6 type { nd-neighbor-solicit,
                                             nd-router-advert,
                                             nd-neighbor-advert }  accept

                # Accept essential icmpv6
                ip6 saddr $any icmpv6 type { echo-reply,
                                             echo-request,
                                             packet-too-big,
                                             destination-unreachable,
                                             time-exceeded,
                                             parameter-problem } accept

                # count and drop any other traffic
                counter log prefix "nftables drop: " drop
        }
}
```

1.20. IPV6 ADDRESS PLANNING

* IPv6 Address Planning

1.21. ROUTING

* Routing / OSPF

1.22. IPV6 MONITORING

* IPv6 Monitoring

1.23. IPV4 SUNSET

* IPv4 Sunset