RIPE NCC
RIPE NETWORK COORDINATION CENTRE

IPv6-mostly na OpenWRT

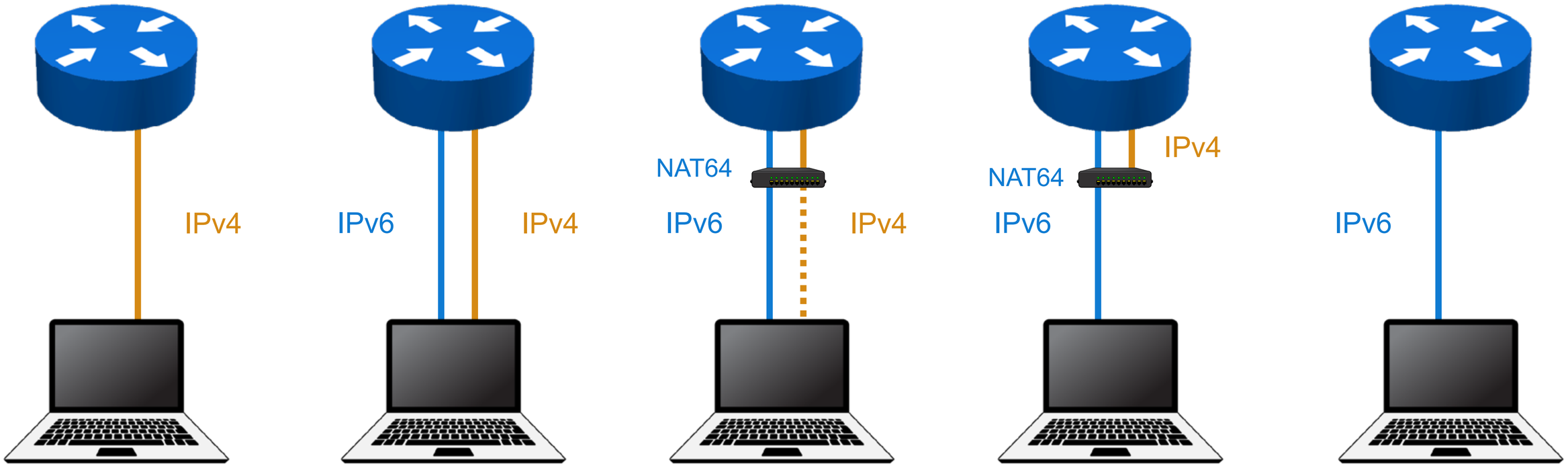
Jak doma zprovoznit NAT64 / PREF64 / DNS64 / DHCP108

Ondřej Caletka | Installfest 2024 | 16. března 2024

Co a proč?

- Máte doma *(nativní)* **IPv4 a IPv6**
- Všechno je **dual-stack**
- Rádi byste se *postupně* zbavili IPv4
- Chcete vědět, **co stále bez IPv4 nefunguje**, abyste to mohli opravit
  - **spoiler alert**: moc nefunkčních věcí neuvidíte

Fáze přechodu na IPv6



Požadavky

- Dual-stack upstream s **delegovaným IPv6 prefixem**
- Router na kterém běží OpenWRT, **ideálně v23.05.2**
- Hardwarové tipy:

Routery Turris používají TurrisOS založený na o něco starším OpenWRT

Routery GL-iNet jsou dodávány s firmwarem založeným na OpenWRT, který se dá snadno přepsat upstream OpenWRT

Co si dnes předvedeme

- Vytvoříme extra síť s **IPv6-only**
- Zprovozníme NAT64 pomocí překladače **Jool**
- Nastavíme nativní podporu **PREF64** v OpenWRT
- Nastavíme DHCP server, aby nabízel “**IPv6-only preferred**”
- Nastavíme **DNS64** pomocí Public DNS/Unbound/Knot Resolver
- Automatizujeme pomocí **Ansible**

Síť IPv6-only

Síť IPv6-only

- Výchozí síť lan ponecháme jako **dual-stack**
- Vytvoříme novou síť lan6 bez konfigurace IPv4
- Alokujeme podsíť velikosti /60 pro dané rozhraní
  - první /64 se použije pro přímo připojená zařízení
  - zbytek bude k dispozici **kaskádně zapojeným routerům** pomocí DHCP-PD

```
config device
    option type 'bridge'
    option name 'br-lan6'
    option bridge_empty '1'
    list ports 'lan2'
```

```
config interface 'lan6'
    option proto 'static'
    option device 'br-lan6'
    option ip6assign '60'
    option ip6hint '60'
```

```
config dhcp 'lan6'
    option interface 'lan6'
    option ignore '1'
    option ra 'server'
    option dhcpv6 'server'
```

```
config zone
    option name ‘lan'
    …
    list network 'lan6'
```

Co máme teď

- Síť dual-stack: nic zvláštního
- Síť IPv6-only: zcela bez podpory IPv4
  - ideální budoucí Internet
  - spousta věcí už **běžně funguje**
  - ale také spousta věcí **vůbec nefunguje**

IPv6-only

PSK: tutorial

IPv6-mostly

PSK: tutorial

NAT64

Předstírání, že je vše
dostupné pomocí IPv6

NAT64

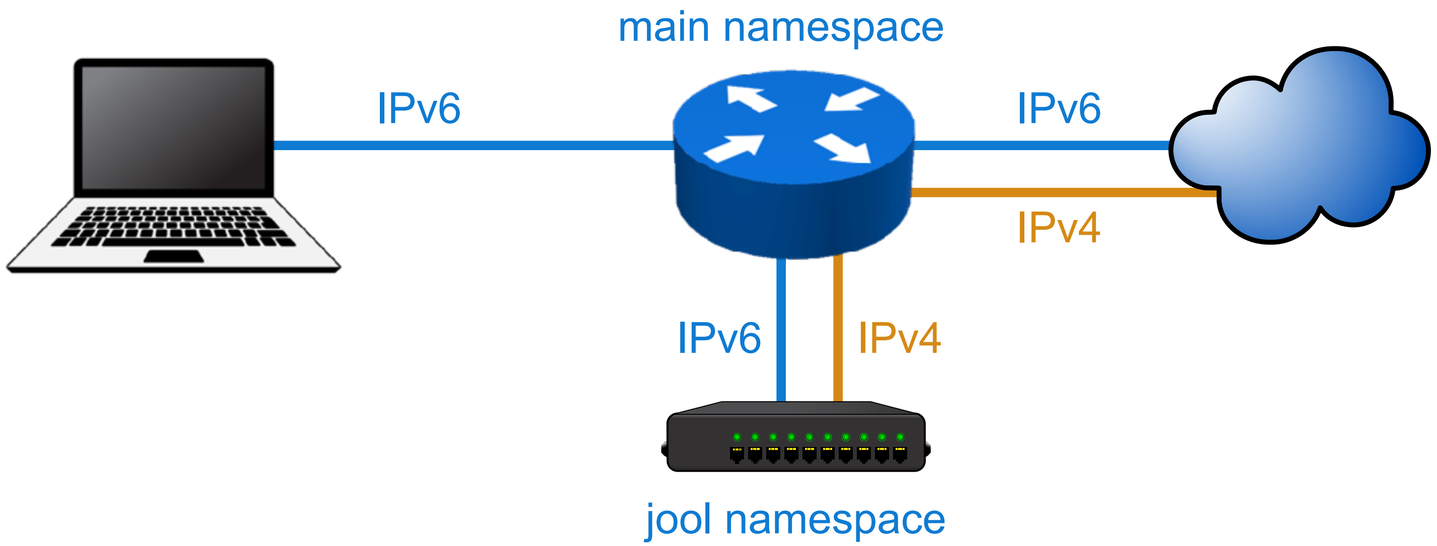
- Překladač paketů mezi IPv4 a IPv6
- Bezestavový nebo **stavový**
  - bezestavový se hodí hlavně pro poskytování IPv6 služeb IPv4-only klientům
  - stavový se hodí k připojení **IPv6-only klientů** k **IPv4 službám**
- Používá **Well-Known** nebo **Network-Specific Prefix**
  - **Privátní IPv4 adresy** nejsou povoleny s Well-Known Prefix

Jool

- Implementace NAT64 v linuxovém jádře
- Dostupná v OpenWRT
- Není integrovaná do konfiguračního systému OpenWRT
- *Krade* pakety v **PREROUTING**, injektuje přeložené pakety do **POSTROUTING**
  - Problematická kombinace s firewallem
  - Překlad není dostupný pro provoz samotného překladače

Jool v síťovém namespace

- Namespaces propojeny pomocí páru veth rozhraní
- Odpadají problémy s firewallem/lokálním provozem



Pojďme to nastavit

Potřebné balíčky

- **kmod-veth**
- **ip-full**
- **kmod-jool-netfilter**
- **jool-tools-netfilter**

```
#!/bin/sh
ip link add jool type veth peer openwrt
ip netns add jool
ip link set dev openwrt netns jool

ip netns exec jool sh <<EOF
        sysctl -w net.ipv4.conf.all.forwarding=1
        sysctl -w net.ipv6.conf.all.forwarding=1
        sysctl -w net.ipv6.conf.openwrt.accept_ra=2
        sysctl -w net.ipv4.ip_local_port_range="32768 32999"
        ip link set dev lo up
        ip link set dev openwrt up
        ip addr add dev openwrt 192.168.164.2/24
        ip addr add dev openwrt fe80::64
        ip route add default via 192.168.164.1
        modprobe jool
        jool instance add --netfilter --pool6 64:ff9b::/96
        jool global update lowest-ipv6-mtu 1500
        jool pool4 add 192.168.164.2 33000-65535 --tcp
        jool pool4 add 192.168.164.2 33000-65535 --udp
        jool pool4 add 192.168.164.2 33000-65535 --icmp
EOF
```

Strana OpenWRT (hlavní NS)

- Použijeme IPv4 podsíť 192.168.164.1/24
- Použijeme jednu IPv6 /64 s podporou SLAAC
- NAT64 prefix posíláme na LL adresu fe80::64
- Rozhraní `jool` přidáme do zóny `lan` firewallu

```
config dhcp 'jool'
    option interface 'jool'
    option ra 'server'
    option ra_default '2'
    option ignore '1'
```

```
config interface 'jool'
   option device 'jool'
   option proto 'static'
   option ip6assign '64'
   option ip6hint '64'
   list ipaddr '192.168.164.1/24'

config route6 'nat64'
   option interface 'jool'
   option target '64:ff9b::/96'
   option gateway 'fe80::64'
```

Testujeme NAT64

- `ping/traceroute 64:ff9b::<oblíbená IPv4 adresa>`
- Ujistěte se, že to funguje i z připojeného zařízení
  - jinak může být problém ve směrování/firewallu

PREF64

Ať všichni vědí, že
máme NAT64

PREF64

- Volba **Ohlášení Směrovače** (RA) obsahující prefix používaný **NAT64** překladačem
- Počítače tedy mohou posílat IPv4 provoz tam do tohoto prefixu
  - Obvykle pomocí **CLAT** - uživatelského překladače mezi IPv4 and IPv6
- PREF64 je **nová funkce** OpenWRT v23.05.0
  - backportováno do TurrisOS 6

```
config dhcp 'lan6'
    option interface 'lan6'
    …
    option ra_pref64 '64:ff9b::/96'
```

Co máme teď

- Síť dual-stack: nic zvláštního
- Síť IPv6-only:
  - funguje **bez problémů** s OS Android (IPv4 pomocí CLAT)
  - funguje **bez problémů** s iOS/macOS (IPv4 pomocí CLAT/nativního překladu)
  - funguje **omezeně** s jinými OS (bez CLAT, bez podpory PREF64 = IPv4 nedostupné)

IPv6-only

PSK: tutorial

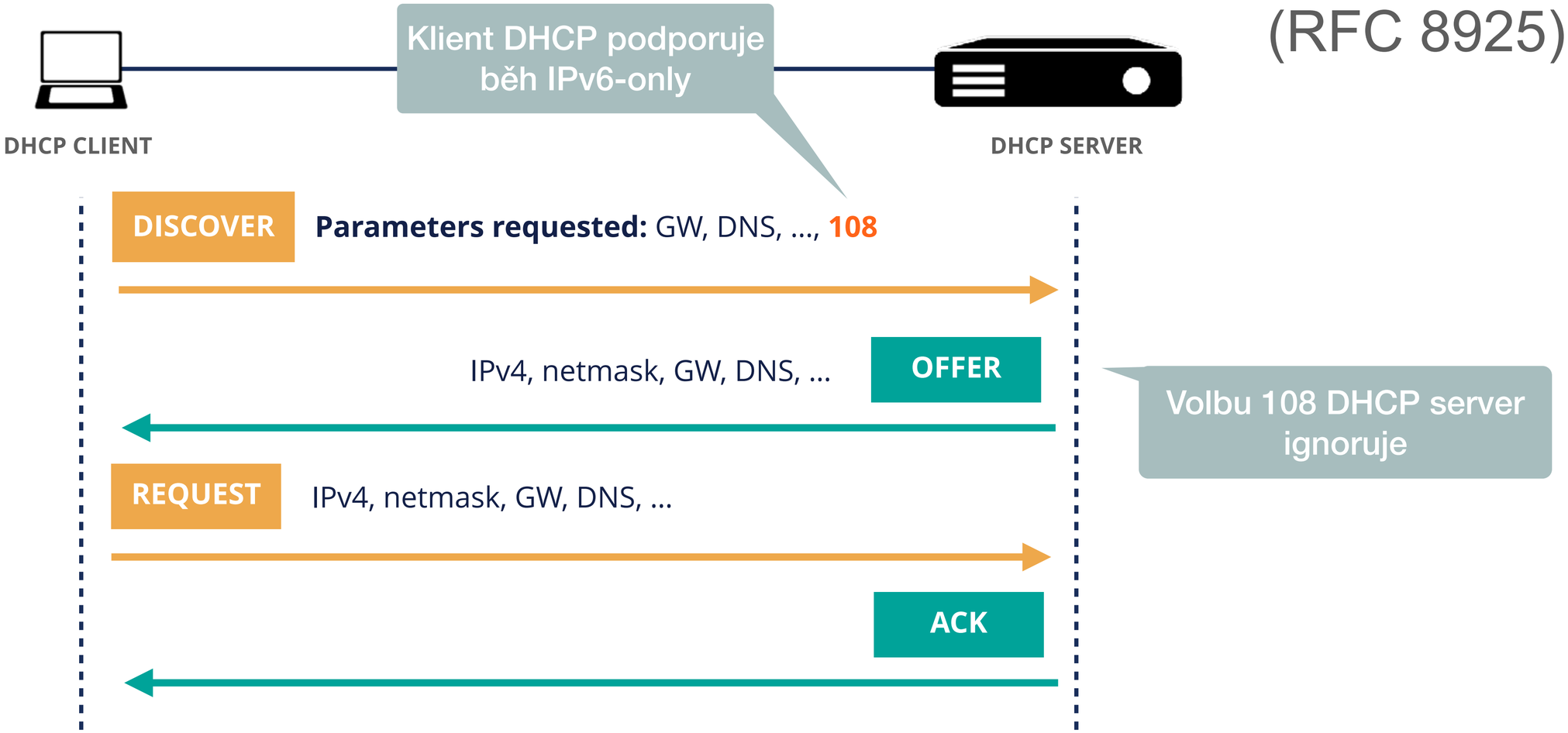
IPv6-mostly

PSK: tutorial

IPv6-only-preferred

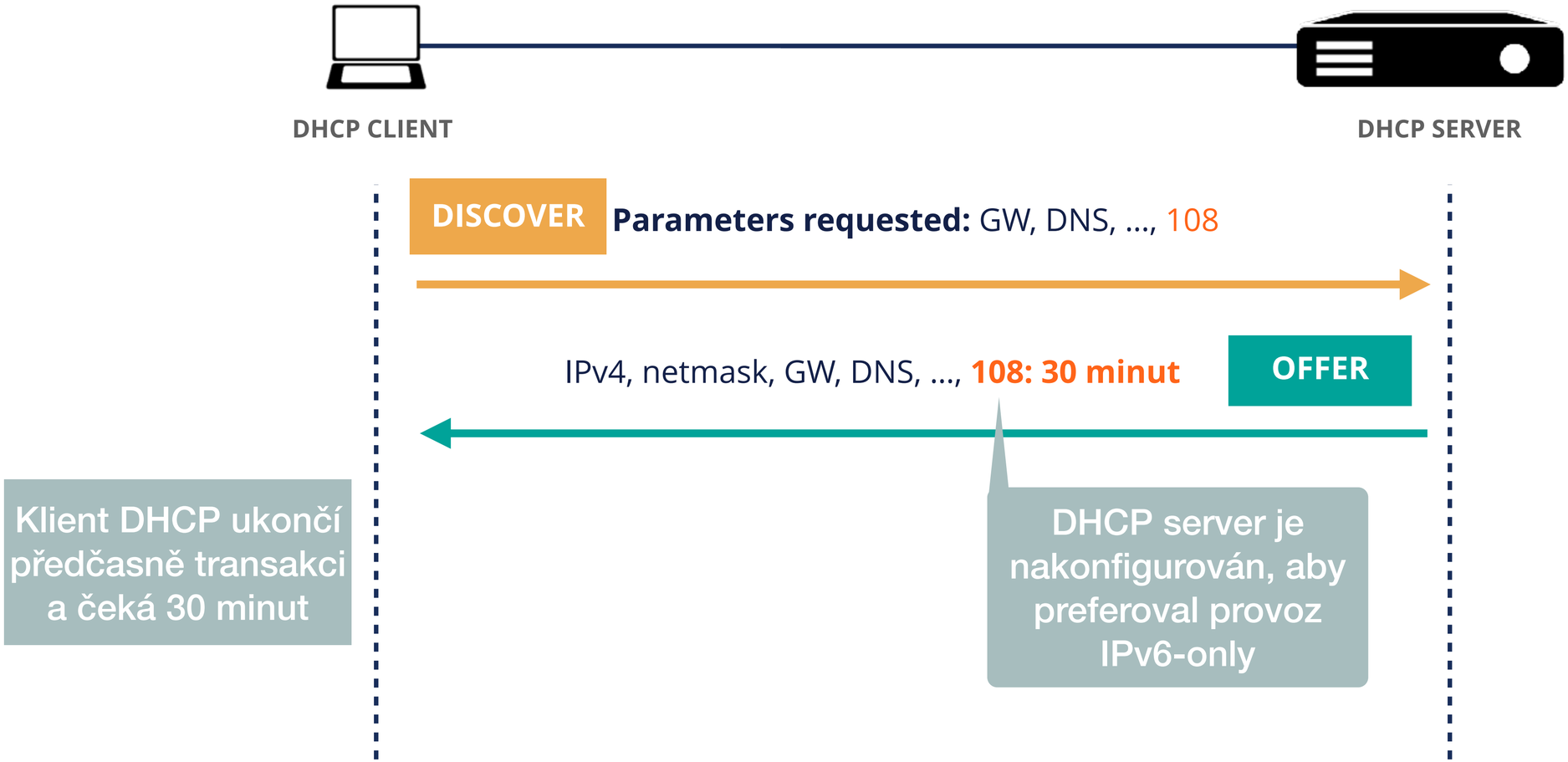
Volba DHCP pro vypnutí IPv4

Volba DHCP: IPv6-only Preferred



Použití DHCP k vypnutí IPv4

(RFC 8925)



Konfigurace volby DHCP č. 108

- Není třeba speciálního chování DHCP serveru
- Jen si musíme sami zakódovat hodnotu
  - 30 minut = 1800 sekund = 0x708 sekund

```
config dhcp 'lan'
    option interface 'lan'
    list dhcp_option '108,0:0:7:8'
    …
```

Co máme teď

- Síť IPv6-only: beze změny
- Síť ~~dual-stack~~ IPv6-mostly:
  - **beze změny** pro Windows, Linux
  - **stejná jako IPv6-only** pro Android, iOS a macOS

IPv6-only

PSK: tutorial

IPv6-mostly

PSK: tutorial

DNS64

Falšování DNS
s dobrými úmysly

DNS64

- Jednoduchý trik jak přinutit klienty k použití NAT64
- **Neovlivňuje** nativní IPv6
- Dotazy na IPv4-only zdroje dostanou **falešnou IPv6 adresu** mířící do prostoru NAT64 překladače
- Díky tomu si klient myslí, že **každé doménové jméno má IPv6**
- Funguje velmi dobře, až na *drobnosti*:
  - IPv4 literály
  - validace DNSSEC
  - zastaralá IPv4-only socket API

Potřebujeme opravdu DNS64?

- DNS64 bude nejspíš v budoucnu **nahrazeno PREF64** a překladem uvnitř klienta
- Androidu stačí **NAT64/PREF64**
- To samé platí pro nejnovější macOS/iOS
- DNS64 způsobí preferenci NAT64 před nativním IPv4
  - dobré pro sítě IPv6-only
  - **ne až tak dobré** pro sítě IPv6-mostly, kde je k dispozici i nativní IPv4 pro starší zařízení

Jednoduché: Public DNS64

- Google Public DNS64
- Cloudflare Public DNS64
- Pouze pokud používáte **Well-Known Prefix** pro NAT64

```
config dhcp 'lan'
    option interface 'lan'
    list dns '2001:4860:4860::64'
    list dns '2606:4700:4700::64'
    …
```

Jednoduché řešení na TurrisOS

- TurrisOS používá jako výchozí **Knot DNS Resolver**
- Knot DNS Resolver má dobrou podporu pro DNS64

```
modules = { 'dns64', 'view' }

-- Custom prefix example
-- dns64.config({ prefix = '64:ff9b:face:b00c::' })

-- Disable dns64 for IPv4 clients
view:addr('0.0.0.0/0', policy.all(policy.FLAGS('DNS64_DISABLE')))

-- Reenable it for a specific prefix:
view:addr(‘127.0.0.0/8’, policy.all(policy.FLAGS(nil, 'DNS64_DISABLE')))
```

Náhrada dnsmasq za Unbound

- Musíme vypnout funkci DNS forwarderu, ale zachovat funkci DHCP serveru
- Vypnutí DNS vypne nabízení DNS ve volbách DHCP
- Potenciální problém s přeložitelností lokálních DHCP jmen lokálním DNS resolverem

```
config dnsmasq
    option port '0'
    …

config dhcp 'lan'
    list dhcp_option '6,0.0.0.0'
    …
```

```
config unbound ‘ub_main'
    …
    option dns64 '1'
    option dns64_prefix '64:ff9b::/96'
    …
    option validator '1'
    list iface_lan 'lan'
    list iface_lan 'lan6'
```

Unbound Recursive DNS server with UCI

Co máme teď

- Síť IPv6-only:
  - pracuje **normálně** na Android, iOS a macOS
  - na ostatních OS pracuje **s drobnými problémy**
- Síť ~~dual-stack~~ IPv6-mostly:
  - funguje přesně stejně jako IPv6-only pro Android, iOS a macOS
  - část provozu IPv4 prochází přes **DNS64** namísto nativního IPv4 pro Windows, Linux

IPv6-only

PSK: tutorial

IPv6-mostly

PSK: tutorial

Použití Ansible

Automatizace nastavení OpenWRT

Problémy

- Python není ve výchozím stavu k dispozici na OpenWRT
- Chybějící podpora pro konfigurační systém UCI

Oba problémy řeší role `gekmihesg.openwrt`

Moje kolekce rolí

- openwrt-lan6
- openwrt-jool
- openwrt-pref64
- openwrt-dhcp108
- openwrt-unbound

Používejte a sdílejte:

https://github.com/oskar456/ansible-openwrt-ipv6-mostly

Otázky

Ondrej@Caletka.cz
https://Ondřej.Caletka.cz