Exploring IPv6 at Home

Preparing for the Future of the Internet Protocol

Matt Layher

Orchestructure, March 27, 2019

Matt Layher

- Engineer at Fastly*
- GitHub + Twitter: @mdlayher
- mdlayher.com
- github.com/mdlayher/talks

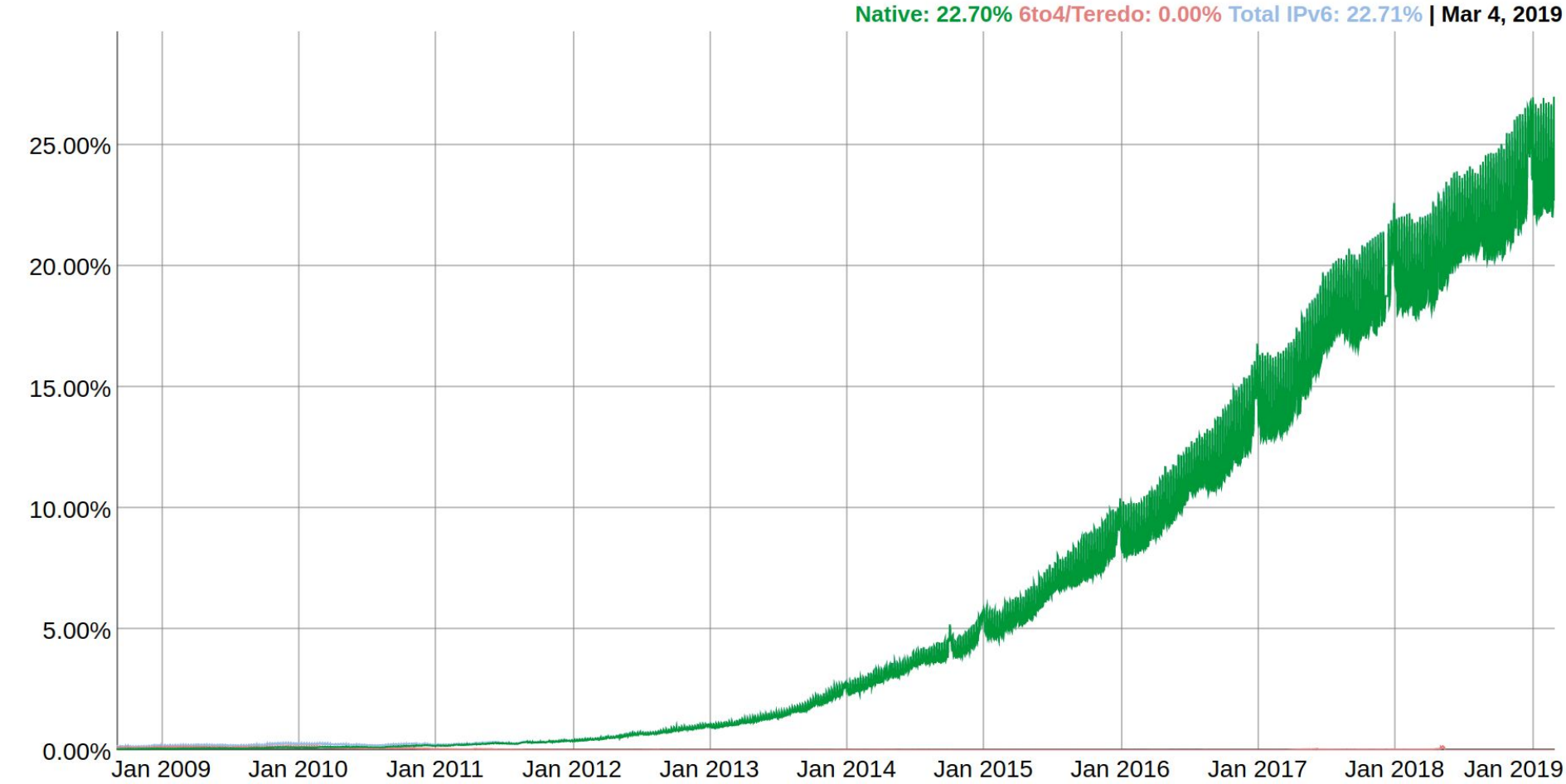
* **Disclaimer:** this talk is entirely based on my own experiences, and is **not related to Fastly in any way!**

Agenda

- Introduce some of the fundamental concepts of IPv6 (RFC 8200)
- Explain the differences between IPv6 and IPv4
- Demonstrate how to set up IPv6 at home with a Ubiquiti® EdgeRouter™ device
    - Comcast and Charter/Spectrum should be fine

Introduction to IPv6

How many audience members are using IPv6 at home?



User IPv6 adoption on Google services: google.com/intl/en/ipv6/statistics.html

IPv6: more than just a bigger address space

- **128 bit** IP addresses: **64 bits** for networks, **64 bits** for hosts
- IPv6-enabled interfaces **will** have multiple addresses

Global unicast `2600:6c4a:787f:d200::/56`

Unicast local `fd00::/7`

Link local* `fe80::/10`

* Link local IPv6 addresses are **mandatory** and **unique per link**

“... you can give every millimetre from one side of the universe to the opposite side of the universe in a straight line about 3.6 billion IPv6 addresses...”

- redditor /u/Accendil

reddit.com/r/theydidthemath/comments/2qxgxw/self_just_how_big_is_ipv6

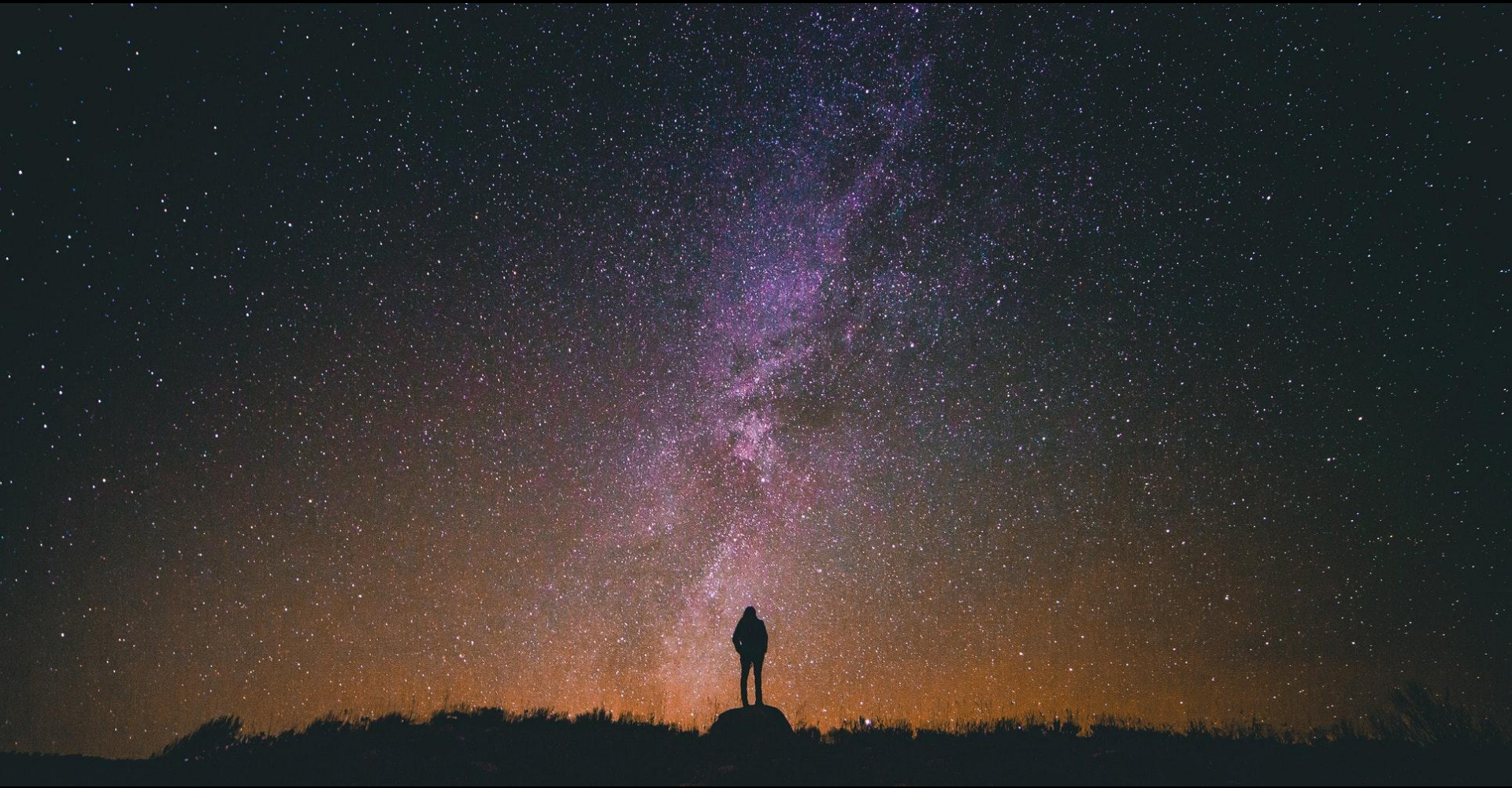


Image credit: unsplash.com/@grakozy

IPv6 address assignment

- Modified EUI-64 format:
  - MAC address: `06:cb:90:4d:a2:59`
  - IPv6 address: `fe80::4cb:90ff:fe4d:a259/64`
- Temporary: `2600:6c4a:787f:d200:203e:861b:b6c6:e6de/64`
  - IPv6 privacy extensions

IPv6 prefix delegation

- Your ISP can route an **entire IPv6 prefix** to your house
  - If you're paying $$$, maybe you'll get a handful of static IPv4 addresses
  - Spectrum: **/56 IPv6 prefix, 256 networks of 2^{64} addresses**
    - NAT no longer a necessity

IPv6 tips and tricks

- Many shell utilities have a “`-6`” flag to use IPv6
- A useful website for testing IPv6 configuration: ipv6-test.com
- My favorite ping target:

```
$ ping -6 2600::
PING 2600::(2600::) 56 data bytes
64 bytes from 2600::: icmp_seq=1 ttl=48 time=56.9 ms
```

How can I obtain an IPv6 prefix at home?

DHCPv6 Prefix Delegation (DHCPv6-PD)

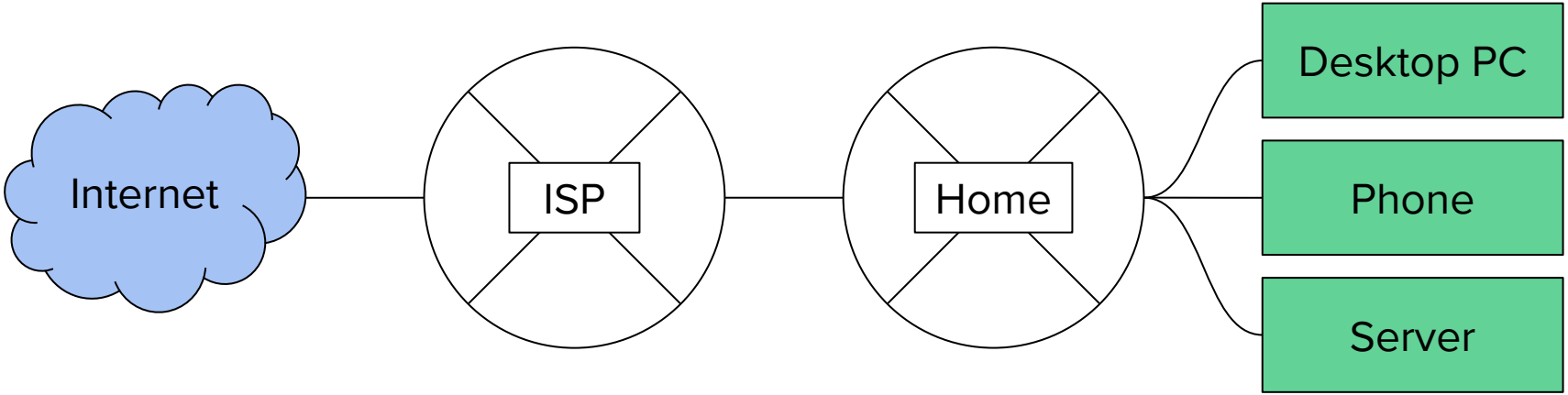
What is DHCPv6-PD?

- Uses DHCPv6 (RFC8415) to request a prefix assignment from your ISP
    - IPv6 traffic with a matching prefix is **delegated to your router**
- A popular method for ISPs to enable **dual-stack** (IPv4 + IPv6) network access

`2600:6c4a:787f:d200::/56`

DHCPv6-PD in action

- Home network is currently set up for IPv4 only



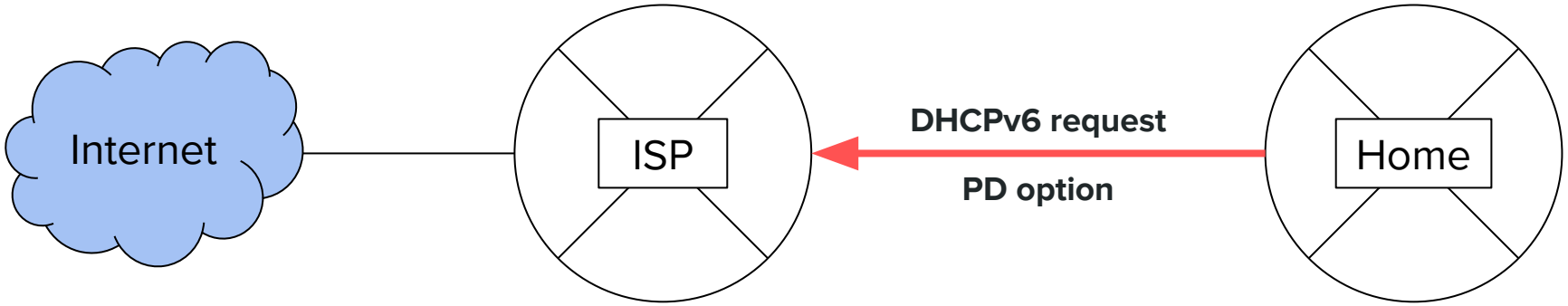

```
$ ping -6 2600::
connect: Network is unreachable
```

IPv4

Image credit: unsplash.com/@cbyoung

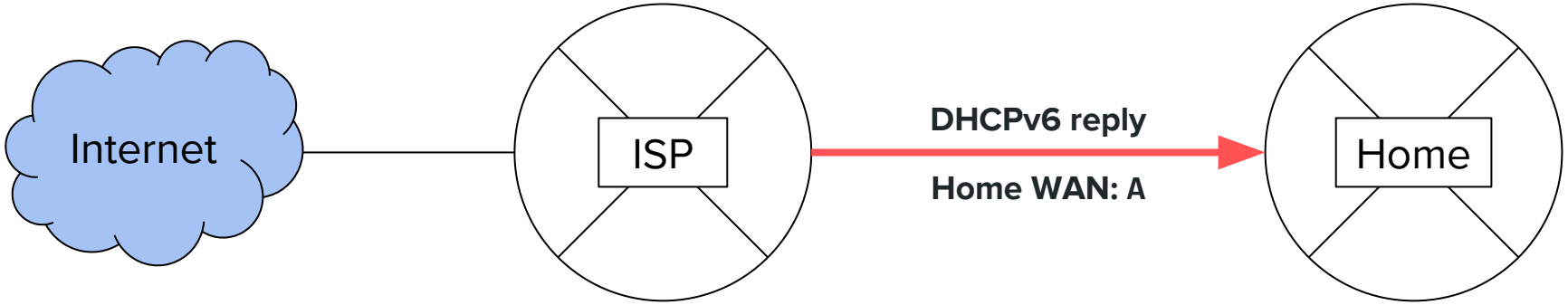
DHCPv6-PD in action

- Home router uses DHCPv6 to request IPv6 connectivity from ISP



DHCPv6-PD in action

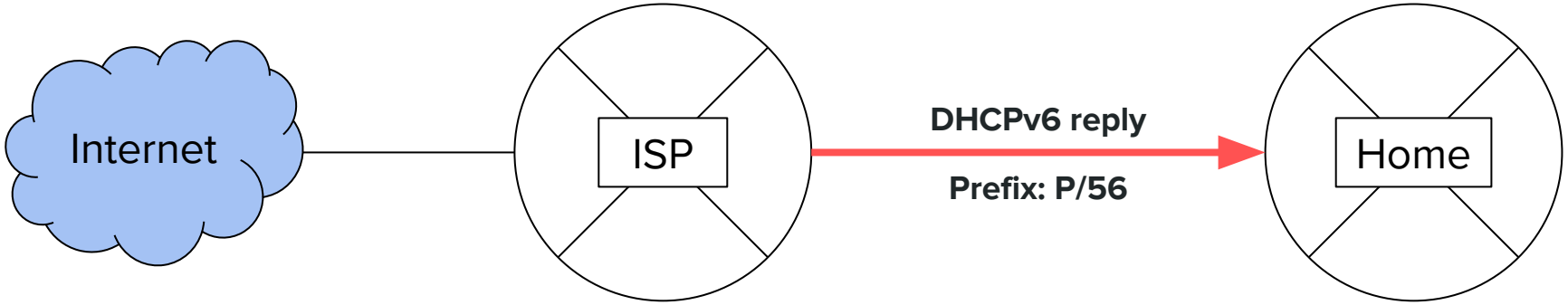
- ISP router assigns IPv6 address **A** to home router's WAN interface



A: `2600:6c4a:7002:100:d59c:8634:4669:65db/128`

DHCPv6-PD in action

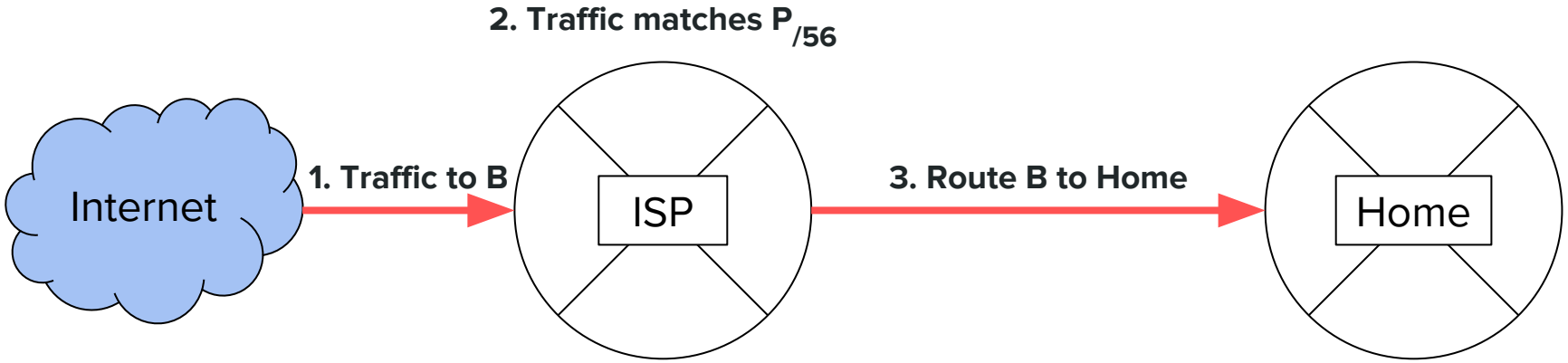
- ISP router delegates and routes IPv6 prefix **$P_{/56}$** to home router



$P_{/56}$: `2600:6c4a:787f:d200::/56`

DHCPv6-PD in action

- ISP router routes IPv6 traffic with matching prefixes to home router



$P_{/56}$: `2600:6c4a:787f:d200::/56`
B: `2600:6c4a:787f:d200::1/128`

How can I distribute addresses from my IPv6 prefix?

IPv6 Neighbor Discovery Protocol (NDP)

What is NDP?

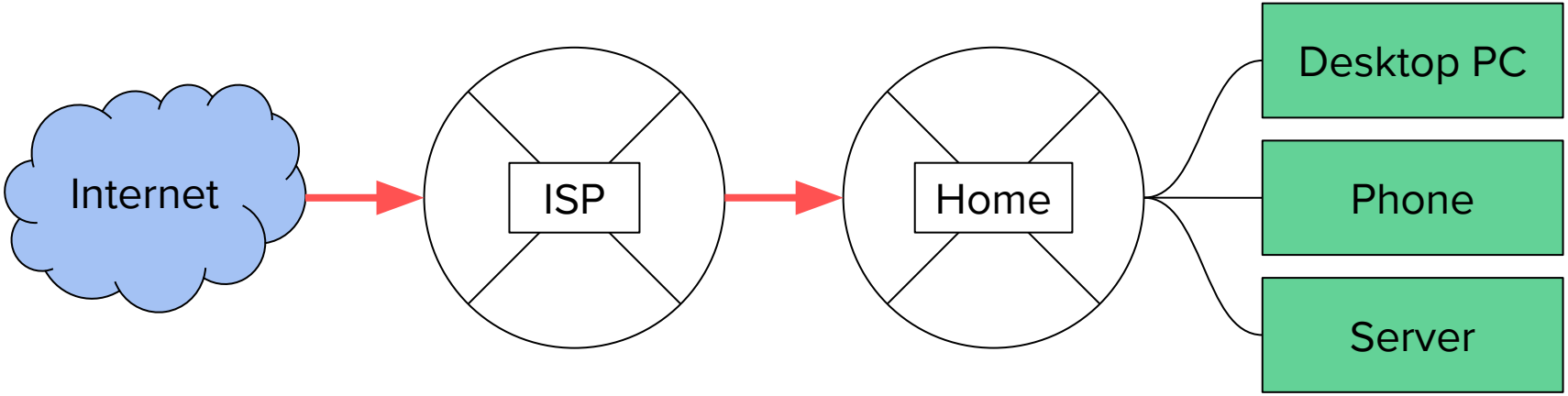
- Akin to an “IPv6 ARP”, but uses ICMPv6 with link-local addresses
- Ask a network neighbor for its MAC address using its IPv6 address:
  - `fd00::4cb:90ff:fe4d:a259:` Who has “**`fd00::1`**”?
  - `fd00::1:` I am at “**`04:18:d6:a1:ce:b7`**”.

IPv6 and NDP’s big advantage

- **DHCPv6 is not necessary** to configure globally-routable IPv6 addresses
- **Stateless Address Autoconfiguration (SLAAC) via NDP**
    - Clients can fetch additional configuration from DHCPv6, if configured

NDP + SLAAC in action

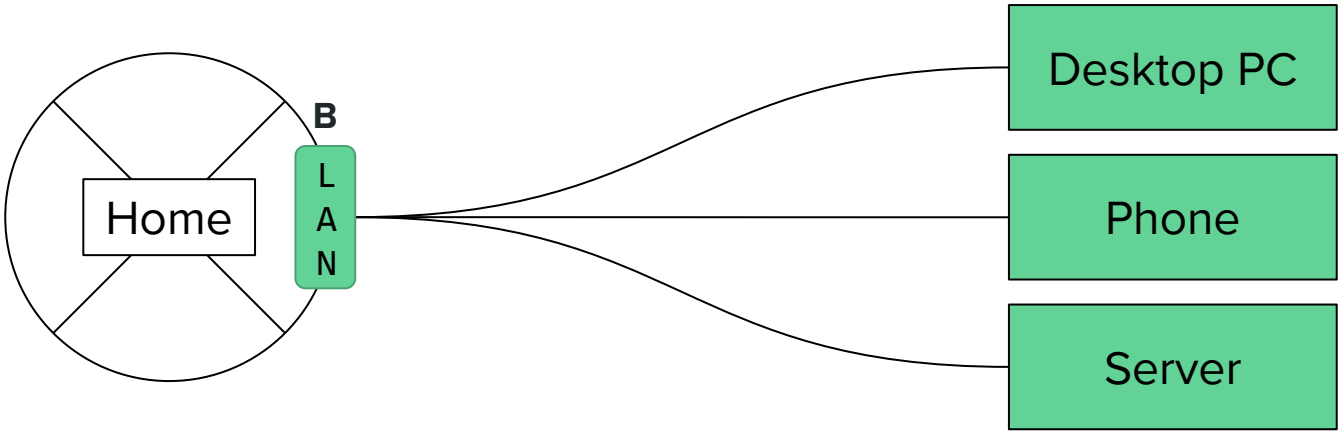
- Home router has been delegated an IPv6 prefix by ISP router



$P_{/56}$: `2600:6c4a:787f:d200::/56`

NDP + SLAAC in action

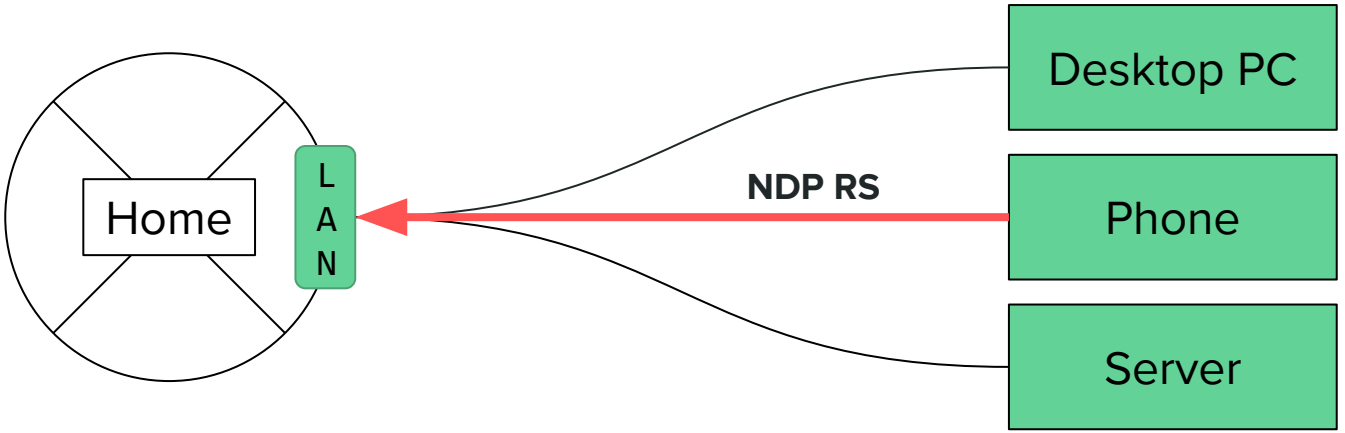
- Home router creates a network within **P**$_{/56}$, creates **P**$_{/64}$, and assigns IPv6 address **B** to its LAN interface



P$_{/64}$: `2600:6c4a:787f:d200::/64`
B: `2600:6c4a:787f:d200::1/128`

NDP + SLAAC in action

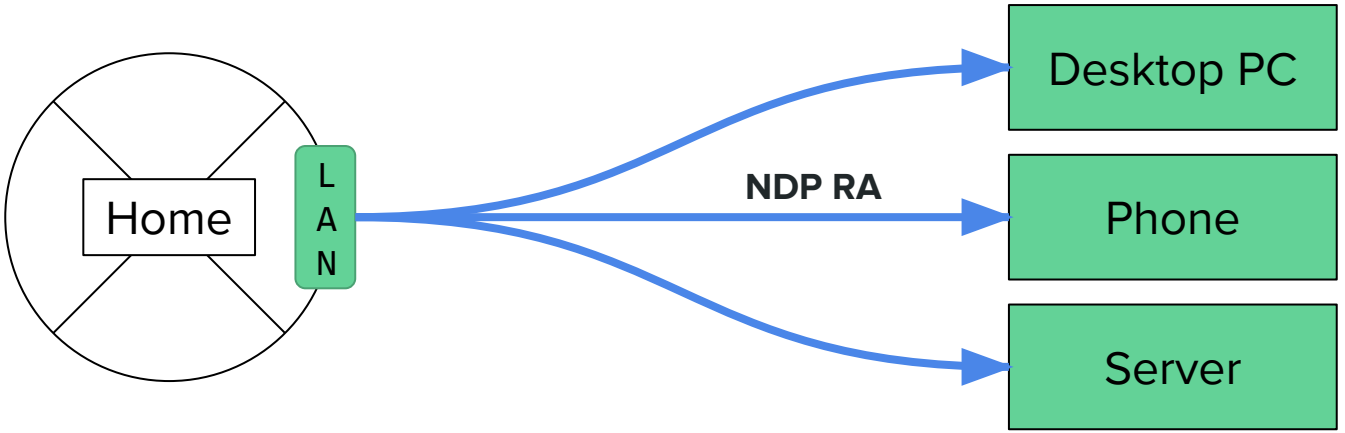
- Devices send a multicast **router solicitation** to discover IPv6 routers



RS: “Any IPv6 routers out there?”

NDP + SLAAC in action

- Home router sends a multicast **router advertisement** for prefix **P**$_{/64}$



RA: “I’m an IPv6 router, use prefix `2600:6c4a:787f:d200::/64`, **use SLAAC,** your address is valid for 24 hours.”

NDP + SLAAC in action

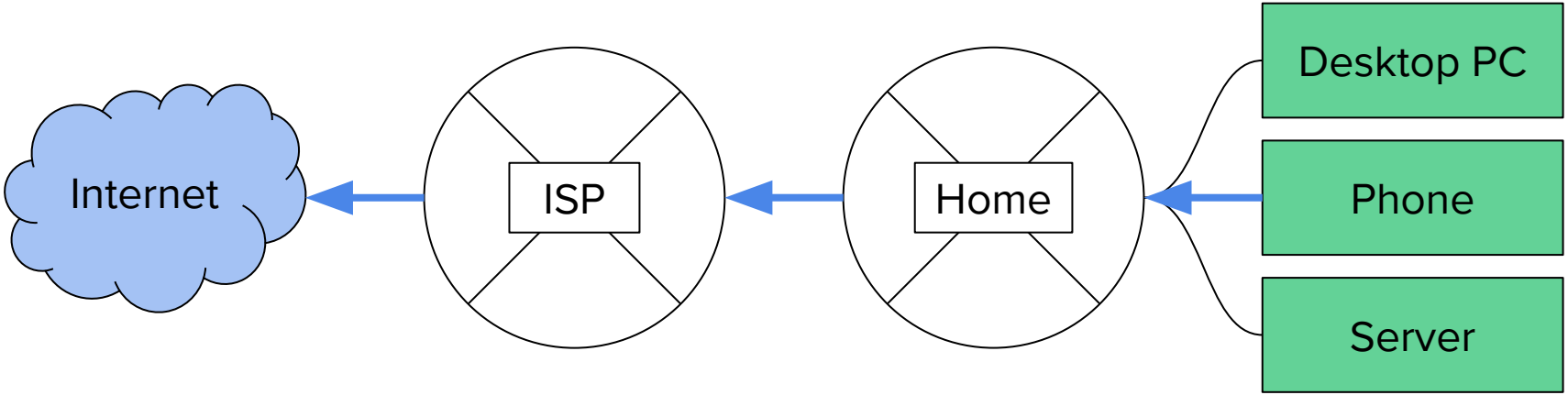
- Each device computes and assigns one or more address from prefix $\mathbf{P}_{/64}$

Desktop PC	2600:6c4a:787f:d200:**76d4:35ff:fee7:cbc4**/64
Phone	2600:6c4a:787f:d200:**3e28:6dff:fe0f:43bd**/64
Server	2600:6c4a:787f:d200:**04cb:90ff:fe4d:a259**/64

$\mathbf{P}_{/64}$: 2600:6c4a:787f:d200::/64

NDP + SLAAC in action

- Home LAN with **globally routable addresses** now has IPv6 internet access




```
$ ping -6 2600::
PING 2600::(2600::) 56 data bytes
64 bytes from 2600::: icmp_seq=1 ttl=48 time=56.9 ms
```

IPv6

Image credit: unsplash.com/@chuttersnap

NDP and Go resources

- Visit mdlayher.com for an index, and see also:
  - github.com/mdlayher/ndp
  - Network Protocol Breakdown: NDP and Go
  - GopherCon 2018: Implementing a Network Protocol in Go

How can I set up IPv6 on my home network?

Configuring IPv6 on a Ubiquiti® EdgeRouter™ device

Introducing our router of choice

- The Ubiquiti® EdgeRouter™ Lite is powerful, customizable, and a great value

Image credit: ui.com/marketing/#edgemax

CLI vs web UI

- Not all IPv6 functionality is available in the web UI
- CLI config tree, but a full Linux CLI is also available by SSH’ing into the device

```
matt@routnerr-1:~$ uname -a
Linux routnerr-1 3.10.107-UBNT #1 SMP Wed Feb 13 18:23:28 UTC 2019 mips64 GNU/Linux
matt@routnerr-1:~$ w
 11:04:31 up 1 day, 23:03,  1 user,  load average: 0.45, 0.27, 0.16
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
matt     pts/1    fd00::2c1d:b96f: 11:04    2.00s  0.19s  0.03s w
```

```
matt@routnerr-1:~$ configure
[edit]
matt@routnerr-1# show system offload
 hwnat disable
 ipsec enable
 ipv4 {
     forwarding enable
     gre enable
     pppoe enable
     vlan enable
 }
 ipv6 {
     forwarding enable
     pppoe disable
     vlan enable
 }
[edit]
```

Step 0: the setup

- We'll need to configure a few things to make this all work:
  - IPv6 firewall **(important, you can't rely on NAT as a firewall!)**
  - DHCPv6-PD on WAN, SLAAC on LAN
- These slides will be based on my current router configuration
  - **`eth0`: LAN**, **`eth1`: WAN**
  - This blog post by Bradley Heilbrun might be the guide I originally used

Step 1: IPv6 firewall rules

- We will set up two named IPv6 firewalls on WAN interface:
  - Inbound IPv6 traffic to the router (`WAN6_LOCAL`)
  - Inbound IPv6 traffic to the LAN behind the router (`WAN6_IN`)

Step 1a: WAN6_LOCAL

- When IPv6 traffic arrives bound for our router's address, it should:
  - Allow established/related state
    - TCP responses to requests we initiate
  - Drop invalid state
    - Nonsensical traffic, ACK on closed TCP connections and etc.
  - Allow ICMPv6 to the router **(important!)**:
    - ping, NDP RAs from ISP, many more...
  - Allow DHCPv6 server responses
    - We need DHCPv6 and DHCPv6-PD to work

```
# set firewall ipv6-name WAN6_LOCAL
# edit firewall ipv6-name WAN6_LOCAL
[edit firewall ipv6-name WAN6_LOCAL]

# set default-action drop
# set description “IPv6 WAN to router”

# set rule 10 action accept
# set rule 10 description “Allow established/related”
# set rule 10 state established enable
# set rule 10 state related enable

# set rule 20 action drop
# set rule 20 description “Drop invalid state”
# set rule 20 state invalid enable
```

```
# set rule 30 action accept
# set rule 30 description “Allow ICMPv6”
# set rule 30 protocol icmpv6

# set rule 40 action accept
# set rule 40 description “Allow DHCPv6”
# set rule 40 destination port 546
# set rule 40 protocol udp
# set rule 40 source port 547

# up
[edit firewall]
```

Step 1b: `WAN6_IN`

- When IPv6 traffic arrives bound for our LAN subnet, it should:
  - Allow established/related state
  - Drop invalid state
  - Allow ICMPv6 to the LAN **(important!)**
  - *Optional:* allow individual TCP/UDP services
    - SSH, HTTP, and HTTPS

```
# set ipv6-name WAN6_IN
# edit ipv6-name WAN6_IN
[edit firewall ipv6-name WAN6_IN]

# set default-action drop
# set description “IPv6 WAN to LAN”

# set rule 10
# set rule 10 action accept
# set rule 10 description “Allow established/related”
# set rule 10 state established enable
# set rule 10 state related enable

# set rule 20
# set rule 20 action drop
# set rule 20 description “Drop invalid state”
# set rule 20 state invalid enable
```

```
# set rule 30 action accept
# set rule 30 description “Allow ICMPv6”
# set rule 30 protocol icmpv6

# set rule 40 action accept
# set rule 40 description “Allow SSH to server”
# set rule 40 destination address [IPv6 address]
# set rule 40 destination port 22
# set rule 40 protocol tcp

# top
# set interfaces ethernet eth1 firewall in ipv6-name WAN6_IN
# set interfaces ethernet eth1 firewall local ipv6-name WAN6_LOCAL

# commit
# save
Saving configuration to '/config/config.boot'...
Done
```

Step 2: DHCPv6-PD on WAN interface

- Our router must request an IPv6 prefix via DHCPv6-PD on its WAN interface
  - Specify the prefix length: **`/56`**
  - Subnet the **first /64** out of our prefix for use
  - Set up address **`P::1/128`** on the router for convenience
  - Specify that we should use **SLAAC** to distribute addresses on the LAN
  - *Recommended:* ignore ISP's DNS servers

```
# edit interfaces ethernet eth1
[edit interfaces ethernet eth1]

# set dhcpv6-pd pd 0 prefix-length 56
# set dhcpv6-pd pd 0 interface eth0 prefix-id :0
# set dhcpv6-pd pd 0 interface eth0 host-address ::1
# set dhcpv6-pd pd 0 interface eth0 service slaac
# set dhcpv6-pd no-dns

# top
# commit
# save
Saving configuration to '/config/config.boot'...
Done
```

Step 3: verify IPv6 connectivity on router

- Check interface status to see IPv6 address assignments
- Ping the IPv6 internet

```
matt@routnerr-1:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface    IP Address                        S/L  Description
---------    ----------                        ---  -----------
eth0         192.168.1.1/24                    u/u  LAN
             2600:6c4a:787f:d200::1/64
             fd00::1/64
eth1         24.176.13.215/22                  u/u  WAN
             2600:6c4a:7002:100:d59c:8634:4669:65db/128
eth2         -                                 A/D
lo           127.0.0.1/8                       u/u
             ::1/128

matt@routnerr-1:~$ ping6 2600::
PING 2600::(2600::) 56 data bytes
64 bytes from 2600::: icmp_seq=1 ttl=49 time=48.8 ms
```

Step 4: verify IPv6 connectivity on LAN device

- Verify your router’s NDP RAs are reaching your LAN devices
- Ping the IPv6 internet
- Open an IPv6 test in your browser: ipv6-test.com

```
matt@nerr-2:~$ ndp rs
ndp> interface: enp5s0, link-layer address: 74:d4:35:e7:cb:c4, IPv6 address:
     fe80::76d4:35ff:fee7:cbc4
ndp rs> router solicitation:
    - source link-layer address: 74:d4:35:e7:cb:c4

ndp rs> router advertisement from: fe80::618:d6ff:fea1:ceb7:
    - hop limit:        64
    - preference:       Medium
    - router lifetime:  30m0s
    - options:
        - prefix information: 2600:6c4a:787f:d200::/64, flags: [OA], valid: 48h0m0s,
              preferred: 24h0m0s
        - prefix information: fd00::/64, flags: [OA], valid: 48h0m0s, preferred: 24h0m0s
        - source link-layer address: 04:18:d6:a1:ce:b7

matt@nerr-2:~$ ping 2600::
PING 2600::(2600::) 56 data bytes
64 bytes from 2600::: icmp_seq=1 ttl=48 time=47.5 ms
```

IPv4 connectivity

IPv4	Supported
Address	24.176.13.215
Hostname	24-176-13-215.dhcp.klmz.mi.charter.com
ISP	Spectrum

IPv6 connectivity

IPv6	Supported
Address	2600:6c4a:787f:d200:f4e8:edae:6ed1:ae15
Type	Native IPv6
SLAAC	No
ICMP	Reachable
Hostname	2600-6c4a-787f-d200-f4e8-edae-6ed1-ae15.dhcp6.chtrptr.net
ISP	Spectrum

Score

20 / 20

Browser

Default	IPv6
Fallback	to IPv4 in 1 second

DNS

DNS4 + IP6	Reachable
DNS6 + IP4	Reachable
DNS6 + IP6	Reachable

More

Speed test »

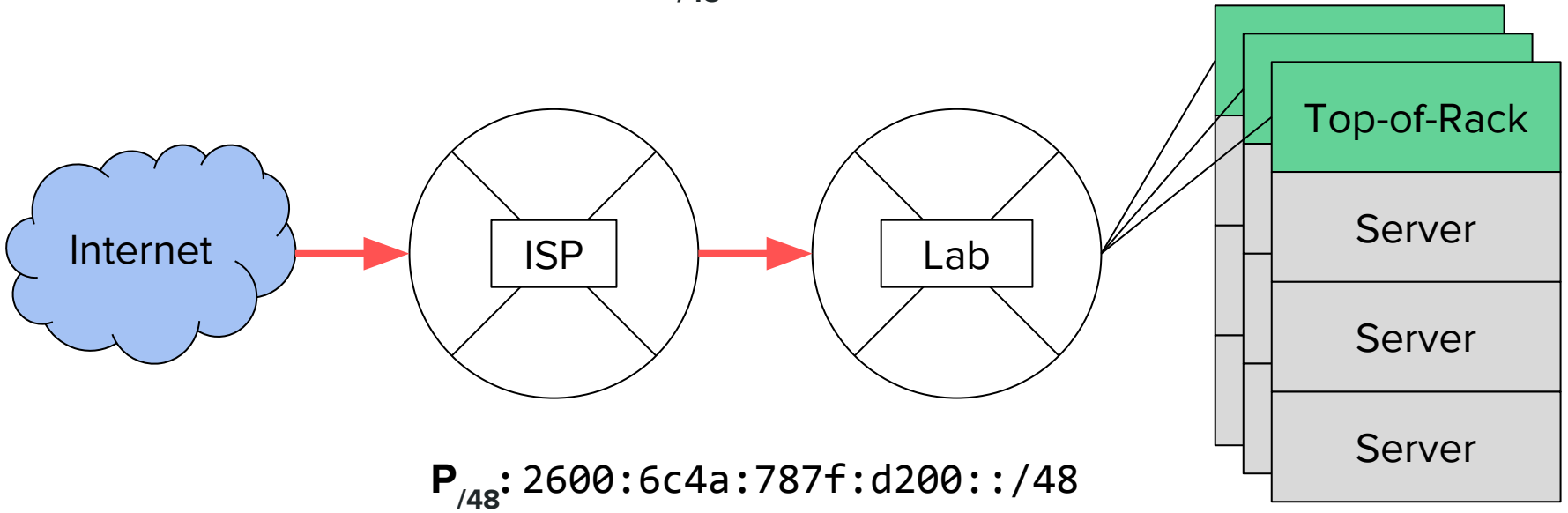
Ping test »

What if we take our IPv6 network a step further?*

* There are some caveats, but I think this would be fun to try in a lab!

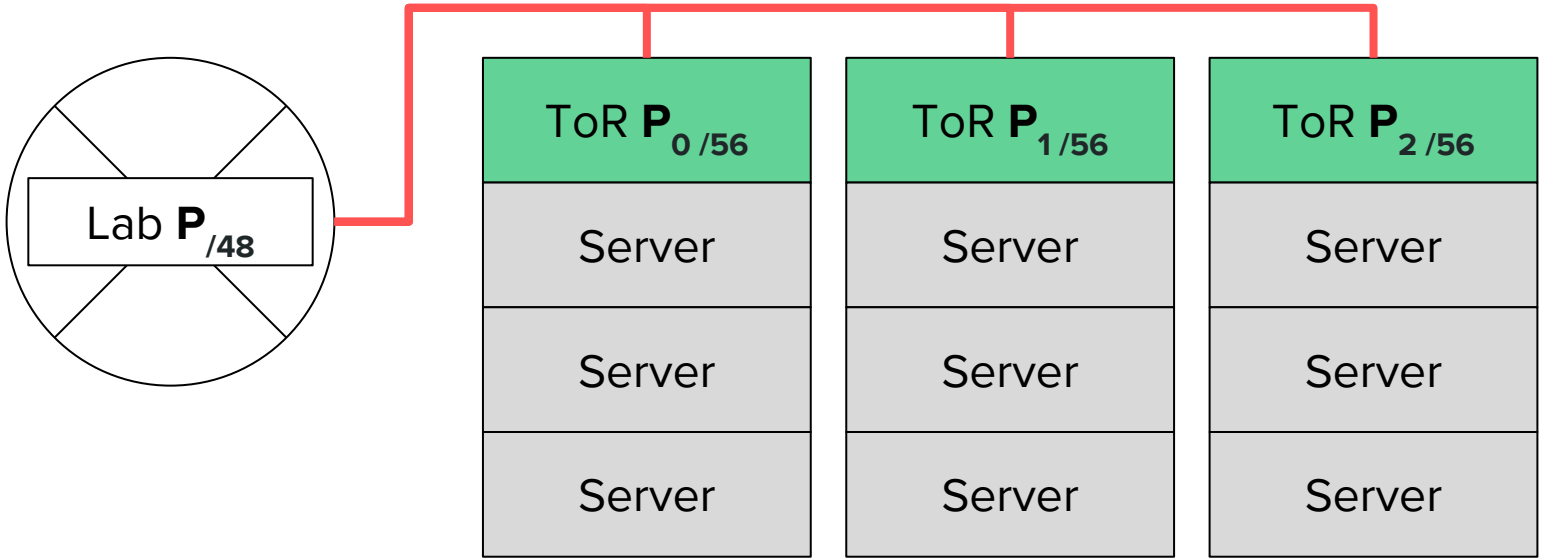
IPv6 in a larger network

- Lab router has been delegated **P**$_{/48}$ IPv6 prefix by ISP router



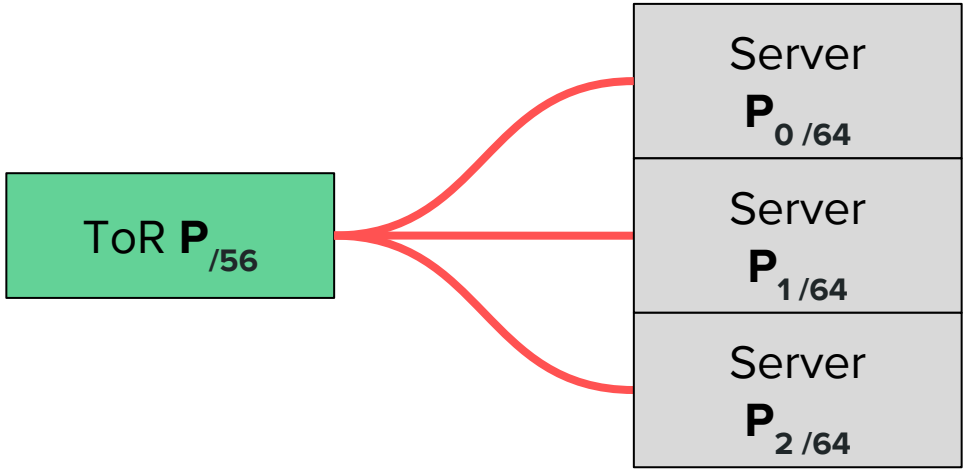
IPv6 in a larger network

- Lab router can use DHCPv6-PD to delegate and route individual **P**$_{/56}$ IPv6 prefixes to **Top-of-Rack (ToR)** devices



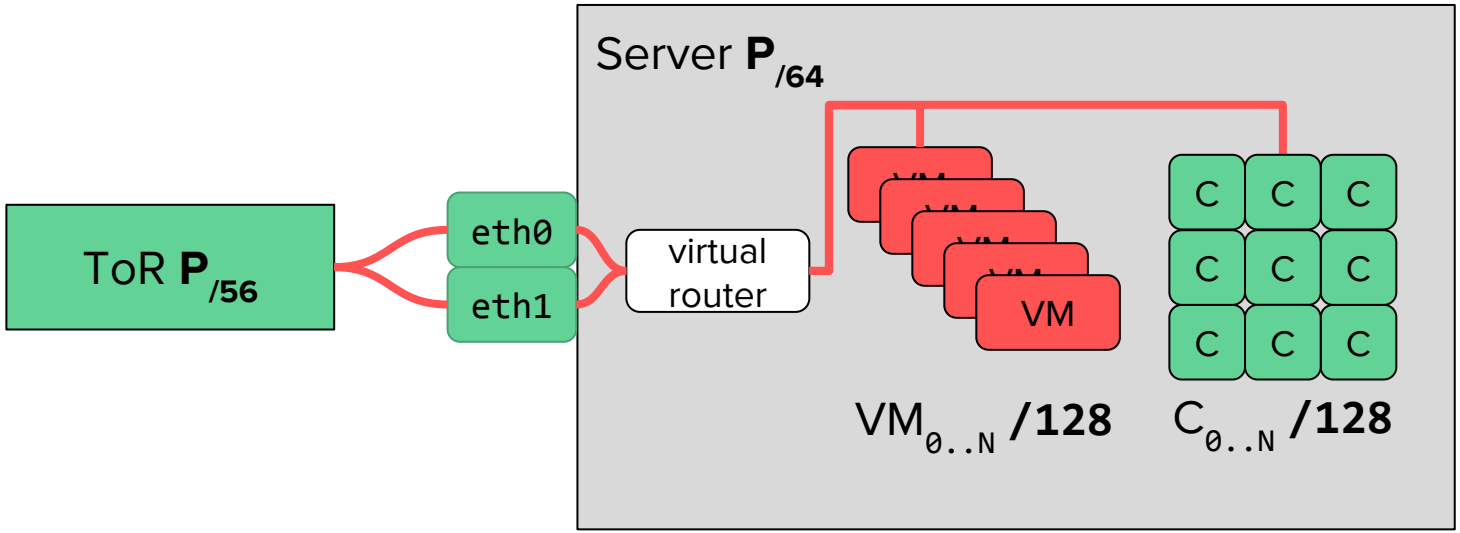
IPv6 in a larger network

- ToR devices can use DHCPv6-PD to delegate and route individual **$P_{/64}$** IPv6 prefixes to **servers**



IPv6 in a larger network

- Servers can use SLAAC to assign individual, **globally routable IPv6 addresses** to each of their **containers or virtual machines**, from their own **P**$_{/64}$ prefix



* **Caveat:** address mobility between servers would be limited in this setup: use a real routing protocol!

1 IPv4

Image credit: unsplash.com/@cbyoung


~∞ IPv6

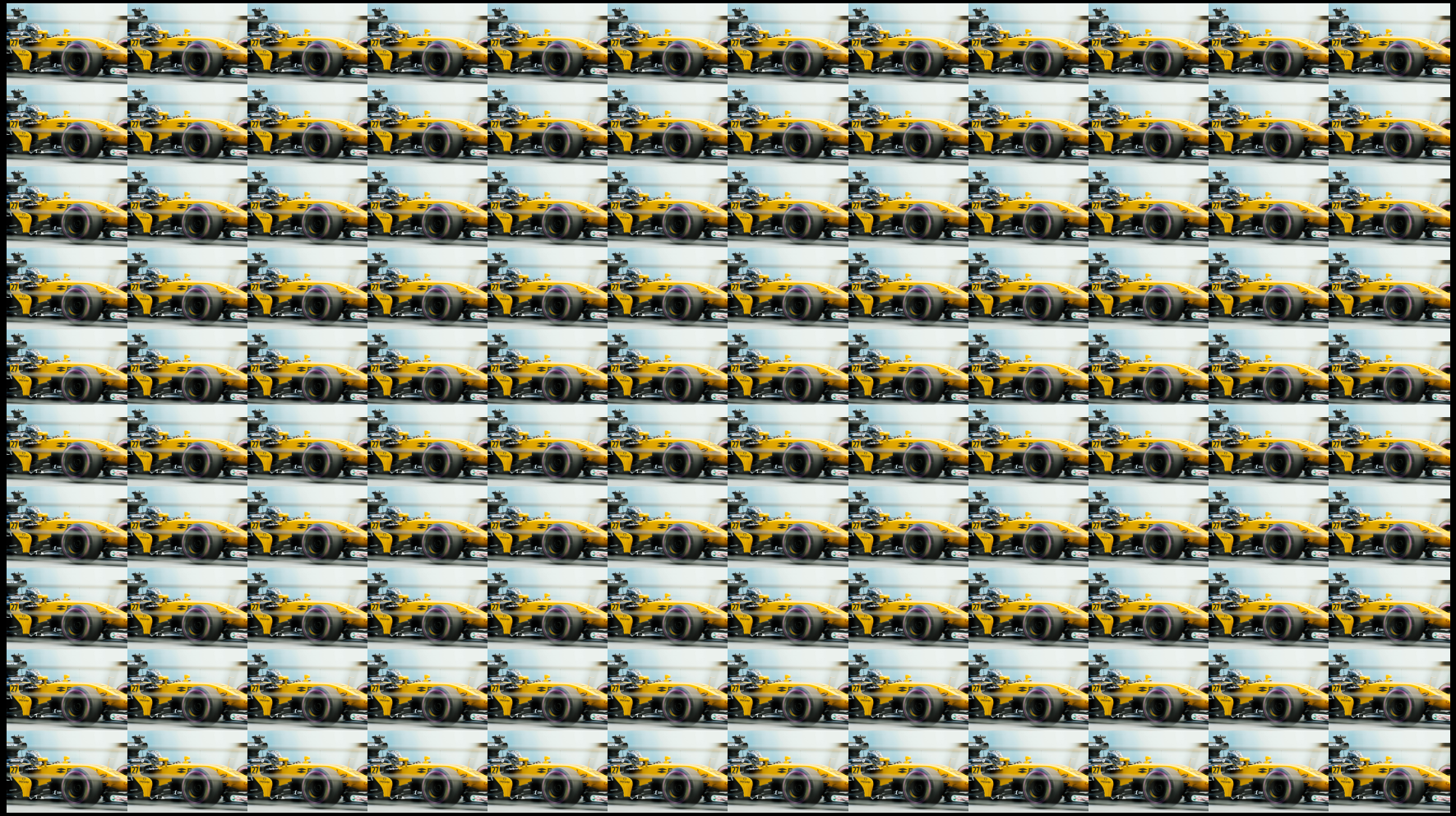


Image credit: unsplash.com/@chuttersnap

Thanks!

Matt Layher
mdlayher.com
github.com/mdlayher
twitter.com/mdlayher

Image credit: worldipv6launch.org