More Addresses, Routing and Control

Unit 16 - Hands-On Networking - 2018

Prof. Dr.-Ing. Thorsten Herfet, Andreas Schmidt, Pablo Gil Pereira

Telecommunications Lab, Saarland Informatics Campus, 26th Feb. 2018

Recap

- **Socket Programming**
    - Sockets
    - Server Types (Iterative, Forked, Threaded)
- **Time synchronization**
    - NTP
    - PTP
- **DHCP**
    - Stateful configuration of IP parameters (Addresses, Prefix, DNS suffix, ...)

⚠ IPv4 address space is fully exhausted since 2015

IPv6

Internet Protocol (Version 6)

- **Successor to IPv4**
- Defined in RFC2460 (in 1998)
- **128 bit** addresses
- Referred to as **IPv6**

Even though it works similar to and shares many design principles with IPv4, IPv6 is **not an extension** to IPv4, but a **new protocol**.

New in IPv6

- **Expanded Addressing Capabilities**

  IPv6 increases the IP address size from 32 bits to 128 bits, to support more levels of addressing hierarchy, a much greater number of addressable nodes, and simpler auto-configuration of addresses.

- **Header Format Simplification**

  Some IPv4 header fields have been dropped or made optional, to reduce the common-case processing cost of packet handling and to limit the bandwidth cost of the IPv6 header.

- **Improved Support for Extensions and Options**

  More efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future.

- **Authentication and Privacy**

IPv6 vs. IPv4

- No Broadcasts
  IPv6 tries to avoid broadcasts wherever possible.
- No ARP
  Link layer address resolution is not handled by ARP anymore, but is now part of ICMPv6.
- No Checksums
  IPv6 headers do not include a checksum anymore.
- Routers don't fragment
  IPv6 routers will not fragment packets if the packet size is bigger than the MTU. A router will notify the sending host about the smaller MTU and drop the packet.
- Built-in Autoconfiguration
  IPv6 nodes can be fully configured using features of the IPv6 protocol suite instead of manual configuration or special services such as DHCP.

IPv6 | Header Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |  Next Header  |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Source Address                        |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      Destination Address                      |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv6 | Header Fields

Version (4 bit):
IP protocol version (It's 6).

Traffic Class (8 bit):
Like the ToS field in IPv4. Can be used by nodes and routers to distinguish between different classes or priorities of packets.

Flow Label (20 bit):
Can be used to label a sequence of packets as belonging to the same flow in order to help routers process packets faster to meet e.g. real-time requirements.

Payload Length (16 bit):
Amount of data carried after the IP header in bytes.

Next Header (8 bit):
Like   in IPv4. Designates the type of data following the IP header.

Hop Limit (8 bit):
Same as   in IPv4.

IPv6 | Extensions Headers

Extension headers are daisy chained to the main IPv6 header.

```
+---------------+------------------------
|  IPv6 header  | TCP header + data
|               |
| Next Header = |
|      TCP      |
+---------------+------------------------


+---------------+----------------+------------------------
|  IPv6 header  | Routing header | TCP header + data
|               |                |
| Next Header = |  Next Header = |
|    Routing    |      TCP       |
+---------------+----------------+------------------------


+---------------+----------------+-----------------+-----------------
|  IPv6 header  | Routing header | Fragment header | fragment of TCP
|               |                |                 |  header + data
| Next Header = |  Next Header = |  Next Header =  |
|    Routing    |    Fragment    |       TCP       |
+---------------+----------------+-----------------+-----------------
```

IPv6 | Address Format

- Defined in RFC4291
- **8 blocks of 16 bits each** represented in **hexadecimal** notation

2a00:1450:0000:0815:0000:0000:0000:2003

Leading zeros in each block may be ommitted

2a00:1450:0:815:0:0:0:2003

Exactly one group of **consecutive zero blocks** may be compressed using **"::"**.

2a00:1450:0:815::2003

It be the **longest one**. Single zero blocks be compressed.

HON

It's all about the prefixes ...

Subnets

- IPv6 also uses the **longest matching prefix** pattern
- Notation resembles IPv4 CIDR

2a01:4f8:a0:54c8::/64
2a00:1450:4000::/37
::1/128

Commonly ISPs allocate a prefix of **/48** or **/64** per customer (i.e. **end user**).

It's all about the prefixes ...

IPv6 address types are identified by their **prefix**

Allocation	Binary Prefix	Hex/IPv6 Prefix
Reserved	0000 0000	::/8
Aggregatable Global Unicast	001	2000::/3
Unique Local (RFC4193)	1111 110	fc00::/7
Link-Local Unicast	1111 1110 10	fe80::/10
Multicast	1111 1111	ff00::/8

Quiz

What is the lowest address in the subnet 2a01:4f8:a0:54c8::/64?

A: 2a01:4f8:a0:54c8::	**B:** 2a01:4f8:a0::
C: 2a01:4f8::	**D:** 2a01::a0:54c8::

Answer:

- ✔ A: The prefix padded with zeros.
- ✖ B: This address has a different prefix.
- ✖ C: This one also has a different prefix.
- ✖ D: This isn't even a valid address...

Link-Local Unicast Addresses

- **Generated** from the **interface MAC address**
- **Every IPv6-capable interface** has one of these
- Used for IPv6 **Neighbor Discovery Protocol** and **DHCPv6**
- **Must not be forwarded** by routers

```
              |    10    |                        |                           |
              |   bits   |         54 bits        |          64 bits          |
              +----------+------------------------+---------------------------+
    fe80::/64 |1111111010|           0            |       interface ID        |
              +----------+------------------------+---------------------------+

              +-----------------------------------+
interface ID =|     MAC1'    | 0xFFFE |    MAC2    |
              +-----------------------------------+

       MAC1' = ((MAC Address & 0xffffff000000) >> 24) | 0x020000
       MAC2  = (MAC Address & 0x000000ffffff)
```

70:54:d2:7b:7a:db → fe80::7254:d2ff:fe7b:7adb

Unique Local Addresses (RFC4193)

- Similar to RFC1918-Addresses
- ISP independent
- Can be used within a site without having a connection to the public Internet
- Allow sites to be privately interconnected or combined without address conflicts
- **Must not be forwarded** to the public Internet
- **Designed** to be **hard to aggregate**

```
| 7 bits   |1|  40 bits   |  16 bits  |          64 bits           |
+----------+-+------------+-----------+----------------------------+
| 1111 110 |1| Global ID  | Subnet ID |        Interface ID        |
+----------+-+------------+-----------+----------------------------+
```

```
   Global ID:   Pseudo-Random ID to ensure disjunct subnets.

   Subnet ID:   Subnet ID to identify a subnet within a site.

Interface ID:   See previous slide
```

Quiz

What is the *highest* address in the subnet 2a01:4f8:a0:54c8::/64?

✔ The **"all-ones address"**: 2a01:4f8:a0:54c8**:ffff:ffff:ffff:ffff**

How many addresses are in the subnet 2a01:4f8:a0:54c8::/64?

✔ 2^{64}

How many *usable* addresses are in the subnet 2a01:4f8:a0:54c8::/64?

✔ 2^{64}

IPv6 has **no reserved addresses per subnet**.
Instead of broadcasts IPv6 uses **multicast**.

IPv6 | Special Use Addresses

Unicast

	Scope	Address
Unspecified Address		::
Loopback Address	host	::1

Multicast (,)

	Scope	Address
All-Nodes Address ("Broadcast")	link	ff02::1
All-Routers Address	link	ff02::2
All DHCP Agents	link	ff02::1:2
All-Routers Address	site	ff05::2

This is just an excerpt. In practice there are many more.

IPv6 | Required Addresses (↗)

Every IPv6 capable node is required to recognize the following addresses as **identifying itself**:

- **Loopback** Address
- **Link-Local** Address(es)
- **All-Nodes Multicast** Address
- **Multicast** Addresses for **all groups the node belongs to** (e.g. DHCPv6 servers)

IPv6 routers additionally have to recognize the following addresses

- The **All-Routers multicast** addresses
- The **Subnet-Router Anycast** addresses for all interfaces for which it is **configured to act as a router**

ICMPv6

ICMPv6

- Defined in RFC4443
- IP protocol number: **58**
- Used to send error / informational messages

General Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                         Message Body                          +
|                                                               |
```

ICMPv6 | Error Messages

Type	Code	Description
1		Destination Unreachable
	0	
	1	
	3	
	4	
2	0	Packet Too Big
3		Time Exceeded
	0	
	1	
4		Parameter Problem
	0	
	1	

ICMPv6 | Informational Messages

Type	Description
128	Echo Request
129	Echo Reply
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect Message

Neighbor Discovery (ND)

ND (RFC4861)

- Runs on **ICMPv6**
- Uses **link-local unicast addresses** and **special use multicast addresses**
- Translates **IPv6** addresses to **link-layer** addresses
- **Router Discovery**
- Supports **address auto-configuration** and **prefix discovery**
- Corresponds to a combination of ARP, Router Discovery and ICMP Redirects from IPv4

Neighbor Solicitation

- **Analogous** to **ARP request** in IPv4
- **Less protocol overhead** than the corresponding ARP message

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     0x87      |      0x0      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Reserved                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                       Target Address                          +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options ...
+-+-+-+-+-+-+-+-+-+-+-+-
```

Neighbor Advertisement

Analogous to **ARP reply** in IPv4

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     0x88      |      0x0      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R|S|O|                     Reserved                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                       Target Address                          +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options ...
+-+-+-+-+-+-+-+-+-+-+-+-
```

```
R:  Router bit. 1 if the sending station is a router, 0 else.
S:  Solicited Flag. 1 if the message is a response to a solicitation, 0 else.
O:  Override Flag. 1 if the advertisement should override a cached entry, 0 else.
```

Auto-Configuration

DHCPv6 (RFC3315)

Stateful auto-configuration

- DHCPv6 server **keeps track of addresses**
- DHCPv6 server can issue **specific addresses** to **specific clients**
- DHCPv6 server can issue **additional parameters** (e.g. DNS servers)

SLAAC (RFC4862)

Stateless auto-configuration

- No additional servers required
- Automatic generation of link-local address(es)
- Automatic generation of global unicast address(es) with prefix received via **Router Advertisements**

Stateless Address Auto Configuration

- **Routers** periodically send out **Router Advertisements** to the **All-Nodes Multicast Address**
- Router Advertisements contain
  - information about **all prefixes on the link**
  - a **maximum lifetime** for each prefix
- To **obtain advertisements quickly** nodes can also **solicit for advertisements**
- The node **combines the prefix** with its **interface identifier** (see Link-Local Addresses)

```
|           128 - N bits               |        N bits         |
+--------------------------------------+-----------------------+
|           link prefix                | interface identifier  |
+--------------------------------------------------------------+
```

In **IEEE802 networks** SLAAC can only be used with a prefix length of **exactly 64 bit**

Router Advertisement

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     0x86      |     0x0       |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Cur Hop Limit |M|O|  Reserved |       Router Lifetime         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Reachable Time                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Retrans Time                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options ...
+-+-+-+-+-+-+-+-+-+-+-+-
```

```
 Hop Limit: Default hop limits for IPv6 packets on this link.
         M: Managed Address bit. 1 if DHCP is in use, 0 else.
         O: Other Configuration bit. 1 if other config information
            (e.g. DNS options) is available via DHCP, 0 else.
```

```
Router Lifetime:
 Reachable Time:    Options for NDP. We will not discuss them here.
   Retrans Time:
```

Router Advertisement | Options

Prefix Information Option

Specifies on-link prefix which can be used for SLAAC.
If there is more than one prefix, this option may be included once for each prefix.

MTU Option

Specifies the MTU that should be used on the current link.

IPv6 **requires** underlying link-layer protocols to support a **minimum MTU of 1280 bytes** or to **offer fragmentation**.

Privacy vs IPv6

Privacy Extensions

- SLAAC generated addresses contain the interface ID
- The interface ID can be easily distinguished from the prefix
- Interface ID can be used to track per-host activity

Idea: ***Randomize the interface ID***

Prefix Randomization

- Most internet users are home users
- The prefix itself already gives away who is transmitting

Idea: ***Randomize the prefix***

Transition

IPv6 and IPv4

- **Very hard** to get rid of IPv4
  There are still sites / ISPs **not offering IPv6** to their users (Saarland Informatics Campus is one of them).

- **Users don't care** much for IPv6 and many **admins don't understand it** or **don't bother**.

- **IPv6 and IPv4** network stacks currently **coexist**
  This is going on for ~20 years now and probably will be going on a lot longer.

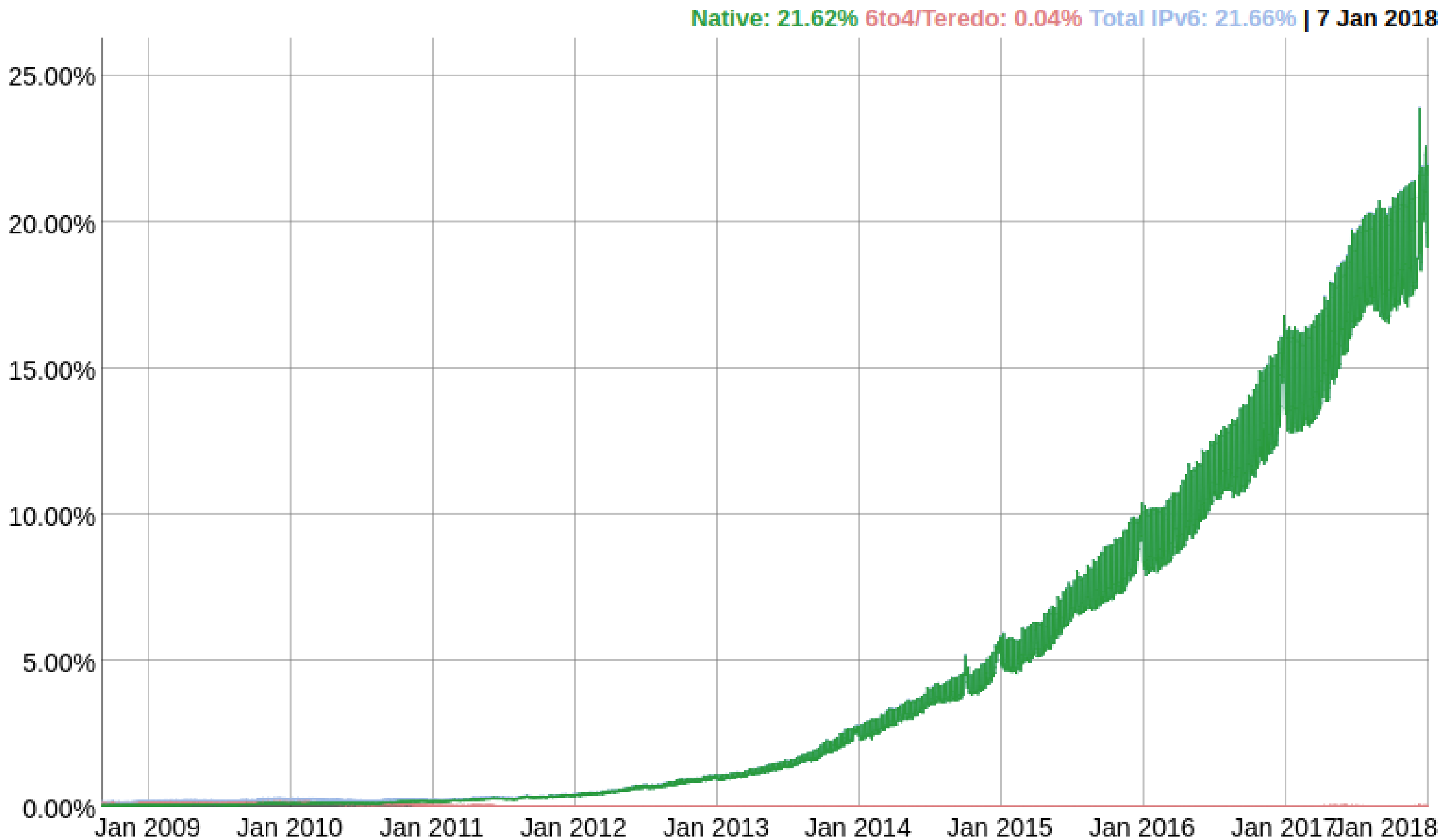
- Internet **core router support is spotty**
  Not all routers making up the Internet are supporting IPv6 just yet.

- **Many companies don't offer their services via IPv6**

- Several **transition techniques** exist to deal with spotty IPv6 connectivity and help introduce IPv6.

IPv6 | Adoption

2018 - IPv6 Adoption at ~22%



Dual-Stack (aka the best-case scenario)

- **Full IPv6/IPv4 connectivity** for every device
- **No tunnels / NAT needed**

Dual-Stack Lite (aka the next best thing)

- **No public IPv4 address**
- Internal network uses private IPv4 and public IPv6 Addresses
- Connection to Internet purely IPv6

👎 Requires NAT for IPv4
👎 NAT deep inside the ISPs network

IPv6 Transition Techniques

- **6in4** (RFC4213)
  Encapsulate IPv6 datagrams in IPv4 datagrams.

- **6to4** (RFC3056)
  Bidirectional address mapping which uses 6in4 for datagram encapsulation.

- **Teredo** (RFC4380)
  IPv6 tunnels for devices behind a IPv4 NAT gateway.

- **NAT64** (RFC6146)
  IPv6 to IPv4 NAT at the ISP level.

- and many more ...

Tools

- iproute2 (Linux)
- ping6 (Linux, macOS), ping (Windows)
- nmap (Linux, macOS, Windows)

```
# Current IP address configuration

$ ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    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
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 70:54:d2:7b:7a:db brd ff:ff:ff:ff:ff:ff
    inet 134.96.86.110/25 brd 134.96.86.127 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::7254:d2ff:fe7b:7adb/64 scope link
       valid_lft forever preferred_lft forever
```

Tools

```
# Show IPv6 neighbors

$ ip -6 neighbor show
fe80::ea40:f2ff:fe3e:952c dev eth0 lladdr e8:40:f2:3e:95:2c REACHABLE
fe80::468a:5bff:fe27:880b dev eth0 lladdr 44:8a:5b:27:88:0b STALE
fe80::468a:5bff:fe97:aa9d dev eth0 lladdr 44:8a:5b:97:aa:9d STALE
```

```
# Echo requests

$ ping6 -c4 -I eth0 fe80::468a:5bff:fe97:aa9d
PING fe80::468a:5bff:fe97:aa9d(fe80::468a:5bff:fe97:aa9d) from fe80::7254:d2ff:fe7b:7adb eth0: 56 data bytes
64 bytes from fe80::468a:5bff:fe97:aa9d: icmp_seq=1 ttl=64 time=0.261 ms
64 bytes from fe80::468a:5bff:fe97:aa9d: icmp_seq=2 ttl=64 time=0.195 ms
64 bytes from fe80::468a:5bff:fe97:aa9d: icmp_seq=3 ttl=64 time=0.258 ms
64 bytes from fe80::468a:5bff:fe97:aa9d: icmp_seq=4 ttl=64 time=0.257 ms

--- fe80::468a:5bff:fe97:aa9d ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.195/0.242/0.261/0.033 ms
```

Note: When using link-local addresses you have to specify the link / interface to be used.

Wrap-Up

Questions?

Take-Home Messages

- IPv6 is a long-defined successor to IPv4 with many design improvements:
    - More addresses (to cope with increasing number of Internet hosts).
    - More clever addressing scheme with useful default addresses.
    - Less fragmentation and checksums.
    - Avoiding broadcast as much as possible.
- ICMPv6 replaces the functionality of ARP (with NDP).
- Transition from IPv4 to IPv6 is still slow, even though it has been published in the nineties.

Further Reading

- The mentioned RFCs
- Hagen, Silvia: "IPv6 Essentials" (O'Reilly)