IPv6 Address Configuration

SLAAC and the role of Router in IPv6 networks

Carsten Strotmann

CREATED: 2025-01-27 MON 20:12

Agenda

- The role of router in IPv6 networks
- SLAAC Addressing Problems
- IPv6 Privacy Extensions
- Stable Privacy-Enhanced Addresses

The role of router in IPv6 networks

IP-Addresses for hosts in an IPv6 world

- In IPv6 networks, it is perfectly fine ...
  - For one interface to have multiple IPv6 addresses (possibly from different prefixes)
  - One network segment to have multiple routers
    - Redundant for the same network prefix or
    - Routers announcing different prefixes
  - One host to have more than one gateway addresses (default route)

The role of router

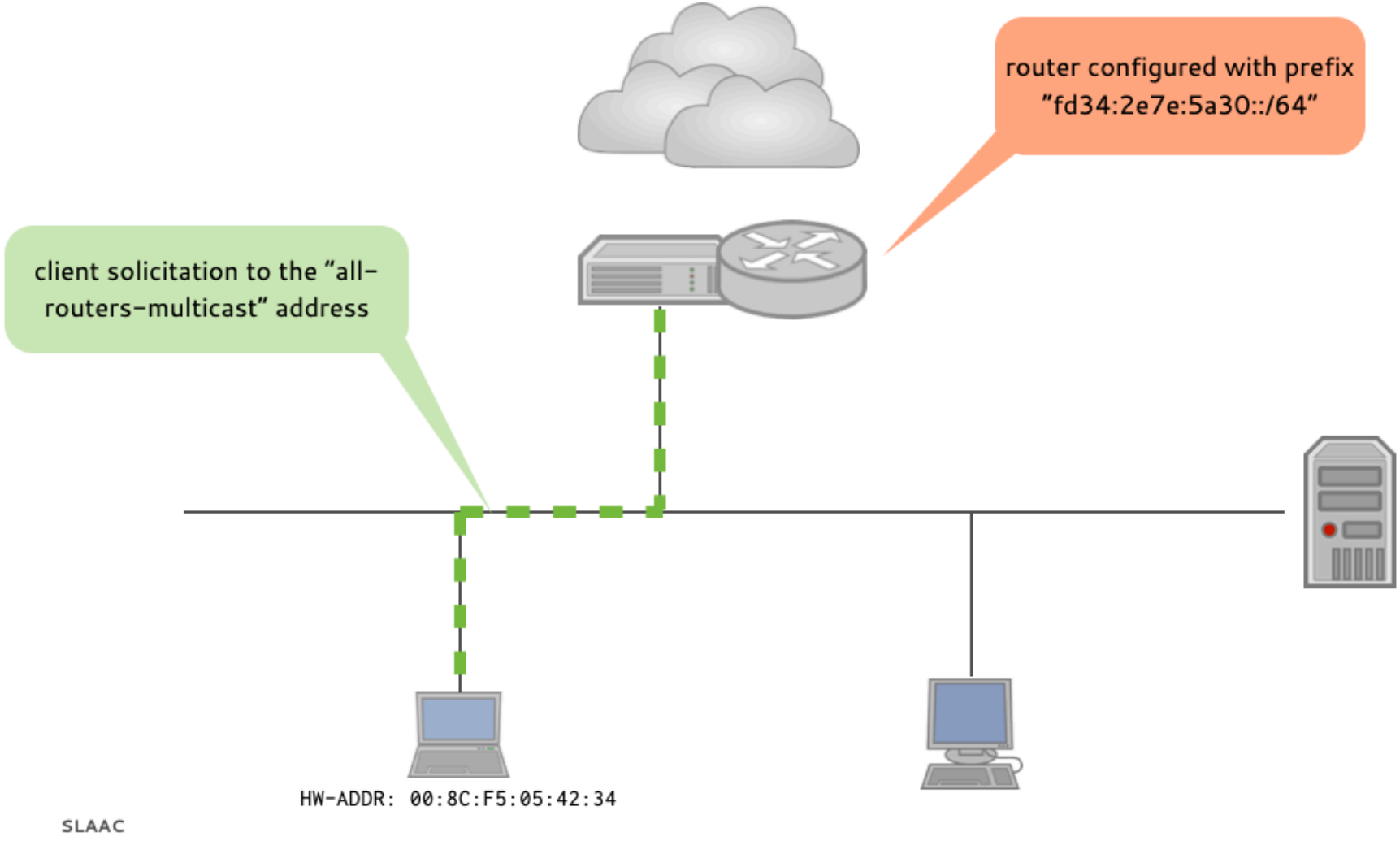
- Routers play a central role in managed IPv6 networks
- The routers advertise:
  - Available network prefixes and their lifetimes
  - Gateway addresses
  - Address configuration policy (SLAAC, DHCPv6)

- IPv6 implements “fate sharing” for gateway addresses (↪RFC 5505 - Principles of Internet Host Configuration)

Router advertisements (RA)

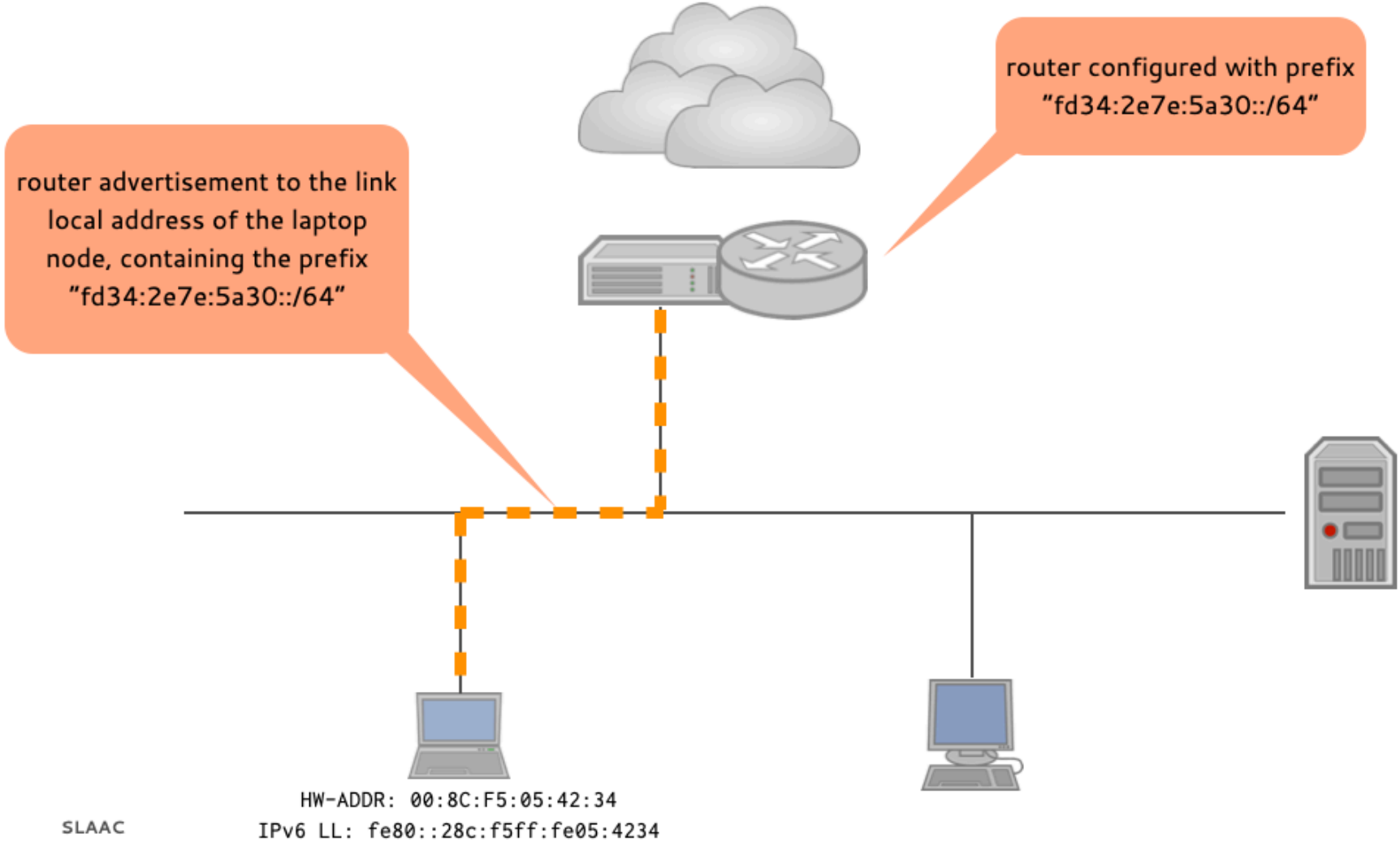
- An IPv6 node sends a *router solicitation* to the *all-routers* address: `ff02::2`.
- Routers reply with “router advertisement” messages, unicasted to the node's LLA (link-local `fe80::/10` address).
- Router send *router advertisement* (RA) messages periodically to the *all-nodes* address: `ff02::1`.

IPv6 Address Configuration

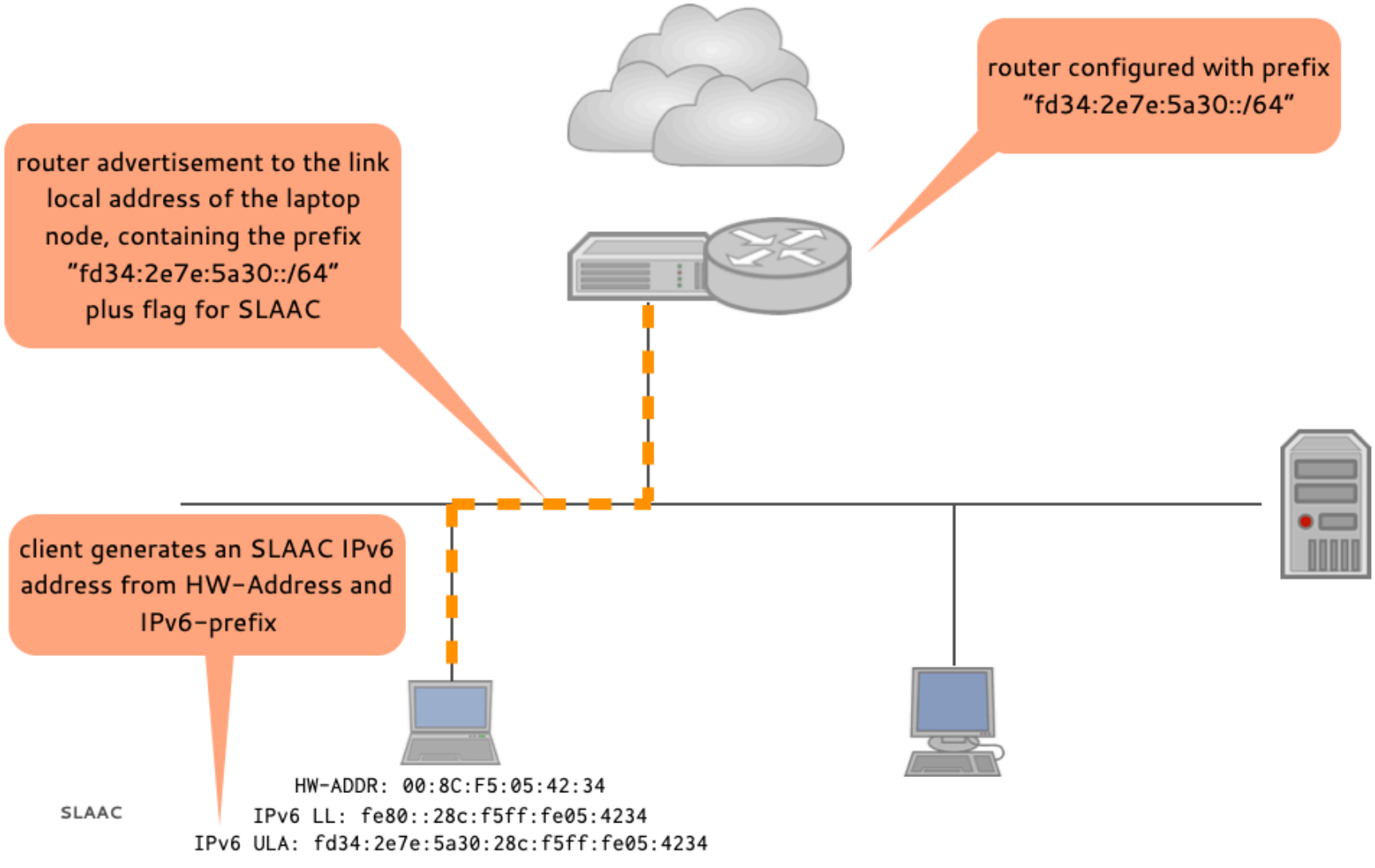


SLAAC

IPv6 Address Configuration



SLAAC


router configured with prefix
"fd34:2e7e:5a30::/64"
router advertisement to the link local address of the laptop node, containing the prefix "fd34:2e7e:5a30::/64" plus flag for SLAAC
client generates an SLAAC IPv6 address from HW-Address and IPv6-prefix
SLAAC
HW-ADDR: 00:8C:F5:05:42:34
IPv6 LL: fe80::28c:f5ff:fe05:4234
IPv6 ULA: fd34:2e7e:5a30:28c:f5ff:fe05:4234

Router Advertisement

DHCPv6 for IPv6 addresses

Lifetime of this router

MTU for all network nodes

DHCPv6 for network configuration

Hop limit for new IP packets

preference of this router (high/medium/low)

advertise routers link-layer address (used as the gateway address by hosts)

```
interface eth0
{
    AdvSendAdvert on;
    AdvManagedFlag off;
    AdvOtherConfigFlag off;
    AdvCurHopLimit 64;
    AdvDefaultLifetime 1800;
    AdvDefaultPreference medium;
    AdvSourceLLAddress on;
    AdvLinkMTU 1280;

    prefix fd34:2e7e:5a30::/64
    {
        AdvValidLifetime 2592000;
        AdvPreferredLifetime 604800;
        AdvOnLink on;
        AdvAutonomous on;
    }; # End of prefix definition


    prefix 2001:db8:100::/64
    {
        AdvValidLifetime 2592000;
        AdvPreferredLifetime 604800;
        AdvOnLink on;
        AdvAutonomous on;
    }; # End of prefix definition


    RDNSS 2a01:198:2b6:0:20b:2fff:fee1:ac2f
    {
        AdvRDNSSLifetime 900;
    }; # End of RDNSS definition


    DNSSL home.strotmann.de
    {
        AdvDNSSLLifetime 900;
    }; # End of DNSSL definition
}
```

“radump” output

Router Advertisement

"radump" output

```
interface eth0
{
    AdvSendAdvert on;
    AdvManagedFlag off;
    AdvOtherConfigFlag off;
    AdvCurHopLimit 64;
    AdvDefaultLifetime 1800;
    AdvDefaultPreference medium;
    AdvSourceLLAddress on;
    AdvLinkMTU 1280;

    prefix fd34:2e7e:5a30::/64
    {
        AdvValidLifetime 2592000;
        AdvPreferredLifetime 604800;
        AdvOnLink on;
        AdvAutonomous on;
    }; # End of prefix definition


    prefix 2001:db8:100::/64
    {
        AdvValidLifetime 2592000;
        AdvPreferredLifetime 604800;
        AdvOnLink on;
        AdvAutonomous on;
    }; # End of prefix definition


    RDNSS 2a01:198:2b6:0:20b:2fff:fee1:ac2f
    {
        AdvRDNSSLifetime 900;
    }; # End of RDNSS definition


    DNSSL example.com.
    {
        AdvDNSSLLifetime 900;
    }; # End of DNSSL definition
}
```

A ULA prefix

valid lifetime: nodes can use this prefix for connections for this amount of time

preferred lifetime: nodes can use this prefix for **new** connections for this time

Prefix is on this network link

Nodes should use SLAAC to generate an IPv6 address from this prefix

GUA (Global Unicast Address) prefix for the same link

IPv6 address of a DNS resolver

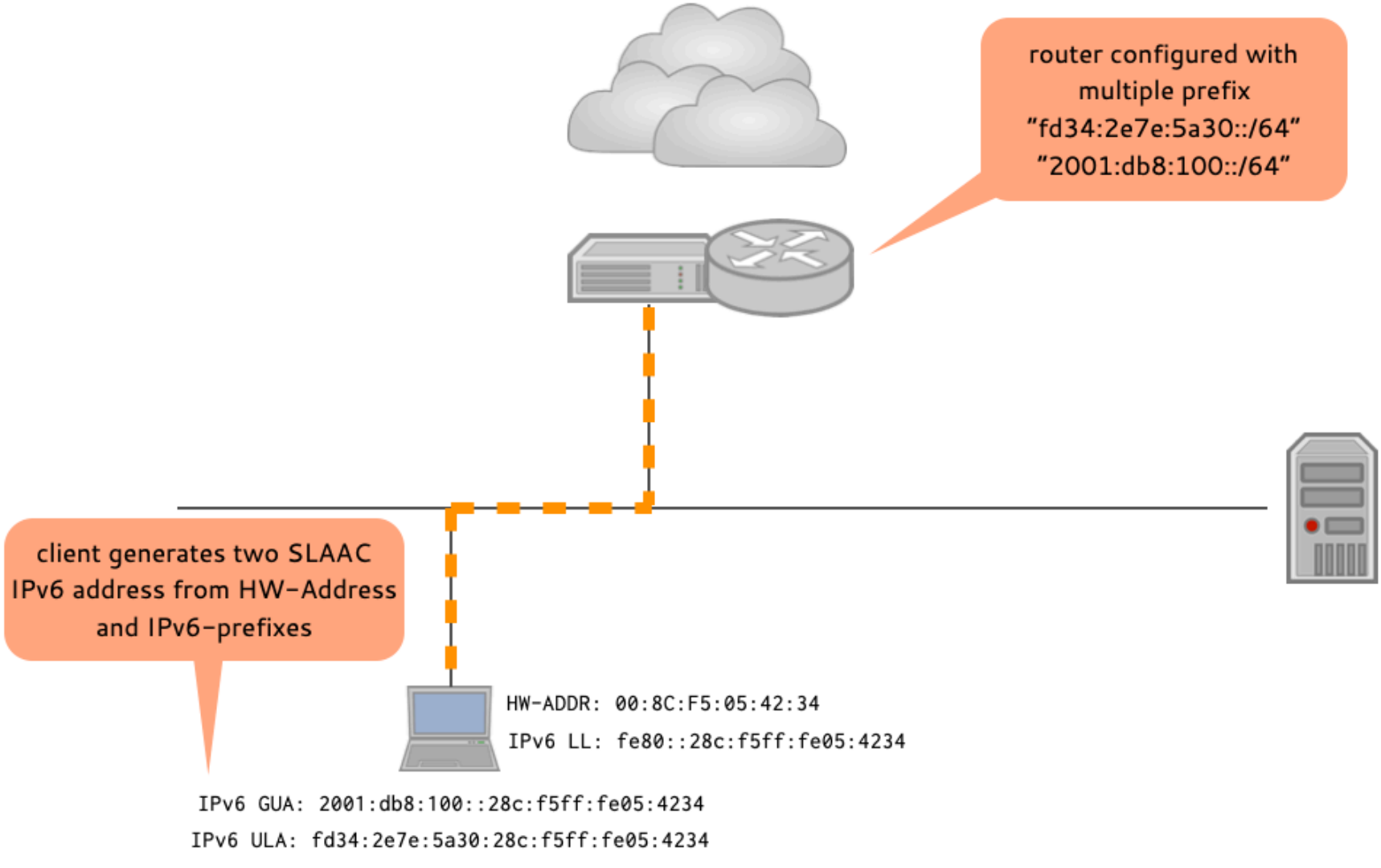
DNS search list (and local domain name)

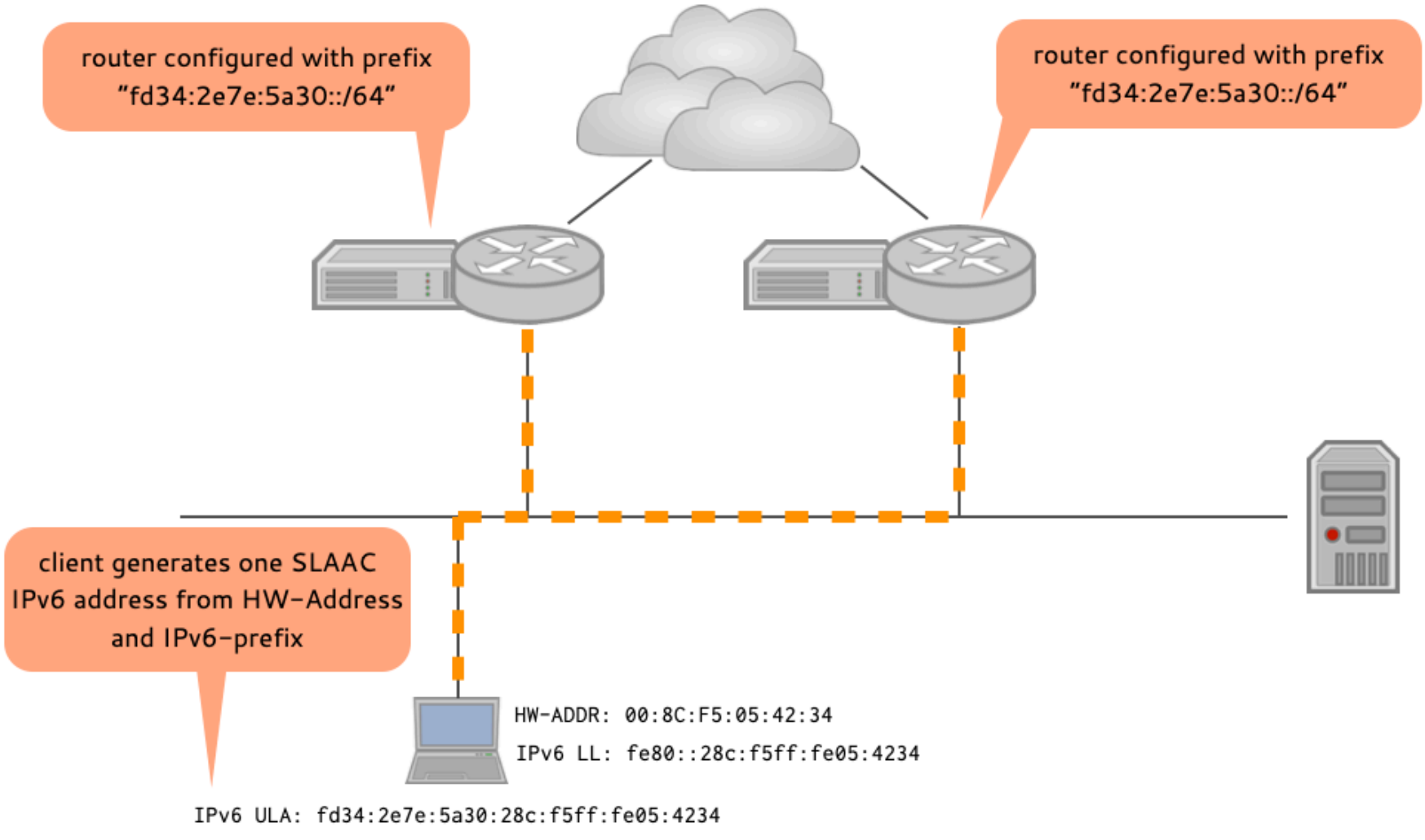
Lifetime of the DNS resolver

Lifetime of the DNS search list

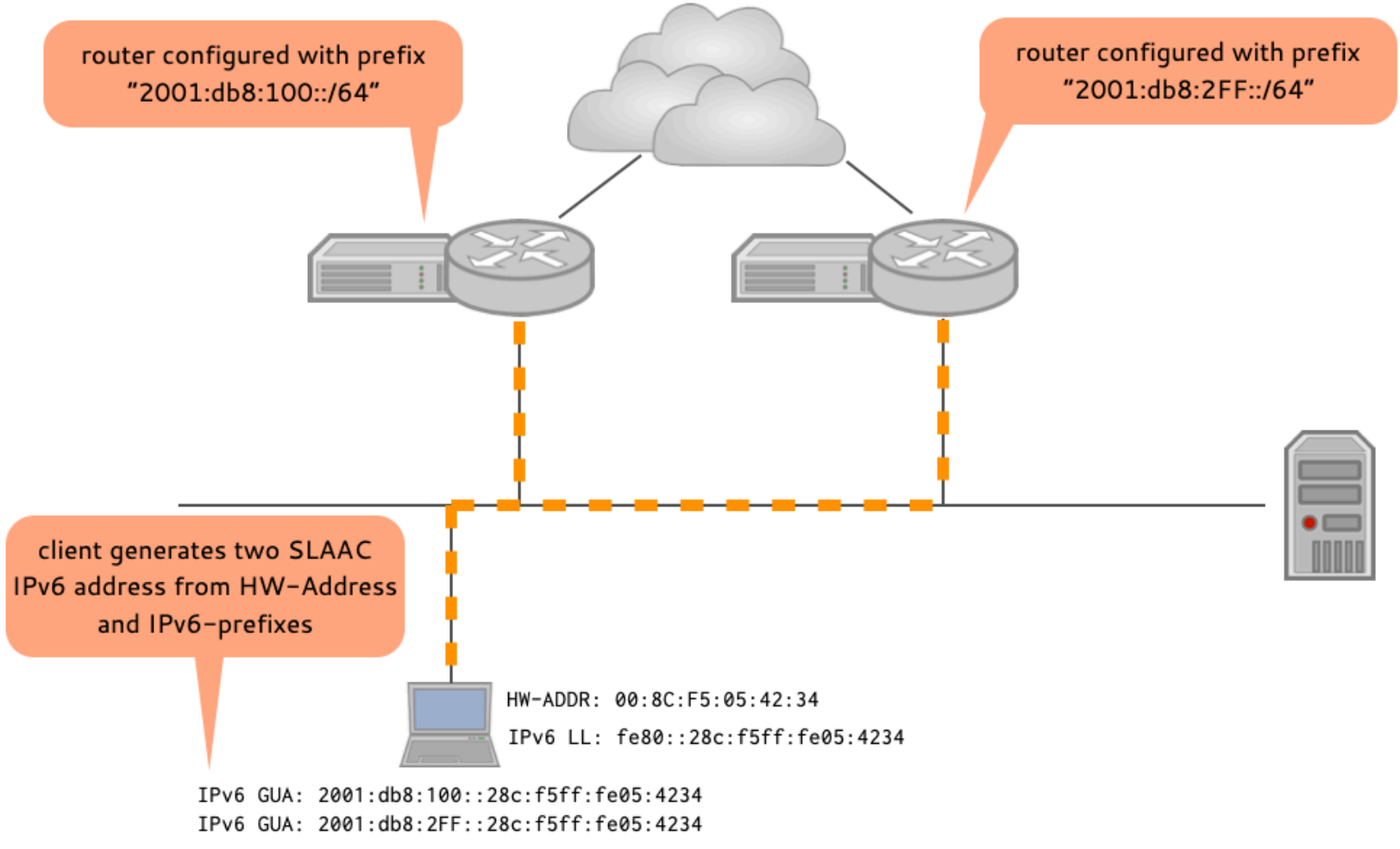
Router Advertisements

- RDNSS and DNSSL (↪RFC 8106 “IPv6 Router Advertisement Options for DNS Configuration”) is not supported on Microsoft Windows OS before Windows 10
  - Therefore networks with Windows clients require DHCPv6 for DNS resolver configuration.
- RDNSS and DNSSL works on Windows 10+, BSD, Linux, macOS, Android and Apple iOS


router configured with
multiple prefix
"fd34:2e7e:5a30::/64"
"2001:db8:100::/64"
client generates two SLAAC
IPv6 address from HW-Address
and IPv6-prefixes
HW-ADDR: 00:8C:F5:05:42:34
IPv6 LL: fe80::28c:f5ff:fe05:4234
IPv6 GUA: 2001:db8:100::28c:f5ff:fe05:4234
IPv6 ULA: fd34:2e7e:5a30:28c:f5ff:fe05:4234


router configured with prefix
"fd34:2e7e:5a30::/64"
router configured with prefix
"fd34:2e7e:5a30::/64"
client generates one SLAAC
IPv6 address from HW-Address
and IPv6-prefix
HW-ADDR: 00:8C:F5:05:42:34
IPv6 LL: fe80::28c:f5ff:fe05:4234
IPv6 ULA: fd34:2e7e:5a30:28c:f5ff:fe05:4234

- Two router with different GUA prefixes



SLAAC Addressing Problems

Stateless Address Autoconfiguration (SLAAC) for IPv6

- Originally SLAAC results in hosts configuring one or more *stable* addresses composed of a network prefix advertised by a local router, and an Interface Identifier (IID) that typically embeds a hardware address (e.g., an IEEE LAN MAC address)

Problems with stable IPv6 addresses created from hardware information (1)

- Because these Interface Identifiers do not vary over time, they allow correlation of host activities within the same network, thus negatively affecting the privacy of users

Problems with stable IPv6 addresses created from hardware information (2)

- The resulting Interface Identifiers are constant across networks, the resulting IPv6 addresses can be leveraged to track and correlate the activity of a host across multiple networks

Problems with stable IPv6 addresses created from hardware information (3)

- The use of hardware addresses reduce the search space when performing address-scanning attacks

Problems with stable IPv6 addresses created from hardware information (4)

- The hardware addresses convey information about the device, allow attackers to launch device-specific attacks

Problems with stable IPv6 addresses created from hardware information (5)

- Replacing the network card hardware results in a new IPv6 address for the host, possibly breaking existing configurations (e.g. for Server type machines)

Using Hardware-Addresses for client type IPv6 addresses is discouraged

- The use of hardware addresses to create IPv6 the IID (Interface ID) for an IPv6 address used by client type machines is discouraged
    - Alternatives:
        - IPv6 Privacy Extensions (↪RFC 8981 "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6")
        - Semantically Opaque Interface Identifiers (↪RFC 7217 "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)")
    - It is still fine for "server" type machines

IPv6 Privacy Extensions

IPv6 Privacy extensions

- With IPv6 stateless auto-configuration, IPv6 addresses might be generated from the hardware link-layer address (MAC-Address)
  - This address is stable for a long time
  - The host-id part of such an IPv6 address is not bound to the network location

The privacy issue with stable IPv6 IID

- **Problem**: by monitoring the addresses of IPv6 traffic, external parties can track the communication- and movement of a host
  - This is seen as a privacy issue by many users of IPv6 networks
  - See ↪RFC 7721 "Security and Privacy Considerations for IPv6 Address Generation Mechanisms" for a detailed discussion

IPv6 Privacy extensions

- RFC 8981 (↪RFC 8981 "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6") defines the privacy extensions for IPv6
  - A way to generate short lived IPv6 addresses with random, but still unique, host-id
    - host-id = lower 64bit of the IPv6 address

IPv6 Privacy extensions

- With IPv6 privacy extensions enabled, a host will use
  - The IPv6 addresses derived from the link-layer address for *local* communication
  - The IPv6 addressed with a random host-id part for communication with machines in the Internet or other external networks

IPv6 Privacy Extensions

- IPv6 privacy extensions are available in Windows (since Vista), Linux, macOS, Solaris and BSD IPv6 stacks
  - They are enabled by default on Windows client, Linux "Desktop" Distributions and MacOS X machines
  - But disabled on Windows Server OS systems, Linux "Server" Distributions (e.g. Red Hat EL) and all Unix/BSD systems

IPv6 Privacy extensions Linux

```
# nano /etc/sysctl.conf

[...]
# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1

# enable IPv6 privacy extensions
net.ipv6.conf.eth0.use_tempaddr=2

[...]
```

IPv6 Privacy extensions Linux

```
# ifconfig
p6p1   Link encap:Ethernet  HWaddr E8:9A:8F:8A:D2:A5
       inet addr:192.168.1.35  Bcast:192.168.1.255  Mask:255.255.255.0
       inet6 addr: fd34:2e7e:5a30:0:ea9a:8fff:fe8a:d2a5/64 Scope:Global
       inet6 addr: 2001:db8:2b6:0:ea9a:8fff:fe8a:d2a5/64 Scope:Global
       inet6 addr: fe80::ea9a:8fff:fe8a:d2a5/64 Scope:Link
       inet6 addr: fd34:2e7e:5a30:0:b983:40fb:85a9:dfa4/64 Scope:Global
       inet6 addr: 2001:db8:2b6:0:b983:40fb:85a9:dfa4/64 Scope:Global
       inet6 addr: 2001:db8:2b6:0:5db7:a8d1:6ff9:37cb/64 Scope:Global
       inet6 addr: fd34:2e7e:5a30:0:5db7:a8d1:6ff9:37cb/64 Scope:Global
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:33015 errors:0 dropped:0 overruns:0 frame:0
       TX packets:24121 errors:0 dropped:0 overruns:0 carrier:1
       collisions:0 txqueuelen:1000
       RX bytes:36079454 (34.4 MiB)  TX bytes:7071904 (6.7 MiB)
       Interrupt:45
```

IPv6 Privacy extensions macOS

```
#  more /etc/sysctl.conf
net.inet6.ip6.use_tempaddr=1
```

IPv6 Privacy extensions macOS

```
# ifconfig
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
  options=27<RXCSUM,TXCSUM,VLAN_MTU,TSO4>
  ether 00:26:b0:d6:a4:e0
  inet6 fe80::226:b0ff:fed6:a4e0%en0 prefixlen 64 scopeid 0x4
  inet6 2a01:198:2b6::226:b0ff:fed6:a4e0 prefixlen 64 autoconf
  inet6 2a01:198:2b6::11a5:386b:ca06:7d8 prefixlen 64 autoconf temporary
  inet6 fd34:2e7e:5a30::226:b0ff:fed6:a4e0 prefixlen 64 autoconf
  inet6 fd34:2e7e:5a30::313c:1c53:3bf1:6753 prefixlen 64 autoconf temporary
  inet 192.168.1.27 netmask 0xffffff00 broadcast 192.168.1.255
  media: autoselect (100baseTX <full-duplex,flow-control>)
  status: active
```

Stable Privacy-Enhanced Addresses

Stable Privacy-Enhanced Addresses

- ↪RFC 7217 “A Method for Generating Semantically Opaque Interface Identifiers” (April 2014) defines a method to create random host-ids that are
  - Different (random) for every network
  - Stay stable over time for each network
  - Supported by MacOS, Windows 10+, "Desktop" Linux Systems (NetworkManager/Systemd-Networkd)

IPv6 Privacy extensions on Windows

```
C:\Users\example>ipconfig /all

[...]
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : win.home.example.com
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Physical Address. . . . . . . . . : 08-00-27-33-8A-DF
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2001:db8:100:0:59ce:9045:5880:aaa(Preferred)
   IPv6 Address. . . . . . . . . . . : fd34:2e7e:5a30:0:59ce:9045:5880:aaa(Preferred)
   Temporary IPv6 Address. . . . . . : 2001:db8:100:0:503a:3cd7:56c2:a32d(Preferred)
   Temporary IPv6 Address. . . . . . : fd34:2e7e:5a30:0:503a:3cd7:56c2:a32d(Preferred)
   Link-local IPv6 Address . . . . . : fe80::59ce:9045:5880:aaa%11(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.1.150(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Freitag, 14. Januar 2011 11:22:25
   Lease Expires . . . . . . . . . . : Freitag, 14. Januar 2011 12:42:25
   Default Gateway . . . . . . . . . : fe80::20b:2fff:fee1:ac2f%11
                                       192.168.1.240
   DHCP Server . . . . . . . . . . . : 192.168.1.2
   DHCPv6 IAID . . . . . . . . . . . : 235405351
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-12-40-47-2B-08-00-27-9F-67-40

   DNS Servers . . . . . . . . . . . : fd34:2e7e:5a30::1
                                       192.168.1.2
                                       192.168.1.240
```

IPv6 Hardware IIDs on Windows Clients

```
C:\>netsh interface ipv6 set global randomizeidentifiers=disabled
Ok.
C:\>ipconfig /all

Windows IP Configuration
[...]
Ethernet adapter Local Area Connection:

[…]
   Physical Address. . . . . . . . . : 08-00-27-33-8A-DF
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2a01:198:2b6:0:a00:27ff:fe33:8adf(Preferred)
   IPv6 Address. . . . . . . . . . . : fd34:2e7e:5a30:0:a00:27ff:fe33:8adf(Preferred)
   Temporary IPv6 Address. . . . . . : 2a01:198:2b6:0:503a:3cd7:56c2:a32d(Preferred)
   Temporary IPv6 Address. . . . . . : fd34:2e7e:5a30:0:503a:3cd7:56c2:a32d(Preferred)
   Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe33:8adf%11(Preferred)
```

IPv6 Hardware IIDs on Windows Clients

```
C:\>netsh interface ipv6 set privacy state=disable store=persistent
Ok.
C:\>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : win.home.example.com
   IPv6 Address. . . . . . . . . . . : 2001:db8:100:0:59ce:9045:5880:aaa
   IPv6 Address. . . . . . . . . . . : fd34:2e7e:5a30:0:59ce:9045:5880:aaa
   Link-local IPv6 Address . . . . . : fe80::59ce:9045:5880:aaa%11
   IPv4 Address. . . . . . . . . . . : 192.168.1.150
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::20b:2fff:fee1:ac2f%11
                                       192.168.1.240
```

Quiz

- which addresses below are (most likely) not privacy addresses?
  - `2001:db8::5efe:169.254.10.170`
  - `2001:db8:2b6:0:5db7:a8d1:6ff9:37cb`
  - `fd34:2e7e:5a30:0:ea9a:8fff:fe8a:d2a5`
  - `2a01:198:2b6::226:b0ff:fed6:a4e0`

Questions?