



RIPE NCC
RIPE NETWORK COORDINATION CENTRE

RIPE Meeting Network behind the scenes

How we run a conference network
for a networking conference

Ondřej Caletka | 24 January 2024 | CSNOG 2024

What is a RIPE meeting



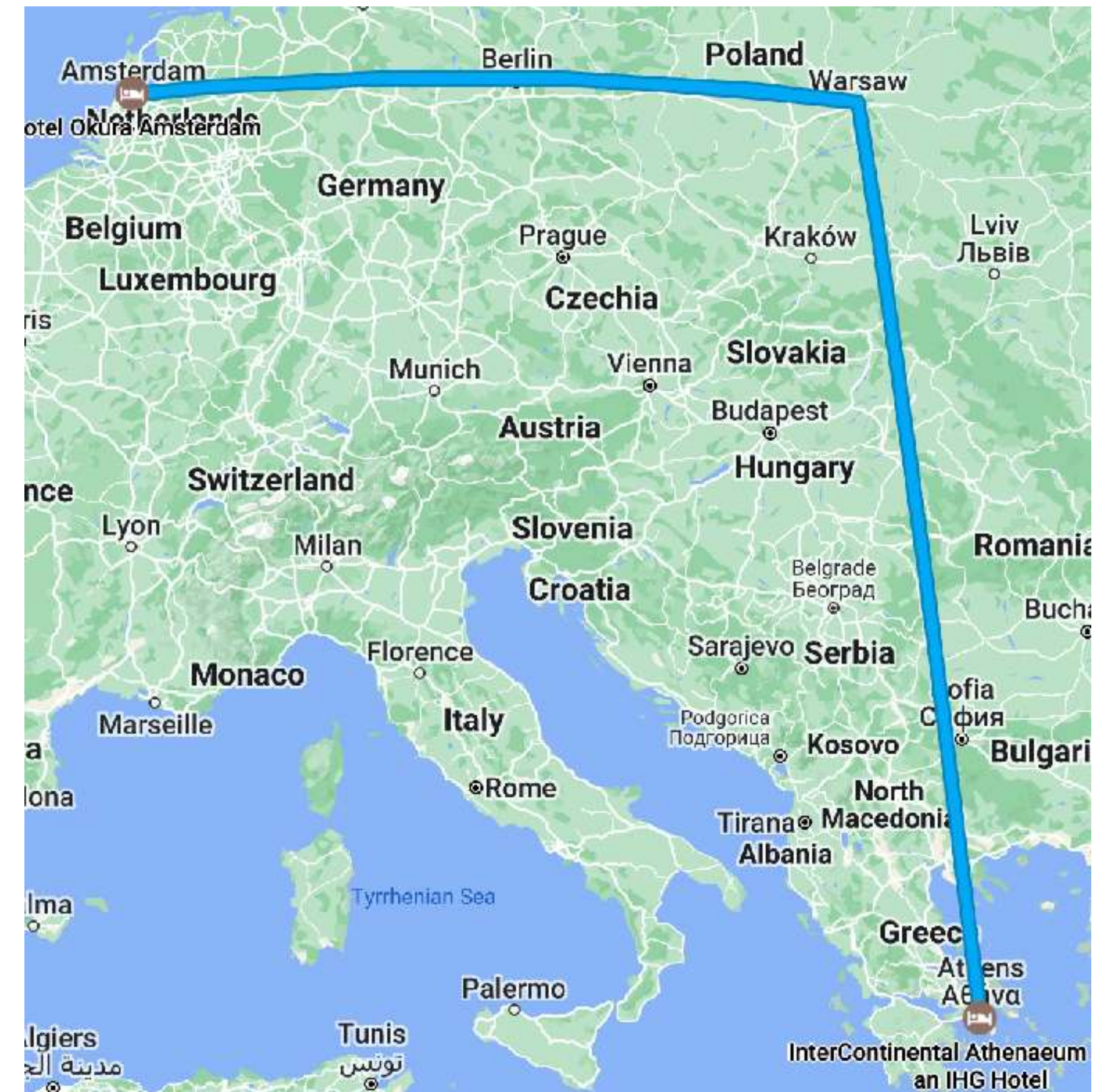
- A week long event twice a year along RIPE NCC service region
- 600+ attendees from all over the world
- **Next up:** RIPE 88 Krakow, 20 - 24 May 2024

- **Custom temporary Wi-Fi network during the event**
 - AS2121
 - 193.0.24.0/21
 - 2001:67c:64::/48

Issues with Geolocation



- BSSID-based geolocation
 - based on assumption that APs don't move
 - issues with Google **disappeared around 2016**
- IP-based geolocation
 - Privately curated lists by many commercial parties
 - We publish a CSV list for Google
 - This is now **standardised as RFC 8805**
 - Many providers support it but you still **have to tell them**; RFC 9092 discovery is not very popular
 - We have a list of 7 geolocation providers to check prior every meeting; **two still need manual updates**

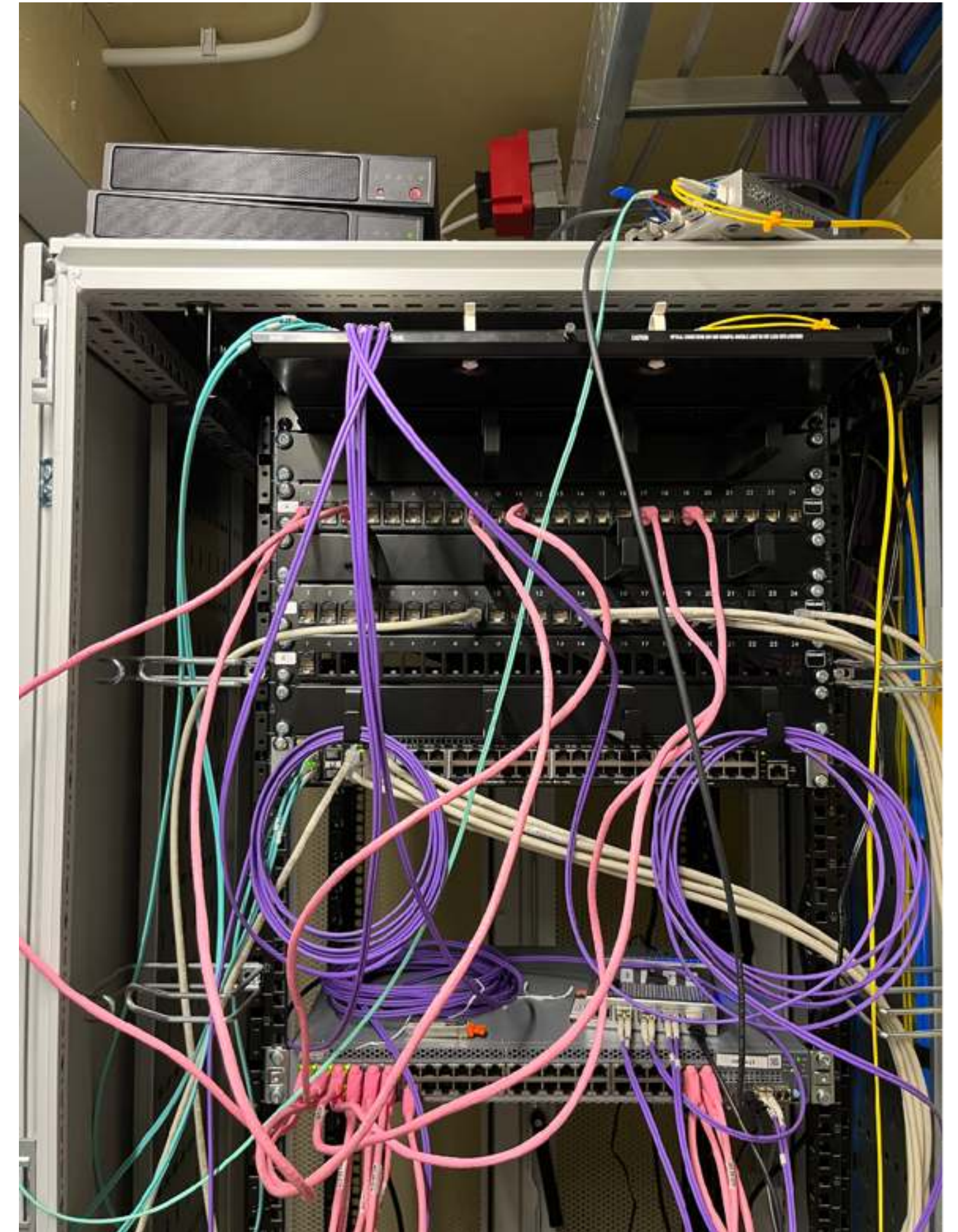


**Amsterdam - Warsaw - Athens
in one hour, 2014, colored**

Physical network



- Two VM hosts running VMware vSphere 8.0
 - SuperMicro SuperServer E300-9D-8CN8TP
 - 25 VMs including routers, firewalls, DHCP servers, DNS resolvers, Wi-Fi controller
- Switches
 - Juniper EX2200 (48×GE PoE+ + 4×10GE SFP+)
 - Zyxel GS-1900-10HP (8×GE PoE+, 2×SFP, VLAN)
 - MikroTik CRS305-1G-4S+IN (4x10GE)
- Access Points
 - Unifi UAP AC (S)HD



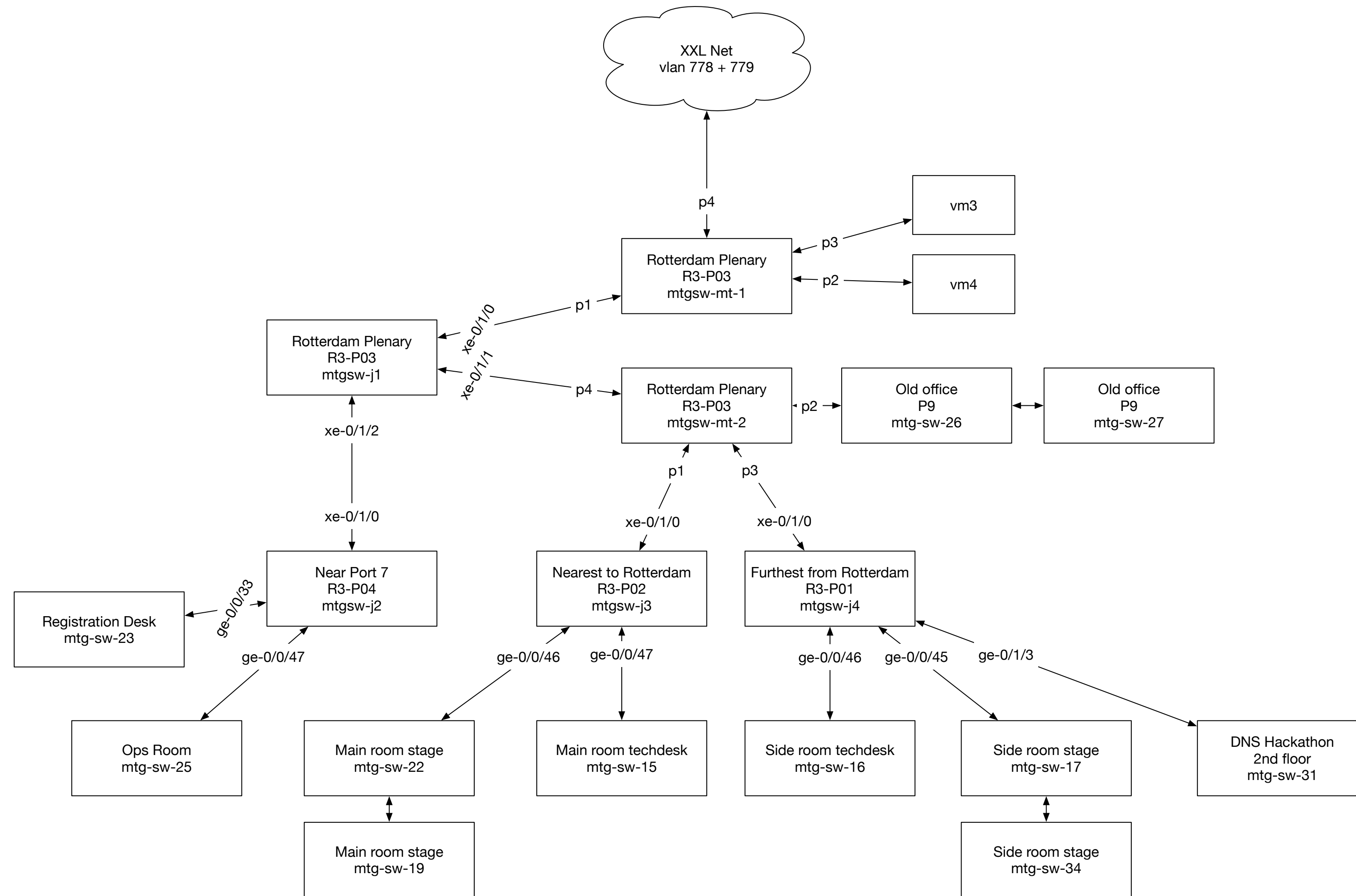
We bring a lot



Testing after covid break



Physical topology (RIPE 86)



The whole network runs Open Source



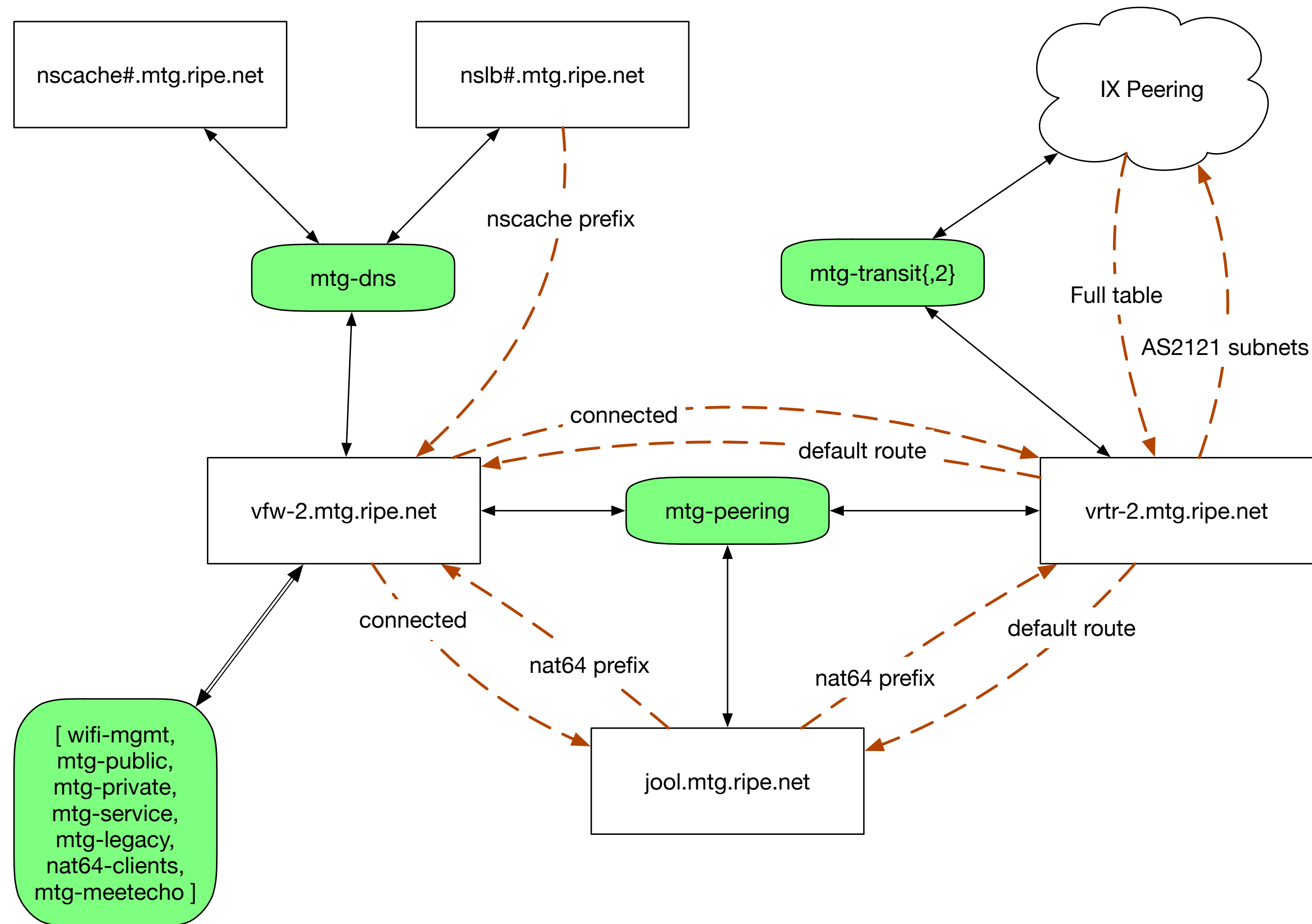
- Edge routers running **BIRD**
- Firewall using **nftables**
- DNS resolver cluster of **Knot Resolver**
- DNS load balancer running **keepalived**
- DHCP servers running **Kea**
- NAT64 using **Jool**
- Statistics collected using **collectd + InfluxDB + Grafana**
- Deployed using **Ansible**

Networks provided



- **Public network**
 - IPv6-mostly dual-stack
- **IPv6-only network**
 - then: NAT64+DNS64
 - now: pure IPv6-only
- **Legacy network**
 - dual-stack without IPv6-mostly signalling
- **Private network**
 - management interfaces
- **Service network**
 - A/V equipment and stenography
- **Wi-Fi management**
 - for APs and Zyxel switches
- **Meetecho network**
 - for video streaming supplier

Logical network topology



Routers on Oracle Linux 9



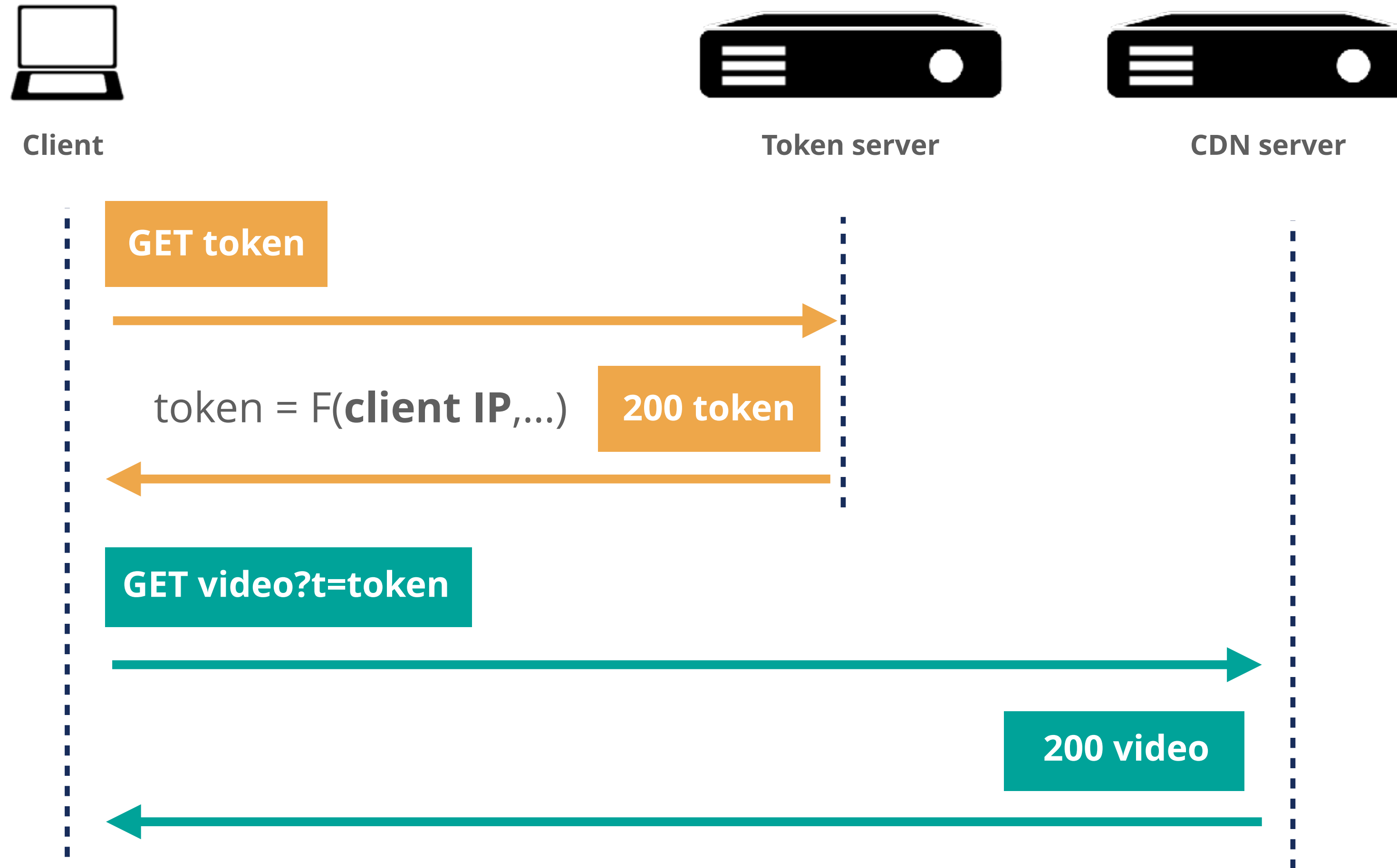
- Two OL9 VMs running BGP with the connectivity provider
- Receive **full BGP feed**, do **RPKI Route Origin Validation**
- Offer default route via OSPF
- **Problem:** high CPU load due to full routing table
 - not a **planned use case** for NetworkManager
 - worked around by **stopping NetworkManager after boot** :)

NAT64



- **Jool** on Debian Linux
- **pool6** = 64:ff9b::/96, **pool4** = 193.0.30.0/24
- BIRD offering **pool6** and **pool4** prefixes via OSPF
- **Problem:** some Video On Demand platforms (iVysilani.cz, NOS.nl,...) **fail to play over NAT64**
 - browser console shows **HTTP 403** error when accessing the CDN
 - so this is apparently an **application layer issue**

How some VOD platforms work



How does Jool allocate IPv4 addresses



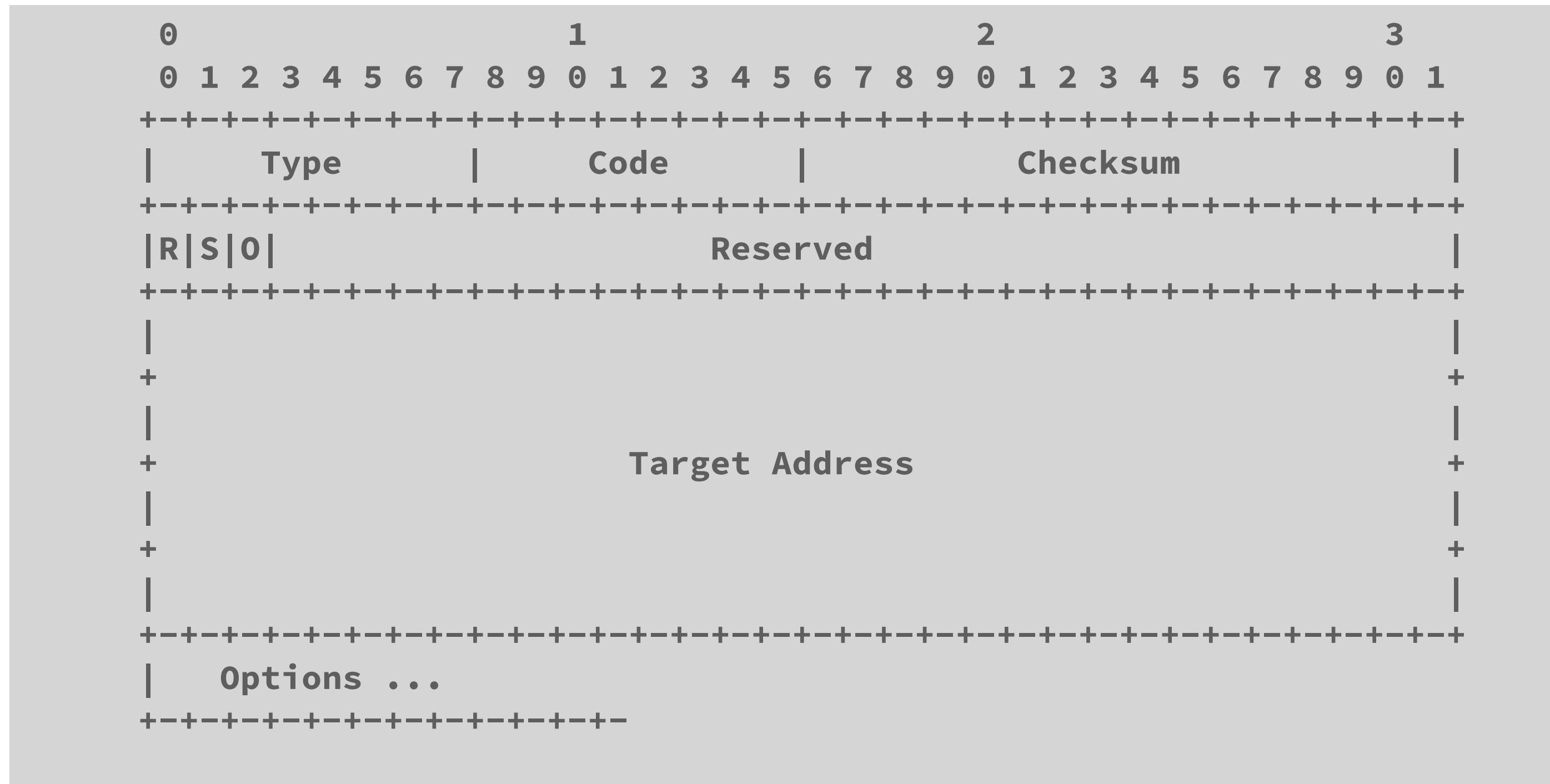
- Address and port tuple is determined by **hashing some parts** of the IPv6 packet
 - by default: source IPv6, **destination IPv6, destination port**
 - hash collisions are resolved by a (slow) iterative process
- Global option **f-args** influence what is hashed
 - setting it to 8 (source IPv6 only) **resolves the issue with the VOD platforms**
 - but all sessions made by one host are causing collisions
- There's a branch of Jool with *Ondřej Caletka's hashing algorithm*
 - uses two hashes, one for choosing IPv4 address, other to choose port
 - no measurement data to prove it is indeed better, not merged

Firewall



- Oracle Linux 9 acting as a default gateway
- BIRD for internal OSPF
- **radvd** from *Git master* to support **PREF64** option for RAs
- **Problems:**
 - Slow throughput due to **extensive firewall logging to the console**
 - IPv6 default gateway **disappears after 6 seconds**, only on macOS
 - Lots of **ARP noise** in public segments

IPv6 neighbor advertisement RFC 4861



R - Router flag. When set, the R-bit indicates that the sender is a router. The R-bit is used by Neighbor Unreachability Detection to detect a **router that changes to a host**.

What makes Linux (un-)set R-flag?



- Turned out this to be the *(only)* feature of **per-interface forwarding sysctl** switch
- IPv6 forwarding is just a **global switch** on Linux
 - yet there are still per interface switches
- NetworkManager used to reset per-interface switch during interface setup
 - **fixed in version 1.44.0**

```
net.ipv6.conf.all.forwarding = 1
net.ipv6.conf.default.forwarding = 1
net.ipv6.conf.lo.forwarding = 1
net.ipv6.conf.eth0.forwarding = 0
net.ipv6.conf.eth1.forwarding = 0
```



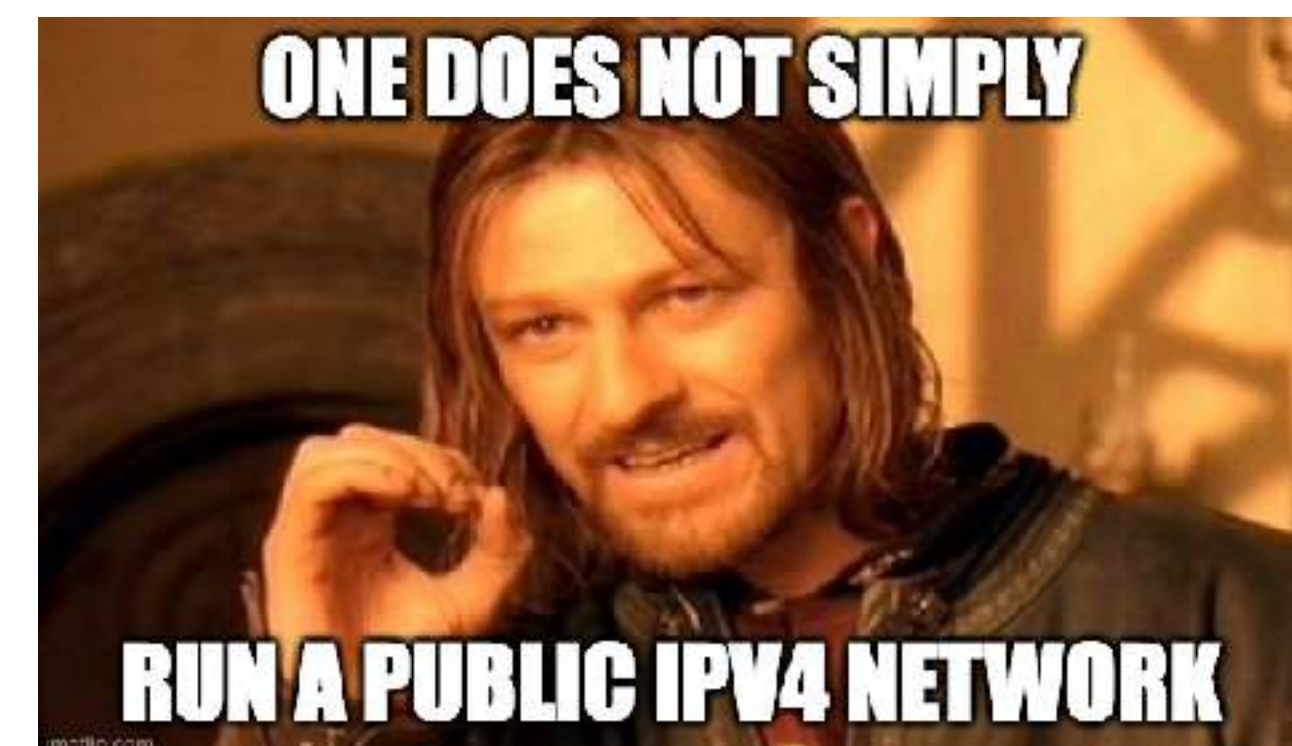

Dealing with ARP noise

- Caused by omnipresent Internet-wide scans
- Kernel-space ARP implementation has **no negative cache**

- **arpd** to rescue!

```
# arpd -k -a2 eth0 eth1 eth2 eth3
```

- part of iproute2
 - implements ARP in userspace
 - has **negative cache**
- **30 times less ARP** messages on an empty network
 - before: **250 pps**, 84 kbps
 - after: **8 pps**, 2.7 kbps



DHCP servers



- Two servers running Kea in **hot-standby** HA mode
- Directly connected to each VLAN, no relay agents
- Stateful DHCPv6 supported but not announced in RAs
- Avoiding addresses **ending with .0 or .255**

```
"pools": [  
  { "pool": "193.0.24.32-193.0.24.254" },  
  { "pool": "193.0.25.1-193.0.25.254" },  
  { "pool": "193.0.26.1-193.0.26.254" },  
  { "pool": "193.0.27.1-193.0.27.254" }  
],
```


DNS Resolver Cluster



- Four worker nodes, two load balancers
- All in one dedicated subnet
- Service address **outside the subnet** on dummy interfaces
- Keepalived in **Direct Routing** mode
 - incoming traffic is bounced to a worker node
 - outgoing traffic goes directly
- Load balancers announce service addresses using OSPF
- Support for **DNS64, DoT, DoH, DDR**

Unexpected issue with DDR



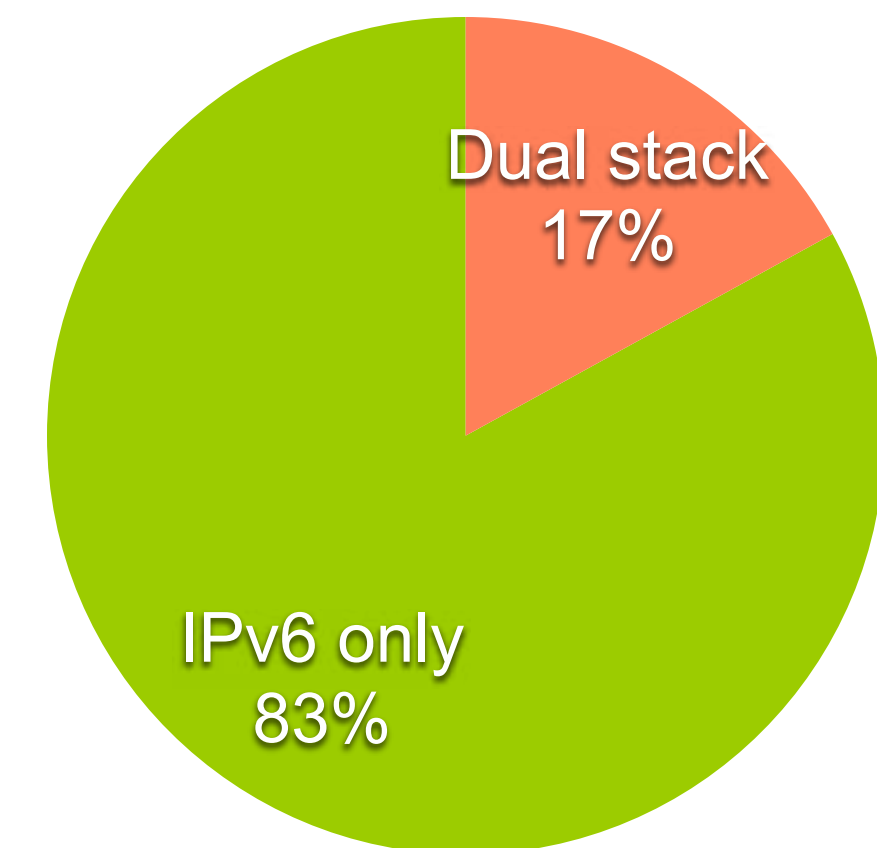
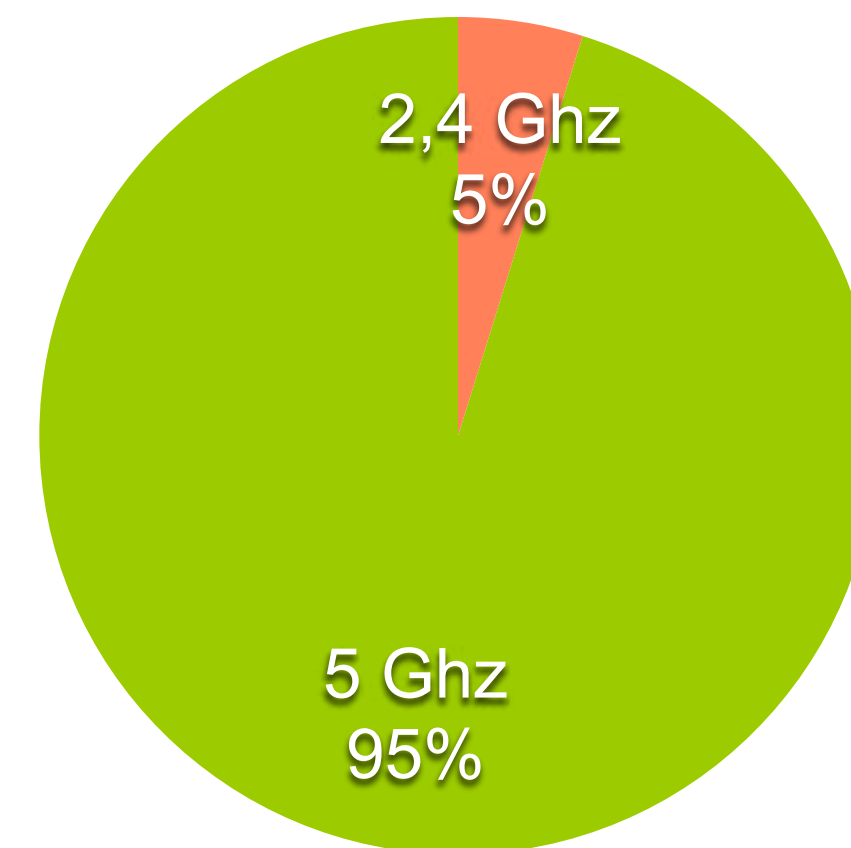
- Only **github.com** and **duckduckgo.com**
- Only in **Safari** and on **iOS**
- Only in **IPv6-only network**
- Only **intermittently**
- Only with **our DNS resolvers**
- Only with **DDR triggered DoH**



Wi-Fi



- Unifi controller running on Debian Linux
- **Manual channel configuration, mostly 5 GHz only**
- Legacy eSSID names **changing every meeting**
- *Multicast and Broadcast control kills IPv6 NDP*
 - you have to **white-list MAC addresses** of all wired IPv6 hosts
 - unintended **RA-guard-like function**



RIPE 87 Wi-Fi stats

Future



- Redundant firewall with VRRP
 - without or with state synchronisation
- Redundant NAT64
 - probably without state synchronisation
- Figure out better solution than stopping NetworkManager
- Figure out what's wrong regarding DDR on Apple
- Wi-Fi 6E or Wi-Fi 7 - depends on the budget



Questions



Ondrej@Caletka.cz
<https://ondrej.caletka.cz>