Úvod do IPv6

Pavel Satrapa, TU v Liberci

Pavel.Satrapa@tul.cz

Historie

Motivace

- počátek 90. let – zjevný nedostatek IP adres

ipv4.potaroo.net

Nový protokol

- upgrade IPv4 nebo nový protokol?
- cca 10 let – dost času na zásadní změnu
- **raději nový protokol, který umožní zapracovat nové vlastnosti**
- 1. specifikace: RFC 1883, vydáno v prosinci 1995
- upravená specifikace: RFC 2460, vydáno v prosinci 1998, stále platná, připravuje se náhrada
- řada doprovodných RFC

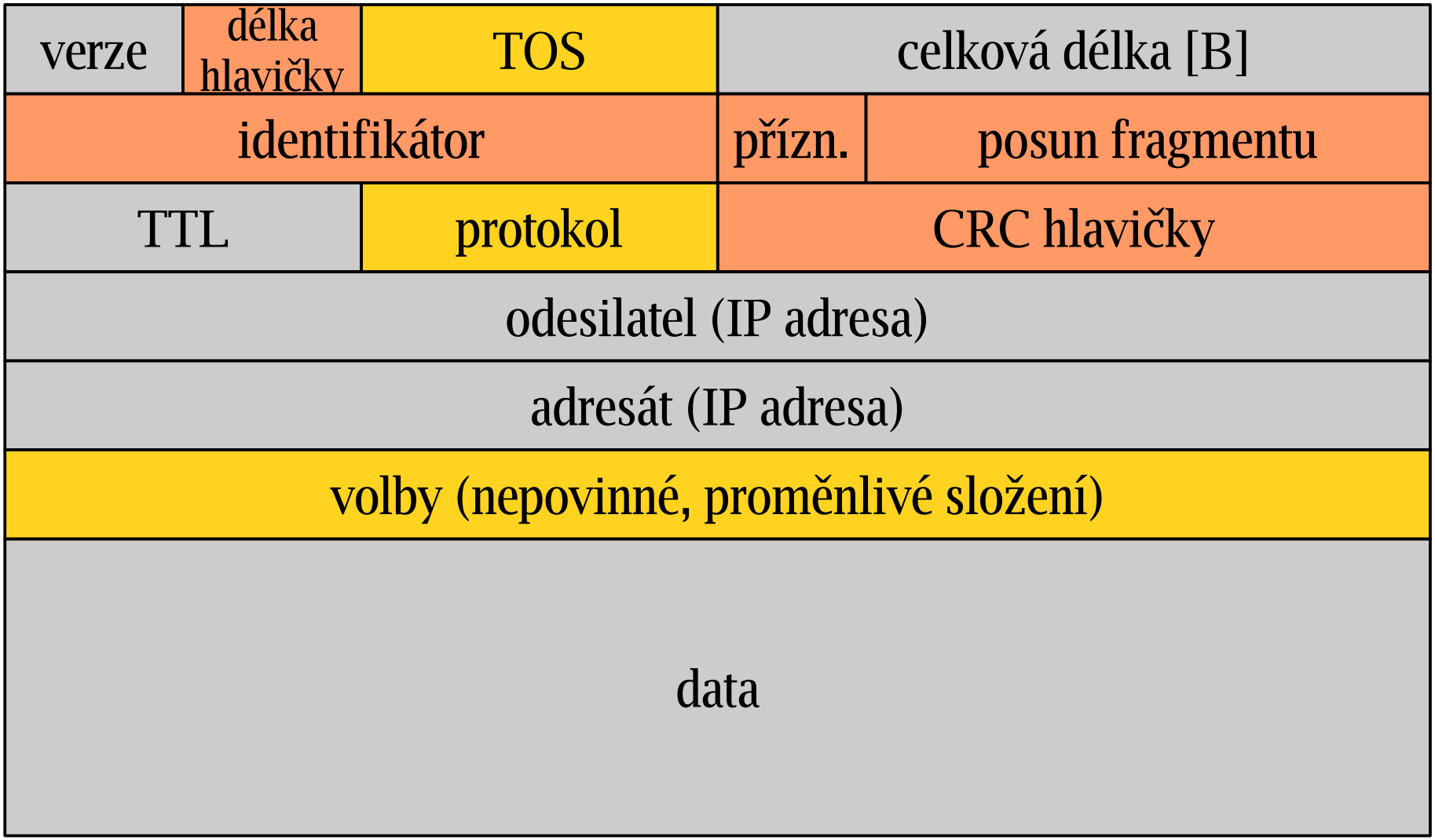
Cíle

- dostatek adres (pokud možno navždy)
- hierarchické směrování a adresace
- zvýšení bezpečnosti (šifrování a autentizace v IP)
- služby se zajištěnou kvalitou
- vysokorychlostní směrování
- podpora mobilních zařízení
- automatická konfigurace
- přechodové mechanismy

Datagram

IPv6 datagram

IPv4 datagram



Položky hlavičky

- **verze:** identifikuje verzi protokolu (=6)
- **třída provozu:** pro služby s definovanou kvalitou
- **značka toku:** identifikuje tok (proud souvisejících datagramů), např. pro firewally
- **délka dat:** počet bajtů za hlavičkou
- **další hlavička:** řetězení hlaviček (viz dále)
- **max. skoků:** analogie TTL, omezuje dosah

Řetězení hlaviček

- délka hlavičky je konstantní (40 B) – urychluje zpracování
- případné **volitelné hlavičky** se připojují za ni
- ***další hlavička*** určuje typ hlavičky, která následuje, nebo protokol vyšší vrstvy
- každá rozšiřující hlavička obsahuje položku *další hlavička*
- pořadí hlaviček urychluje zpracování

Příklady řetězení

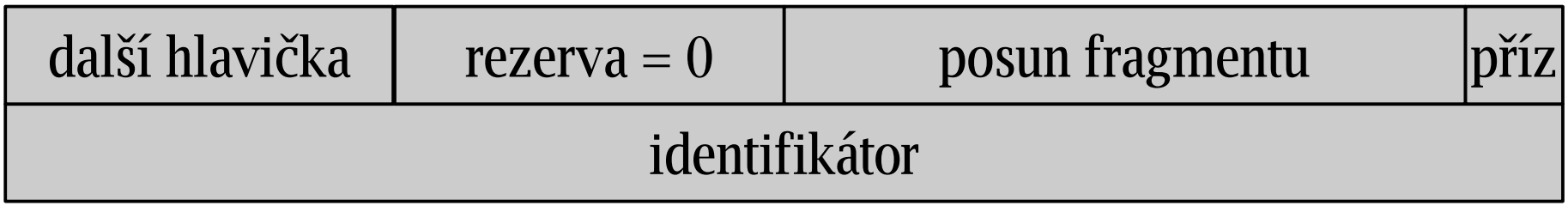
- **jen standardní hlavička:**
  - další hlavička = 6 (TCP)
- **s hlavičkou *Fragmentace*:**
  - další hlavička = 44 (Fragmentace)
  - Fragmentace/další hlavička = 6 (TCP)
- **s hlavičkami *Fragmentace* a *ESP*:**
  - další hlavička = 44 (Fragmentace)
  - Fragmentace/další hlavička = 50 (ESP)
  - ESP/další hlavička = 6 (TCP)

Typy rozšiřujících hlaviček

0	volby pro všechny (hop-by-hop options)
43	směrování (routing)
44	fragmentace
50	šifrování (ESP, Encapsulating Security Payload)
51	autentizace (AH, Authentication Header)
60	volby pro cíl (destination options)
135	mobilita
139	HIP (host identity protocol)
140	shim6
253, 254	experimentální

Fragmentace (1)

- nepopulární, v IPv6 omezována
- IPv6 požaduje MTU alespoň 1280 B
- **fragmentovat smí jen odesilatel**
  - nestačí-li MTU po cestě, datagram se zahodí a pošle se ICMPv6 zpráva odesilateli
- přidá hlavičku *Fragmentace*



Fragmentace (2)

- **identifikátor**: které fragmenty patří k sobě
- **posun fragmentu**: na kterém bajtu původního datagramu začíná fragment
- **příznak M**: následuje další fragment
- skládá příjemce
- důrazně se doporučuje používat **objevování MTU cesty** (path MTU discovery, RFC 1981)

Problémy rozšiřujících hlaviček

- nový prvek – nějakou dobu si sedalo
- **zneužívání**
  - dlouhé řetězce hlaviček (zpomalují zpracování, dojde-li k fragmentaci, firewall nemůže posoudit)
  - obcházení detekčních mechanismů
- RFC 7112: všechny hlavičky se musí vejít do prvního fragmentu
- RFC 7872: měření ukázala, že dramaticky snižují šanci na průchod sítí

Toky (1)

- v IPv4 je datový tok identifikován pěticí
  - zdrojová IPv4 adresa
  - cílová IPv4 adresa
  - transportní protokol
  - zdrojový port
  - cílový port
- problém: potřebuje informace z transportní vrstvy
  - v IPv6 znamená projít všemi hlavičkami – zdržuje

Toky (2)

- koncept toku – proud datagramů, které spolu „nějak souvisí“ – často např. jedno transportní spojení
- tok **identifikován trojicí**:
  - zdrojová IPv6 adresa
  - cílová IPv6 adresa
  - značka toku
- všechny obsaženy v základní hlavičce
- lze využít např. ve stavových firewallech

Adresy

Základní parametry

- **délka 128 bitů** (16 bajtů)
- **zápis v šestnáctkové soustavě, čtveřice číslic odděleny dvojtečkou**
- 2001:0718:1c01:0005:020b:dbff:fea1:d52c
- úvodní nuly ve čtveřici lze vynechat
- jednu skupinu nulových čtveřic lze vynechat a nahradit dvěma dvojtečkami (např. smyčka je ::1)
- prefixy v obvyklém tvaru 2001:718::/32

Kanonický zápis

- RFC 5952
- šestnáctkové číslice **malými písmeny**
- **vynechání počátečních nul** ve skupině **povinné**
- **„::“** musí být použito tak, aby mělo **co největší efekt** – pohltit všechny sousedící nulové skupiny a použít pro jejich nejdelší sekvenci (při několika stejně dlouhých se použije pro první)
- SW by měl podporovat všechny platné zápisy, ale výstup v kanonickém tvaru

IPv6 adresa v URL

- uzavřít do [...]
- např. http://[2001:718:1c01:16::aa]/
- RFC 3986

Typy adres

- **individuální (unicast)**
  - určují jedno rozhraní
- **skupinové (multicast)**
  - určují skupinu rozhraní, data se doručují všem
- **výběrové (anycast)**
  - určují skupinu rozhraní, data se doručují nejbližšímu členovi
- chybí broadcast (speciální případ skupinových)

Globální individuální adresy

- „normální“ adresy
- první tři bity 001 (binárně)
- na začátku je **globální směrovací prefix**
- pak **identifikátor podsítě**
- 64 b **identifikátor rozhraní**

Globální směrovací prefix

- dříve adresa sítě
- přiděluje poskytovatel Internetu (LIR), aktuálně:
  - /12 přidělí IANA pro RIR (RIPE NCC má 2a00::/12)
  - /29 přidělí RIR pro LIR
  - /48 až /64 přidělí LIR pro zákazníka
- **standardní délka 48 bitů**
  - 16 b identifikátor podsítě – 65 536 podsítí
- **pro malé sítě může být delší**: 56 b (256 podsítí) nebo 64 b (bez podsítí)

Identifikátor rozhraní (1)

- původně: požadováno **modifikované EUI-64**
- **odvozeno z MAC adresy**
  - doprostřed vložit fffe
  - invertovat druhý bit
- z MAC adresy 00:37:d7:b9:a1:d0 vznikne 237:d7ff:feb9:a1d0
- nezapamatovatelné
- stroj lze sledovat (v různých sítích má stále stejný identifikátor rozhraní)

Identifikátor rozhraní (2)

- běžně se používaly ručně přidělené identifikátory – např. Facebook: 2a03:2880:f01b:b01:face:b00c:0:1
- RFC 7136
  - zrušilo požadavek na používání modifikovaného EUI-64, nově: pokud identifikátor rozhraní vychází z linkové adresy, musí se použít modifikované EUI-64
  - zrušilo speciální významy jakýchkoli bitů v identifikátoru rozhraní (příznaky globální/lokální, skupinová/individuální)

Náhodné identifikátory (1)

- RFC 4941: **identifikátory rozhraní zachovávající soukromí (privacy extensions)**
- identifikátor rozhraní se generuje náhodně
- krátká životnost (hodiny), mění se
- používají se pro odchozí spojení
- stálý identifikátor (mEUI-64) pro příchozí spojení
- noční můra pro správce

Stálé náhodné identifikátory

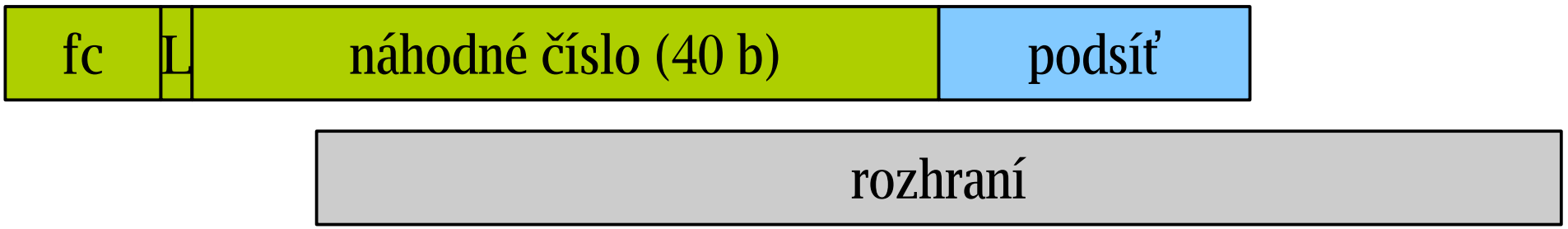
- RFC 7217
- identifikátor je náhodný (bez vazby na MAC)
  - hash z několika hodnot včetně prefixu
- v každé (pod)síti jiný, ve stejné (pod)síti zůstává konstantní
- kompromis mezi ochranou soukromí a spravovatelností
- **místo EUI-64 adres pro bezstavovou autokonfiguraci – doporučeno (RFC 8064)**

Linkové lokální adresy

- prefix fe80::/10
- **jednoznačné pouze v rámci jedné linky** (Ethernet)
- každé rozhraní s aktivním IPv6 má lokální linkovou adresu – vytvoří si sám připojením identifikátoru rozhraní k prefixu fe80::/10
- používají se např. ve směrovací tabulce
- není jednoznačná – připojuje se rozhraní fe80::020b:dbff:fea1:d52c%1

Unikátní lokální adresy (1)

- analogie IPv4 adres podle RFC 1918
- **nevyžadují přidělení, vytvoří si správce sám**
- nesměrují se v Internetu, jen pro lokální použití
- **prefix fc::/7**
- příznak L: 1=generovány lokálně, 0=jinak



Unikátní lokální adresy (2)

- náhodné číslo v prefixu snižuje pravděpodobnost, že stejný ULA prefix použije více sítí
  - pokud by unikly, nenaruší směrování jinde
  - snadnější budoucí propojení sítí
- vzhledem k dostatku IPv6 adres je není nutno používat jako berličku v kombinaci s NATem
- **raději se jim vyhněte**

Adresy obsahující IPv4

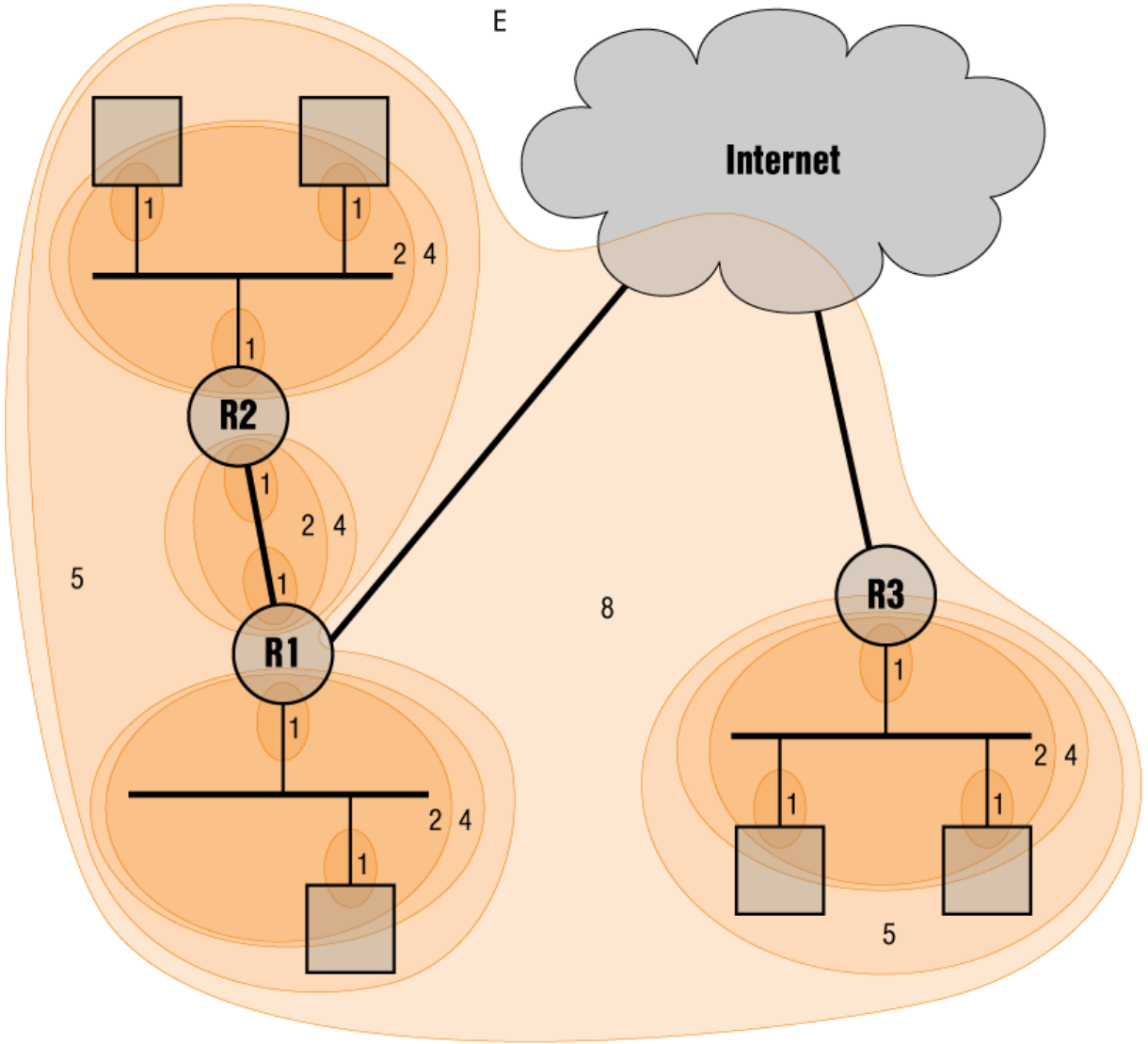
- RFC 6052
- obecná struktura:

- *prefix* definuje správce nebo lze použít univerzální 64:ff9b::/96
- *přípona* umožňuje rozlišit části, nedoporučuje se
- IPv4 adresu lze psát normálně nebo převést do šestnáctkové soustavy

Skupinové adresy

- prefix ff::/8
- volby: 0RPT
  - T = je dočasná (1) nebo dobře známá/trvalá (0)?
  - P = vychází ze síťového prefixu? (musí být T=1)
  - R = obsahuje Rendezvous Point? (musí být P=1, T=1)

Dosah adres



1 = rozhraní
2 = linka
4 = správa
5 = místo
8 = organizace
E = globální

Výběrové adresy

- nemají vyhrazenou část adresního prostoru
- od individuálních se liší jen konfigurací
- typicky vyžadují samostatnou položku ve směrovacích tabulkách – při velkém dosahu špatně škáluje
- používají se spíše vzácně (např. kořenové DNS servery)
- provozem výběrových adres se zabývá RFC 4786

Povinné adresy uzlu

- **IPv6 uzel musí mít nastaveny tyto adresy:**
- lokální linková pro každé rozhraní
- všechny přidělené individuální a výběrové
- lokální smyčka (::1)
- skupinové pro všechny uzly
- skupinová pro vyzývaný uzel pro všechny přidělené individuální a výběrové
- skupinové, jejichž je členem

Povinné adresy směrovače

- **směrovač musí mít všechny adresy jako uzel a navíc:**
- výběrová pro směrovače v podsíti
- skupinové pro všechny směrovače

Výběr adresy (1)

- mnoho adres k dispozici – kandidátské adresy
  - cíl: DNS může vrátit několik adresních záznamů
  - zdroj: stroj má přidělenu řadu adres (lokální linkové, globální, ULA,...)
- nutno vybrat vhodné adresy pro danou komunikaci
- definuje RFC 6724

Výběr adresy (2)

- **postup:**
  - projde všechny cílové adresy (získané z DNS) a ke každé určí nejvhodnější zdrojovou adresu
  - projde dvojice zdrojová–cílová z předchozího kroku a seřadí je podle vhodnosti
  - použije nejvhodnější dvojici
- vhodnost adres lze ovlivňovat **tabulkou politik (policy table)** – přiřazuje **značku (label)** a **prioritu (precedence)**

V čem je IPv6 jiné

- rozhraní má více adres
- náhodné krátkodobé adresy
- adresy s omezeným dosahem
  - lokální linkové
  - ULA

Směrování

Směrovací tabulka

- používají se lokální linkové adresy sousedů
- příklad:

prefix	*next*	*int*
::1/128	::	lo
fe80::22a:cff:fe32:5ed1/128	::	lo
fe80::/64	::	eth0
2001:db8:a319:15:22a:cff:fe32:5ed1/128	::	lo
2001:db8:a319:15::/64	::	eth0
ff00::/8	::	eth0
::/0	fe80::1	eth0

Směrovací protokoly

- žádná velká změna, modifikace obvyklých protokolů
- **RIPng**
  - pro malé sítě, RFC 2080
- **OSPFv3**
  - pro větší sítě, RFC 5340
- **BGP4+**
  - externí směrování mezi ISP, RFC 4760

Objevování sousedů

Funkce

- zjišťování linkových adres (náhrada ARP)
- ověřování dosažitelnosti sousedů
- automatická konfigurace
- detekce duplicitních adres

Hledání linkových adres (1)

- nalezení MAC adresy stroje ve stejné podsíti
- tazatel pošle **Výzvu sousedovi (Neighbor solicitation)**
- posílá se na **skupinovou adresu vyzývaného uzlu** (solicited node address)
  - prefix ff02::1:ff00:0/104
  - posledních 24 bitů se vezme z poptávané adresy
  - např. shání-li MAC pro 2001:db8:1:7:2a5:14ff:fe1b:24c, pošle dotaz na ff02::1:ff1b:24c

Hledání linkových adres (2)

- ve skupině budou všechny stroje, jejichž některé rozhraní končí danými 3 bajty (většinou jen jeden)
- kompletní hledaná adresa je součástí Výzvy sousedovi
- držitel adresy odpoví **Ohlášením souseda (Neighbor advertisement)** obsahujícím poptávanou IPv6 adresu a v připojené volbě MAC adresu

Cache sousedů

- analogie ARP cache
- obsahuje informace o adresách v přímo připojených podsítích a jejich stavech
- položky jsou dynamicky přidávány a odebírány

Dosažitelnost souseda

- snaží se minimalizovat zátěž
- dokud od souseda přicházejí datagramy pro vyšší vrstvy, je považován za dosažitelného
- ustane-li komunikace, po chvíli je adresa převedena do stavu *prošlá*, nic se neděje
- až s odesláním paketu se mění na *odloženou* – čeká se na příchod paketu
- nepřijde-li, ověří dostupnost *Výzvou sousedovi*

Automatická konfigurace

Nastavení síťových parametrů

- **bezstavové**
  - nevyžaduje servery
  - informace podávají směrovače
  - každý si nastavuje sám
- **stavové**
  - DHCPv6
  - staré známé DHCP vylepšené až k nepoužitelnosti

Bezstavová autokonfigurace (1)

- RFC 4862: IPv6 **Stateless Address Autoconfiguration** (SLAAC)
- směrovače opakovaně vysílají **Ohlášení směrovače**
  - jak dlouho má být **implicitním směrovačem**
  - Omezení skoků pro IPv6 hlavičku
  - jak dlouho považovat zdejší stroje za dosažitelné
  - volba: zdejší **prefix**
  - volba: MTU
  - volba: **DNS informace** (RFC 6106)

Bezstavová autokonfigurace (2)

- uzel počká na *Ohlášení směrovače* (nebo o ně požádá *Výzvou směrovači*)
- vytvoří si svůj **identifikátor rozhraní** (nastavený, mEUI-64, náhodně...)
- vytvoří **lokální linkovou adresu** a detekcí duplicit ověří, že je unikátní
- připojí stejný identifikátor rozhraní za každý z ohlášených prefixů a vytvoří **globální adresy** (už netestuje)

Bezstavová autokonfigurace (3)

- **směrování**
  - každý směrovač, který v *Ohlášení směrovače* poslal nenulovou dobu svého fungování v roli implicitního směrovače, je přidán do seznamu implicitních směrovačů
- **jak konfigurovat?**
  - Ohlášení směrovače obsahuje **příznaky M** (Managed address configuration) a **O** (Other stateful configuration)
  - M=1 – vše konfigurovat pomocí DHCPv6
  - M=0, O=1 – adresu a směrování SAAC, zbytek DHCPv6
  - M=0, O=0 – vše bezstavově

DHCPv6

- RFC 3315:
  Dynamic Host Configuration Protocol for IPv6
- základ podobný DHCPv4
- založeno na DHCP serverech, umí řadu informací
- **objevování (discover)** – klient poptává na ff02::1:2
- **nabídka (offer)** – servery nabídnou parametry
- **žádost (request)** – klient si vybere a požádá server
- **potvrzení (acknowledge)**

DUID

- identifikátor klienta – DHCP Unique Identifier
- několik variant:
  - přidělený výrobcem
  - vygenerovaný a uložený operačním systémem
  - vycházející z MAC (jedna MAC pro všechna rozhraní)

Problémy DHCPv6

- nenastavuje směrování
  - lokální linkové adresy nejsou pro server jednoznačné
  - směrování doplňuje bezstavová autokonfigurace
- široce podporované jsou jen DUID generované OS
  - reinstalace systému změní DUID
  - více OS na jednom stroji – více DUID
  - klonování OS – stejný DUID na více strojích
- obtížně použitelné pro pevné přidělování

Bezstavové DHCPv6

- RFC 3736: **Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6**
- kombinace stavové a bezstavové konfigurace
- adresy a směrování se nastaví bezstavově, zbytek (DNS a případné další informace) se nastaví přes DHCPv6
- nemusí používat DUID ani ukládat informace o klientech, tyto parametry bývají pro všechny stejné

Konfigurace reálně

- obvykle se používá bezstavová konfigurace v kombinaci s bezstavovým DHCPv6
  - DHCPv6 bude možné opustit, až se rozšíří podpora pro volbu s DNS – nově ve Windows 10 (v1703)
- druhou možností je DHCPv6 v režimu „kdo dřív přijde, dostane adresu“
  - případně doplněna o pevné přidělování síťových parametrů vybraným strojům (servery, stanice pro dohled a správu sítě apod.)

Bezpečnostní prvky

Objevování sousedů

- zprávy nejsou chráněny – lze je padělat
  - vydávat se za jiného
  - švindlovat se směrováním
  - předstírat duplicitu adres

RA Guard

- RFC 6105: **IPv6 Router Advertisement Guard**
- **implementuje centrální L2 prvek** (přepínač)
- **kontroluje zprávy ohlašování sousedů** a propouští jen ty, které považuje za korektní
- **bezstavový** – vychází jen ze své konfigurace
- **stavový** – využívá dříve zjištěné informace (např. fáze učení a fáze kontroly)
- výhoda: stačí podpora v ethernetovém přepínači

Anihilace falešných ohlášení

- specializované programy – např. **ramond**
- přijímá a kontroluje ohlášení směrovače
- dorazí-li falešné, okamžitě je zruší – pošle stejné s nulovou životností

SEND (1)

- **RFC 3971: SEcure Neighbor Discovery (SEND)**
- rozšiřuje objevování sousedů o bezpečnostní mechanismy
- založeno na asymetrické kryptografii, každý uzel má svůj soukromý a veřejný klíč
- používá speciální adresy (CGA) odvozené z veřejných klíčů zařízení
- ohlášení směrovače ověřuje certifikačními cestami

CGA

- **RFC 3972: Cryptographically Generated Addresses (CGA)**
- identifikátor rozhraní
- vychází z veřejného klíče (SHA-1 hash z klíče a dalších údajů)
- při objevování sousedů doprovázena **volbou CGA** obsahující veřejný klíč a umožňující adresu ověřit
- **dokazuje vlastnictví** příslušného **soukromého klíče**

SEND (2)

- přidává ke zprávám objevování sousedů **volbu s digitálním podpisem**
- funguje jen pro CGA adresy (lze důvěřovat klíči)
- ověření podpisu znamená, že zpráva je bezpečná
- zpracování zpráv, které nejsou bezpečné, je otázkou konfigurace
  - RFC 3971 požaduje, aby implicitně byly přijímány (malé rozšíření SEND)

Problémy SEND

- vyžaduje podporu ve všech koncových zařízeních (která je dosud dost vzácná)
- neumí chránit jiné adresy než CGA
- tato omezení komplikují praktické nasazení – příležitost pro jednodušší alternativy

SAVI (1)

- Source Address Validation Improvement, RFC 7039
- ověření adresy odesilatele **pro lokální provoz**
- 3 kroky:
  1) zjistit, jaké IP adresy je stroj oprávněn používat
  2) svázat je s vlastnostmi linkové vrstvy
  3) vynutit, aby pakety odpovídaly vazbě podle 2)
- lze implementovat v různých místech, nejlépe v přepínačích připojujících koncové stroje

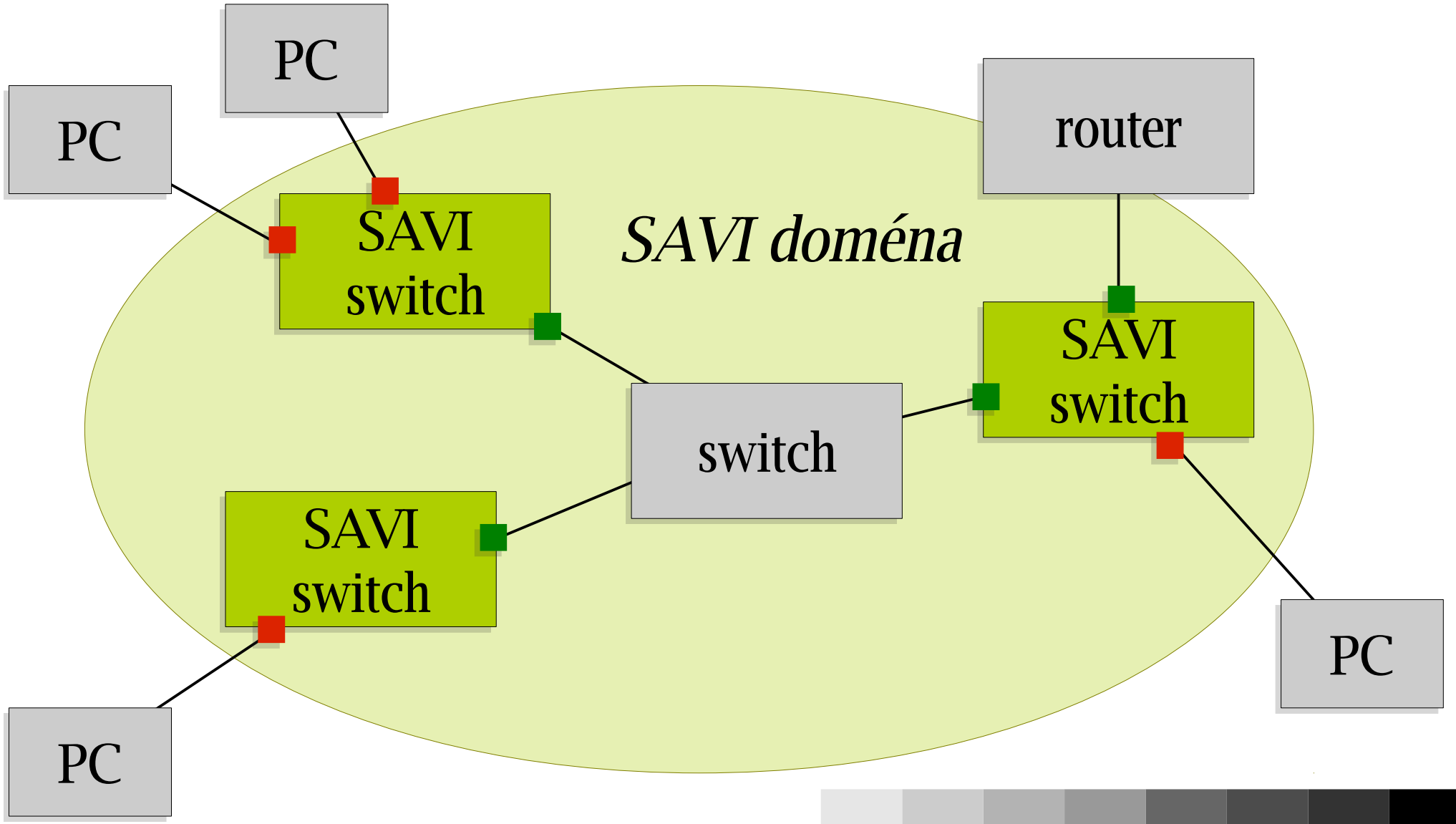
SAVI (2)

- **zjištění korektních adres**
  - závisí na použité metodě přidělování
  - např. sledování DHCPv6 komunikace nebo paketů automatické autokonfigurace
- **vazba IP adresy na L2 prvky**
  - různé metody, různá úroveň bezpečnosti
  - typicky číslo portu na přepínači
  - EUI-64 nebo podobný identifikátor
  - identifikátor tunelu, ...

SAVI (3)

- princip **obranného perimetru**
- **porty k ostatním SAVI prvkům jsou důvěryhodné**
- **porty vedoucí k ne-SAVI prvkům jsou prověřovány**
- SAVI zařízení si uchovává jen informace o paketech přicházejících z prověřovaných portů (o přímo připojených ne-SAVI sousedech)

SAVI (4)


PC
PC
router
SAVI
switch
SAVI doména
SAVI
switch
switch
SAVI
switch
PC
PC

FCFS SAVI

- **First-Come First Served**, RFC 6620
- určeno zejména pro bezstavovou autokonfiguraci
- když se **poprvé objeví IPv6 adresa**, zařízení implementující FCFS SAVI si ji **sváže s L2 údaji** (zde povinně **číslo portu**) a následně propouští s touto adresou jen pakety ze stejného portu
- **při změně L2** (přepojení do jiného portu) **ověří nedostupnost** původních parametrů

SEND SAVI

- pro sítě používající SEND, RFC 7219
- **váže CGA na L2 port** pokud
  - počítač ověřuje detekcí duplicit unikátnost své adresy
  - přijde jiný paket a SAVI prvek ověří detekcí dosažitelnosti, že odesilatel je pravý
  - a v obou případech jsou platné SEND podpisy
- platnost vazby oživuje detekcí dosažitelnosti

IPsec

- bezpečnostní mechanismy v síťové vrstvě, RFC 4301
- dvě rozšiřující hlavičky:
- **Authentication Header (AH)**
  - ochrana proti padělání odesilatele, změně datagramu
- **Encapsulating Security Payload (ESP)**
  - navíc šifrování
- implementace IPsec není povinná (should), pokud je podporováno, musí obsahovat ESP, může AH

Režimy ochrany

- **transportní**: ESP/AH hlavička je vložena přímo do hlaviček datagramu
- **tunelující**: celý původní datagram (včetně hlavičky) se zabalí jako data do nového datagramu opatřeného ESP/AH hlavičkou
  - silnější, umožňuje skrýt adresy
- může probíhat mezi koncovými stroji nebo ji mohou (transparentně) poskytnout bezpečnostní brány po cestě – à la VPN

AH

- RFC 4302
- autentizuje odesilatele a obsah datagramu
- při odeslání převede datagram do „normalizované“ podoby (vynuluje nepředvídatelné položky), vypočte **digitální podpis** a vloží do hlavičky AH
- příjemce provede analogickou operaci a zkontroluje podpis

ESP

- RFC 4303
- **šifrování i autentizace** (služby volitelné, doporučuje se používat obě)
- nestandardní hlavička – **„spolkne“ do sebe celý zbytek datagramu** jako svá nesená data a zašifruje
- provádí se pro celý datagram, případná fragmentace až po zašifrování

Bezpečnostní asociace

- virtuální spojení dvou počítačů
- **určuje kryptografické operace a jejich parametry**
- např. použít ESP v tunelovém režimu,
  šifra AES-CBC s klíčem hychykyry,
  autentizace HMAC+SHA-1 s klíčem befeleme
- jednosměrná a pro jeden protokol (ESP nebo AH)
- navazují se po dvojicích – jedna pro každý směr
  (jiné klíče pro odesílání a příjem)

Správa asociací

- **manuální**
  - jistota je jistota
  - neškáluje, nelze často střídat parametry
- **IKEv2**
  - Internet Key Exchange Protocol Version 2, RFC 5996
  - umožňuje dynamicky vytvářet a rušit asociace
  - nejprve dohodne protokol (jedna strana navrhne, druhá si z návrhů vybere)
  - následně klíče (Diffie-Hellman)

Řízení IPsec

- **Security Policy Database (SPD)**
- sada pravidel, co s příchozím/odchozím datagramem provést
  - zahodit
  - propustit bez IPsec
  - podrobit IPsec s odkazem na bezpečnostní asociaci
- pravidla typu: všechny datagramy směřující do 2001:db8:2::/48 opatřit IPsec podle asociace 123

DNS

IPv6 v DNS

- problematická historie
- RFC 1886 navrhlo minimalistickou verzi
- RFC 2874 ji nahradilo sofistikovanější (a složitější)
- RFC 3596 **DNS Extensions to Support IP Version 6** se vrátilo k původnímu modelu

Záznam AAAA

- uložení adresy: záznam typu AAAA
- jako typ A, adresa je čtyřnásobně dlouhá
- příklad:
  bubo.tul.cz. IN AAAA 2001:718:1c01:16::aa
- DNS resolver se obvykle chová inteligentně – pokud má k dispozici oba protokoly, poptává podle konfigurace nejprve jeden (obvykle AAAA) a následně druhý (A), případně paralelně oba

Reverzní DNS

- standardní záznam typu PTR
- vezme se kompletní zápis IPv6 adresy (včetně všech nul), otočí se pořadí číslic a udělají se z nich domény
- na konec se připojí *ip6.arpa*
- příklad:
  a.a.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.6.1.0.0.1.0.c.1.8.1.7.0
  .1.0.0.2.ip6.arpa. IN PTR bubo.tul.cz.

Koexistence s IPv4 (1)

- **varianta 1: oba pod stejným jménem**
  - www IN A 10.0.1.2
    IN AAAA 2001:db8:1:1::2
  - klient při hledání www.example.com podle svého připojení poptá jeden nebo oba
  - systémový přístup
  - oba protokoly by pro daný stroj měly spolehlivě fungovat

Koexistence s IPv4 (2)

- **varianta 2: odlišná jména**
  - typicky ip6.doména
  - www IN A 10.0.1.2
    www.ip6 IN AAAA 2001:db8:1:1::2
  - klient si vybírá protokol volbou jména (www.example.com nebo www.ip6.example.com)
  - vhodné pro experimenty – případné problémy IPv6 nenaruší komunikaci se serverem
  - využití IPv6 bude malé

Podpora mobilních zařízení

Domácí adresa

- mobilní zařízení na cestách střídá **dočasné adresy** podle sítě, kde se aktuálně vyskytuje
- navazuje-li komunikaci mobilní stroj, použije aktuální adresu
- problém, pokud chce někdo navázat spojení s ním
- mobilní stroj má někde svou domácí síť, v ní **domácí adresu** uvedenou v DNS – na ni se navazuje spojení

Domácí agent

- je-li mobilní stroj na cestách, zastupuje jej v domácí síti **domácí agent**
- jeden ze zdejších směrovačů
- mobilní stroj mu ohlásí svou aktuální dočasnou adresu (*Aktualizace vazby*)
- domácí agent při objevování sousedů předstírá, že je mobilní uzel – stahuje na sebe jeho datagramy a přeposílá je mobilnímu stroji ESP tunelem

Optimalizace cesty

- aby komunikace neprocházela přes domácího agenta
- mobilní uzel (když dostane paket tunelem od agenta) pošle *Zahájení pro dočasnou adresu* a *Zahájení pro domácí adresu*
- protějšek odpoví odesláním *Tokenu pro dočasnou adresu* a *Tokenu pro domácí adresu*
- mobilní uzel pošle *Aktualizaci vazby* doprovázenou oběma tokeny (potvrdí, že obě adresy jsou jeho)
- protějšek potvrdí

Rozšiřující hlavičky

- **Mobilita**
  - zprávy výlučně pro podporu mobility (aktualizace vazby, její potvrzení, zprávy pro optimalizaci cesty apod.)
- **Směrování typu 2**
  - partner po optimalizaci cesty posílá pakety na dočasnou adresu a přidává tuto hlavičku obsahující domácí adresu
- **Volby pro příjemce – volba Domácí adresa**
  - mobilní uzel posílá pakety z dočasné adresy a připojuje informaci o své domácí adrese

Změna adresy

- mobilní uzel si udržuje informace o svých partnerech (kdo všechno má pro něj vazbu)
- změní-li dočasnou adresu, pošle všem (včetně domácího agenta) *Aktualizaci vazby* s novou dočasnou adresou
- návrat domů – speciální případ změny, posílá *Aktualizaci vazby* s nulovou životností (žádost o zrušení vazby) plus několik *Ohlášení souseda*, aby začal dostávat své datagramy

Přechodové mechanismy

Obecné principy

- **dual stack**
  - podporovány jsou oba protokoly
- **tunelování**
  - účastníci komunikace používají stejný protokol, síť mezi nimi jej nepodporuje
  - jeden protokol se zabalí jako data do druhého
- **překlad**
  - účastníci hovoří různými protokoly
  - jeden se překládá na druhý

Koncové zařízení

- potřebuje IPv4 – řada služeb není jinak dostupná
- má-li mít i IPv6, nabízejí se tyto varianty:
  - oba protokoly nativně
  - jeden protokol nativně, druhý tunelem
  - jeden protokol nativně, druhý překládat – počítač komunikuje jen jedním protokolem, dual-stack jsou jen jeho aplikace

Dual stack

- soudobé operační systémy typicky podporují oba
- záleží jen na síti, do níž je zařízení připojeno
- v současnosti asi nejlepší variantou je podporovat v síti oba protokoly, pokud možno rovnocenně
- **výhody:** protokoly pracují nativně, nezávisí na sobě
- **nevýhody:** potřebujete IPv4 adresy, může vést k podivným a obtížně zjistitelným problémům

Happy Eyeballs (1)

- RFC 6555, doporučení pro **autory aplikací**
- nečekat na timeout a zkusit druhý protokol brzy
- seřadí adresy získané z DNS standardně
- zkusí první
- nedorazí-li brzy odpověď (doporučeno čekat 150–200 ms), zkusí první adresu **druhého protokolu** a pak postupně další, dokud neuspěje
- vytvoří-li více spojení, až na jedno všechna zavře

Happy Eyeballs (2)

- výsledek si ukládá do dočasné paměti
- opakuje-li se komunikace se stejným cílem, zkouší nejprve adresu, která minule „zvítězila“
- implementováno v Chrome, Firefoxu (řídí network.http.fast-fallback-to-IPv4), Opeře a v OS X Lion

Tunelování

- překonává části sítě, které nepodporují daný protokol
- např. chcete doma mít IPv6, ale váš poskytovatel Internetu podporuje jen IPv4
- na vstupu do tunelu je původní datagram zabalen jako data do datagramu druhého protokolu
- nový datagram se dopraví na druhý konec tunelu
- zde se vybalí původní datagram a pokračuje dál

Statické tunely

- explicitně konfigurované (manuálně nebo programem)
- obvykle dlouhé trvání
- topologie nebývá ideální
- veřejné tunel servery umožňují připojení komukoli
  - v ČR k dispozici Hurricane Electric (tunnelbroker.net)

Automatické tunely

- tunel není udržován trvale, každý datagram se tuneluje jinam
- IPv6 adresa obvykle obsahuje IPv4 adresu druhého konce tunelu
- méně problémů s topologií – pakety obvykle směřují celkem přímočaře k cíli
- obtížnější hledání a odstraňování problémů

6rd

- RFC 5569: **IPv6 Rapid Deployment**
- v rámci ISP – má IPv4 páteř, chce IPv6 pro klienty
- koncová síť má alespoň jednu IPv4 adresu
- ISP vyhradí IPv6 prefix pro 6rd, za něj vždy připojí IPv4 → prefix koncové sítě
- ISP provozuje relay pro předávání 6rd–IPv6 Internet
- Free (Francie) nasadil během měsíce

6to4

- RFC 3056
- předchůdce 6rd
- stejný princip, ale globální prefix 2002::/16
- vedl k asymetrickému směrování, nepředvídatelnému chování a vysoké nespolehlivosti – opuštěn

ISATAP

- RFC 5214: **Intra-Site Automatic Tunnel Addressing Protocol**
- IPv6 konektivita v lokální IPv4 síti
- používá IPv4 jako linkový protokol
- adresa rozhraní: 0:5efe + IPv4 adresa rozhraní
- implicitní směrovače obvykle zjišťuje pomocí DNS, poptá jméno *isatap* v místní doméně
- nemá valný smysl – nasaďte IPv6 nativně

Teredo

- RFC 4380
- **koncové sítě dnes často za NATem, Teredo se snaží jím procházet**
- poměrně složité – přes vnější Teredo server se snaží otevřít cesty v NATech, aby komunikace mohla procházet přímo
- **notoricky nespolehlivé**
- implementováno ve Windows

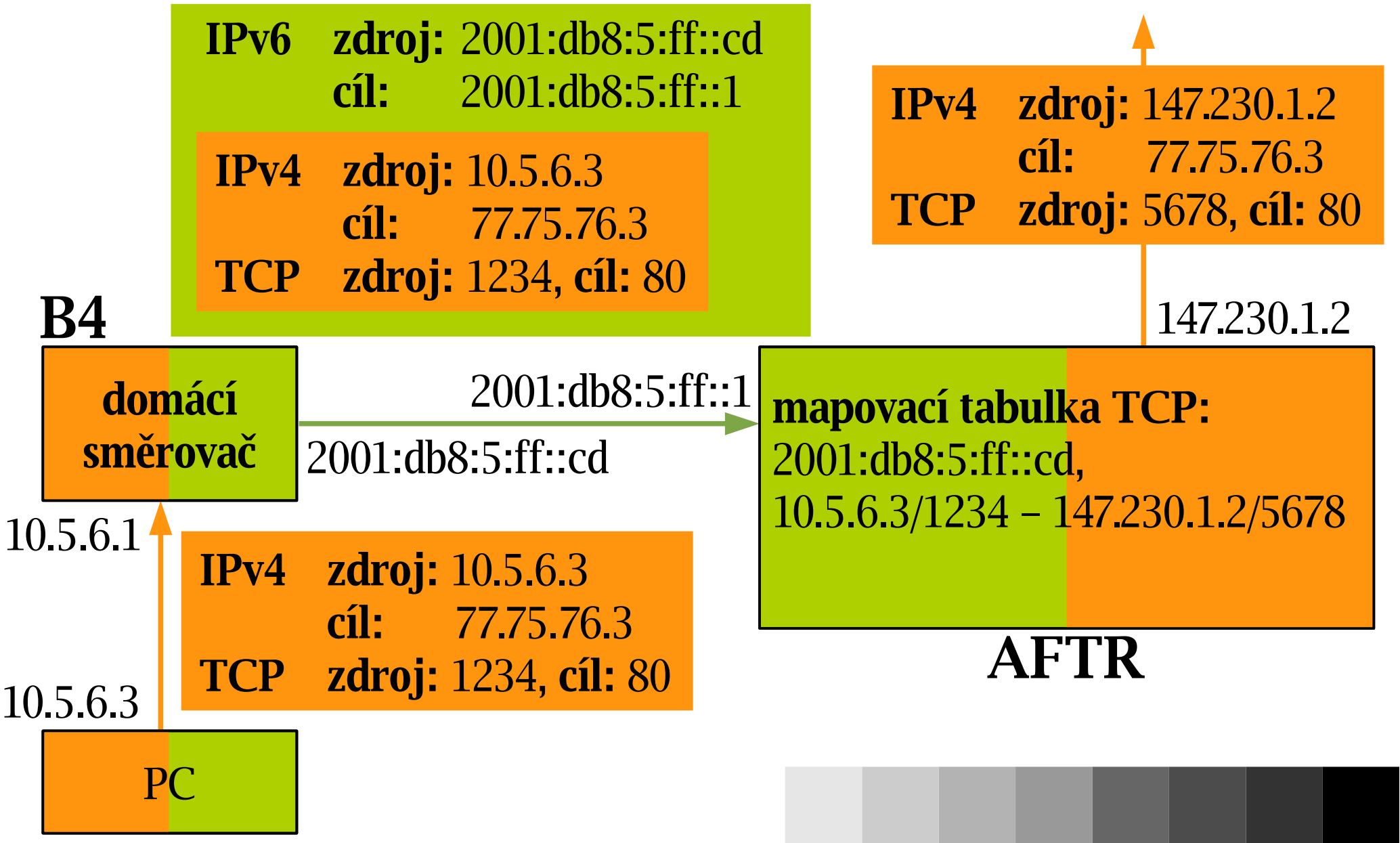
DS Lite (1)

- RFC 6333: **Dual-Stack Lite**
- IPv4 adresy budou stále vzácnější – problém pro dual-stack páteřní sítě
- DS Lite obrací role – **páteřní síť bude jen IPv6, IPv4 pro zákaznické sítě se bude tunelovat na centrální NAT**
  - identifikace zákaznického stroje: jeho IPv4 adresa + IPv6 adresa jeho přístupového směrovače
  - zákazníci mohou mít kolidující IPv4 adresy (10.0.0.0)

DS Lite (2)

- **Basic Bridging BroadBand (B4)**
  - zákaznický prvek
  - není NAT, jen balí/rozbaluje IPv4 datagramy do IPv6 a posílá na adresu centrálního NATu (nastavena nebo přes DHCPv6)
- **Address Family Translation Router (AFTR)**
  - centrální NAT, jen zde se mění IPv4 datagramy
  - koncové stroje identifikovány IPv6 adresou jejich B4, svou IPv4 adresou, protokolem a portem

DS Lite (3)


IPv6 zdroj: 2001:db8:5:ff::cd
cíl: 2001:db8:5:ff::1
IPv4 zdroj: 10.5.6.3
cíl: 77.75.76.3
TCP zdroj: 1234, cíl: 80
IPv4 zdroj: 147.230.1.2
cíl: 77.75.76.3
TCP zdroj: 5678, cíl: 80
147.230.1.2
B4
domácí směrovač
2001:db8:5:ff::1
2001:db8:5:ff::cd
mapovací tabulka TCP:
2001:db8:5:ff::cd,
10.5.6.3/1234 – 147.230.1.2/5678
AFTR
10.5.6.1
IPv4 zdroj: 10.5.6.3
cíl: 77.75.76.3
TCP zdroj: 1234, cíl: 80
10.5.6.3
PC

Lightweight 4over 6 (lw4o6)

- RFC 7596
- podobné DS Lite, ale NATuje domácí směrovač
  - má omezený sortiment portů – zákazníci mohou sdílet stejnou IPv4 adresu
- centrální AFTR zajišťuje jen tunelování
  - nižší nároky

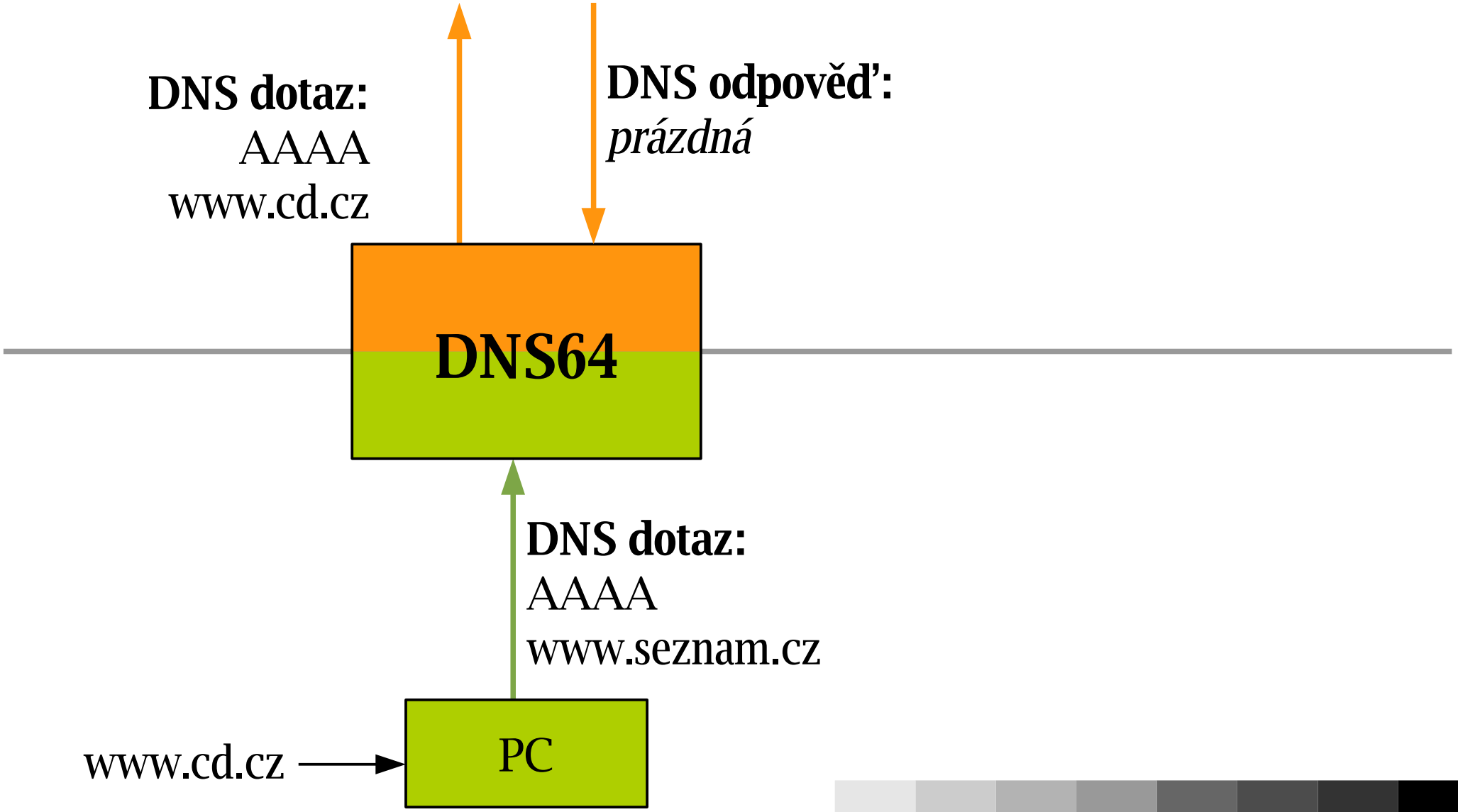
SIIT

- RFC 6145: **IP/ICMP Translation Algorithm**
- **překlad mezi IPv4 a IPv6**
  - přesná pravidla, jak a z čeho vytvořit kterou položku hlavičky
  - jen základní hlavičky, zahazuje rozšiřující hlavičky IPv6 a volby IPv4
- neřeší mapování adres
- používá se jako součást sofistikovanějších mechanismů

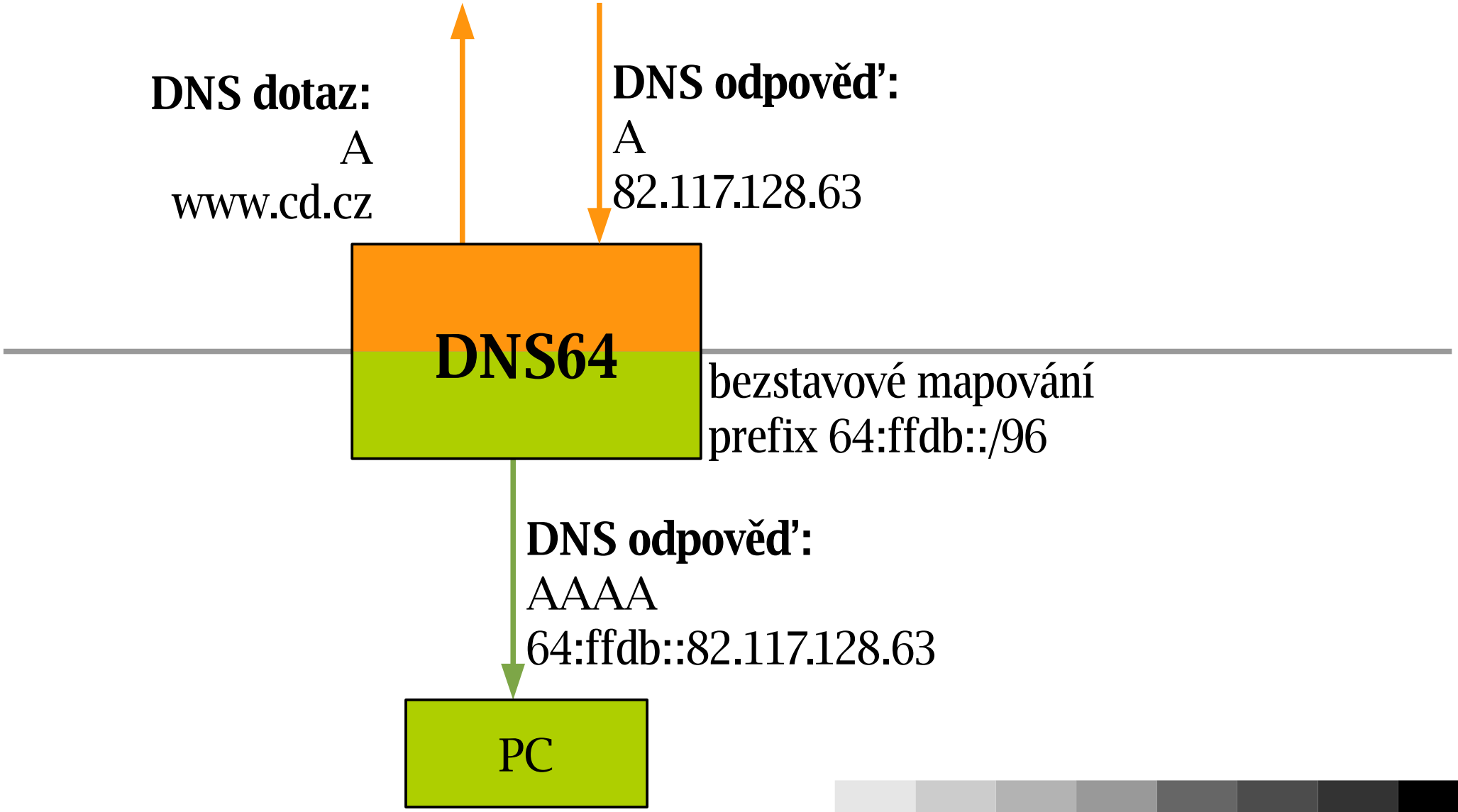
NAT64 + DNS64 (1)

- cíl: **zpřístupnit IPv6 klientům v koncové síti služby dostupné jen po IPv4 (omezeně i opačně)**
- předchůdcem byl **NAT-PT**, snažil se o oboustrannou komunikaci, trpěl řadou problémů
- specifikace:
  - RFC 6144 – obecný rámec
  - RFC 6145 – překlad datagramů
  - RFC 6146 – NAT64
  - RFC 6147 – DNS64

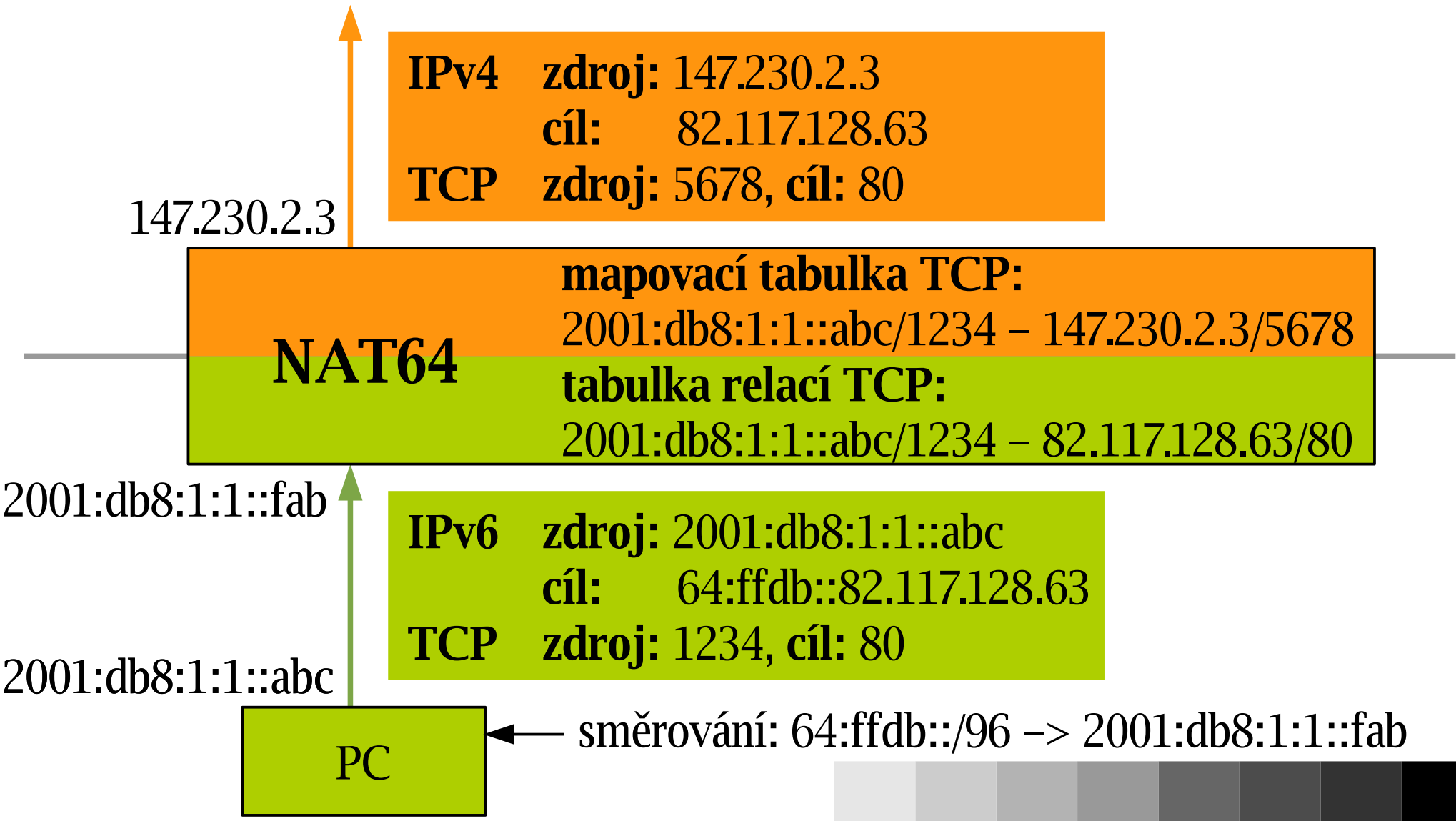
1. fáze: DNS dotaz AAAA



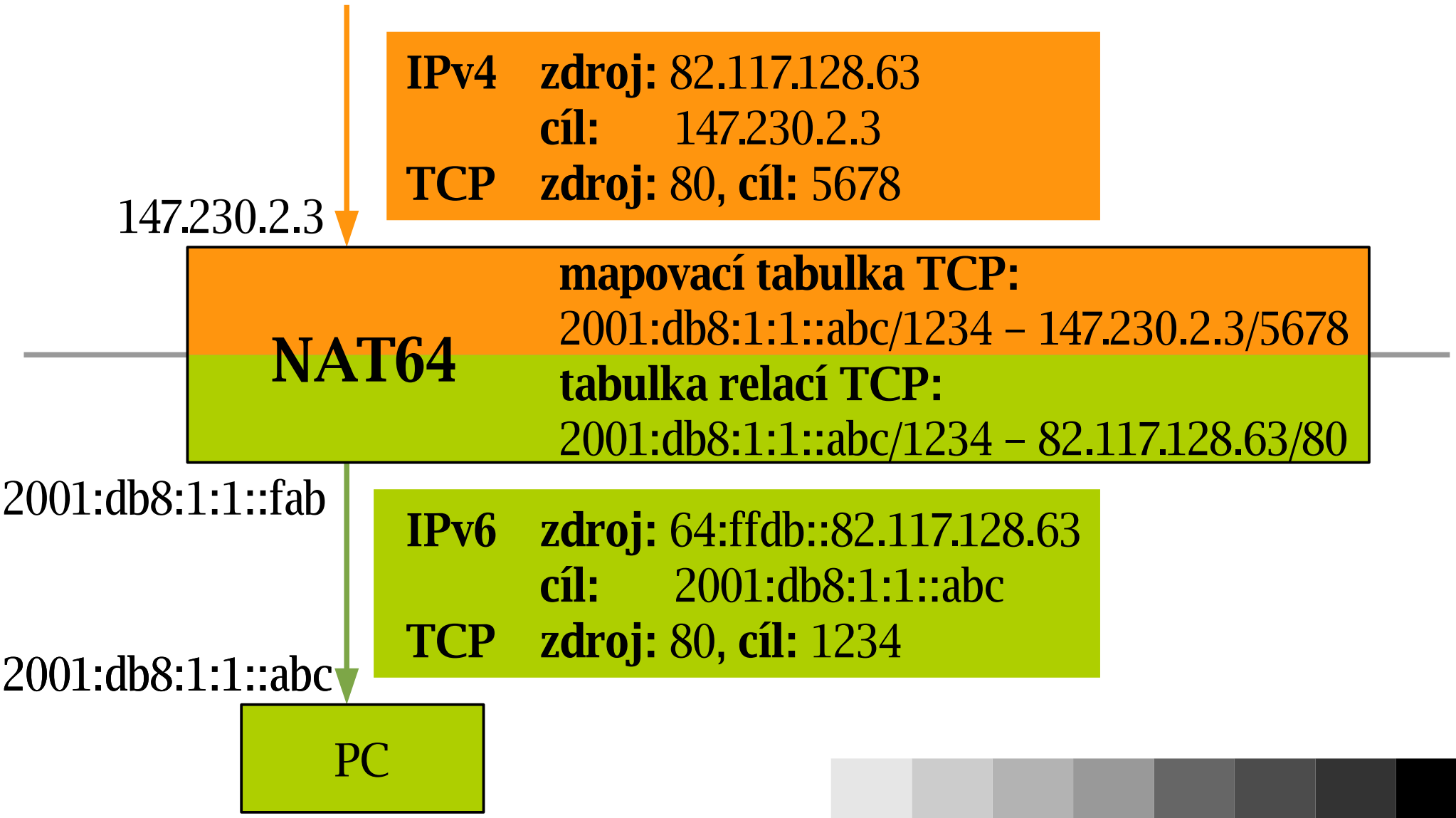
2. fáze: DNS dotaz A



3. fáze: odeslání datagramu



4. fáze: příchod datagramu



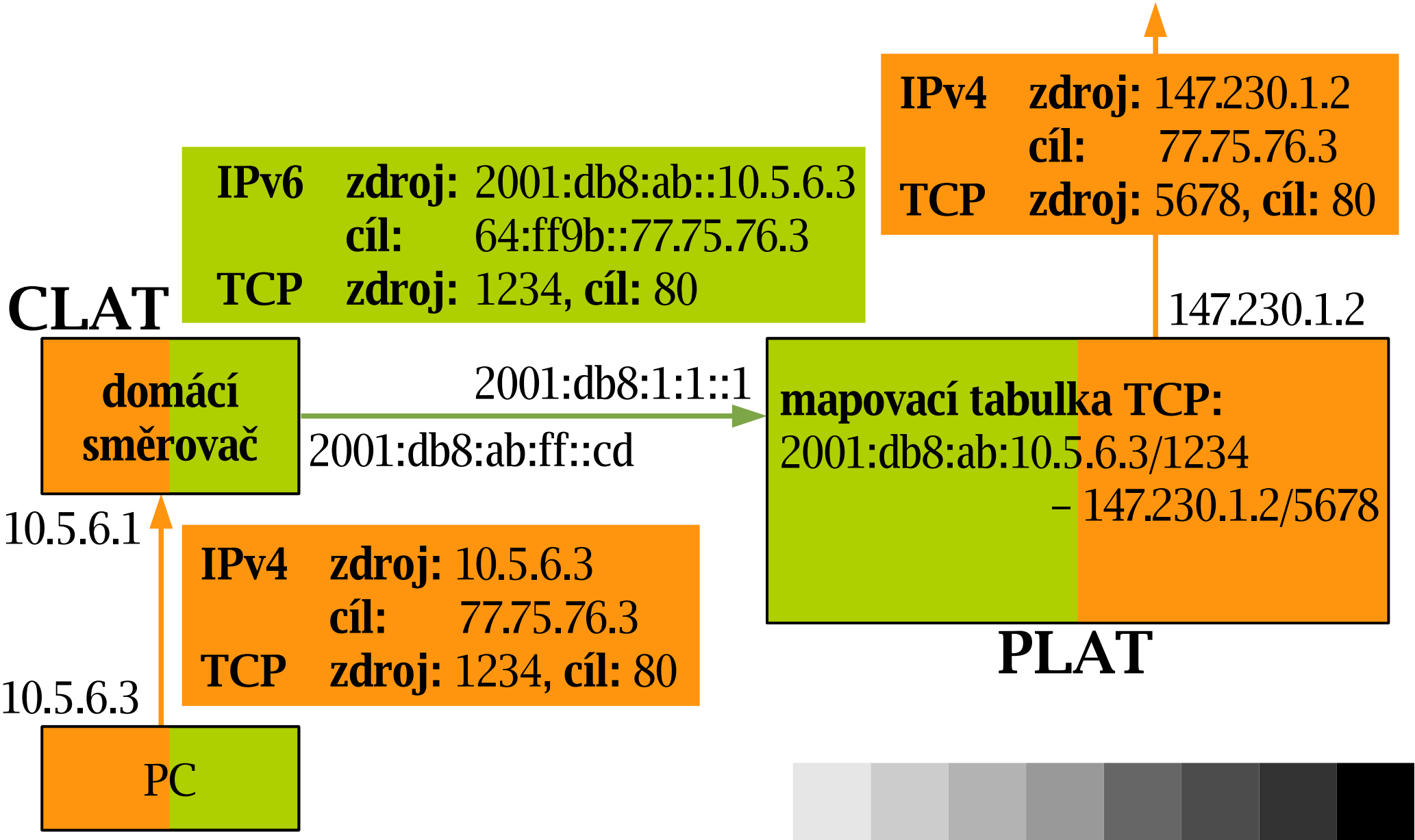
NAT64 + DNS64 (2)

- asymetrické
  - mapování IPv4 do IPv6 staticky (prefix)
  - mapování IPv6 do IPv4 dynamicky (překladová tabulka)
- NAT64 a DNS64 nemusí běžet na stejném stroji
- bezproblémově lze navázat spojení jen z IPv6 (protější směr lze pevnými položkami v tabulce)
- podporuje jen protokoly UDP, TCP, ICMP
- počítá se i s filtrováním a dalšími obvyklými vlastnostmi NATů

464XLAT (1)

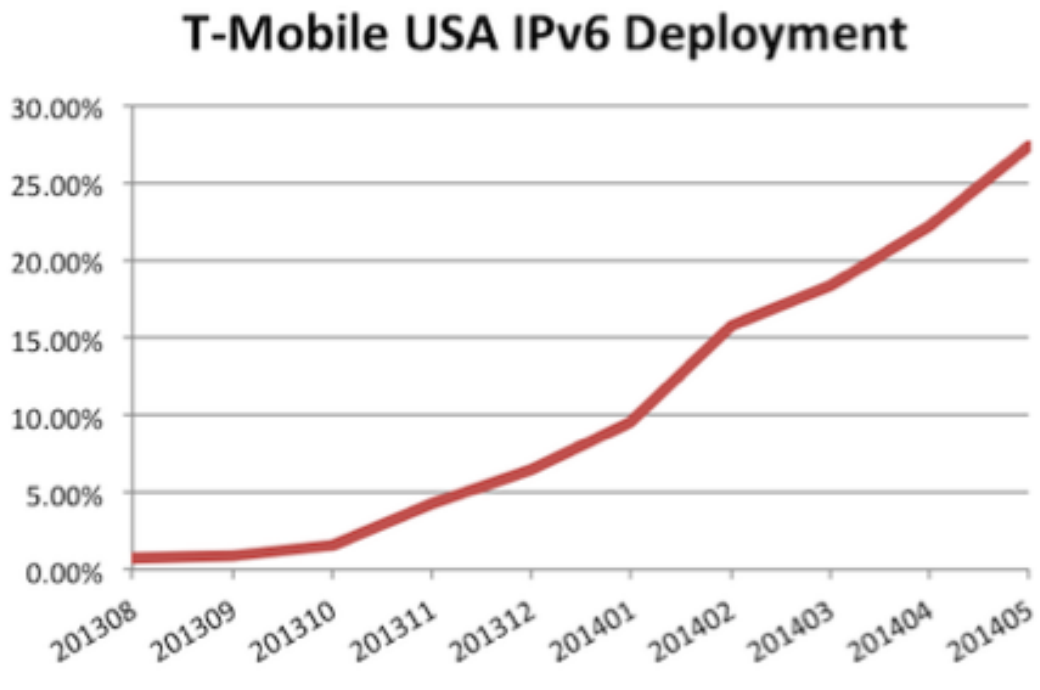
- RFC 6877: **páteř jen IPv6, obě strany hovoří IPv4**, à la DS-Lite, ale místo tunelování používá **dvojitý překlad** IPv4 → IPv6 → IPv4
- **CLAT**
  - NAT na straně klienta
  - bezstavový překlad, prefix získá z DNS (ipv4only.arpa)
- **PLAT**
  - centrální NAT u poskytovatele – NAT64 + DNS64
  - stavový překlad

464XLAT (2)


IPv4 zdroj: 147.230.1.2
cíl: 77.75.76.3
TCP zdroj: 5678, cíl: 80
IPv6 zdroj: 2001:db8:ab::10.5.6.3
cíl: 64:ff9b::77.75.76.3
TCP zdroj: 1234, cíl: 80
CLAT
147.230.1.2
domácí směrovač
2001:db8:1:1::1
2001:db8:ab:ff::cd
mapovací tabulka TCP:
2001:db8:ab:10.5.6.3/1234
– 147.230.1.2/5678
PLAT
10.5.6.1
IPv4 zdroj: 10.5.6.3
cíl: 77.75.76.3
TCP zdroj: 1234, cíl: 80
10.5.6.3
PC

464XLAT (3)

- masivně nasadil americký T-Mobile ve své mobilní síti (nové Androidy standardně připojuje takto)
- https://www.nanog.org/sites/default/files/wednesday_general_byrne_breakingfree_11.pdf



Další překladové mechanismy

- **Transport Relay Translator (TRT)**
  - podobně jako NAT64, ale překlad se odehrává v transportní vrstvě
  - RFC 3142
- **Bump-in-the-Host (BIH)**
  - soukromý překladač v koncovém zařízení (v síťové vrstvě nebo mezi ní a aplikacemi)
  - RFC 6535

Implementace

Požadavky na implementaci IP

- podle RFC 6540 je IPv6 povinné
- **nové implementace IP musí podporovat IPv6**
- **aktualizace stávajících by měly podporovat IPv6**
- kvalita IPv6 alespoň na úrovni IPv4
- měla by podporovat dual stack, ale fungování nesmí záviset na IPv4
- konkrétní vlastnosti viz RFC 6434 Požadavky na IPv6 uzel

Realita

- IPv6 podporováno (a implicitně zapnuto) snad ve všech současných operačních systémech a platfromách
- **IPv6 Ready**
  - www.ipv6ready.org
  - certifikační program pod hlavičkou IPv6 Fóra
  - různé stupně (stříbrný, zlatý) a kategorie (uzel, směrovač, domácí agent, bezpečnostní brána,...)

IPv6 ve Windows

- poprvé experimentálně v XP (nutno aktivovat)
- **od Vista standardní součástí systému, implicitně zapnuto**
- obsahuje tunelovací mechanismy ISATAP a Teredo, implicitně zapnuty
  - máte IPv6 připojení prakticky kdekoli
  - pokud je stroj připojen jen přes ISATAP nebo Teredo, nepoptává v DNS záznamy AAAA – reálně IPv6 nepoužívá

IPv6 v Linuxu

- první implementace 1996
- několik let spaní na vavřínech
- první pořádná implementace v jádře 2.6 (2005) – vycházela z projektu USAGI
- **v současnosti podpora ve všech velkých distribucích, implicitně zapnuto**
- často nasazen na serverech

Další systémy

- **OS X**
  - podporováno od verze 10.3 Panther (2003)
  - implicitně zapnuto
- **Android**
  - podporováno na straně mobilního připojení, na straně Wi-Fi chybí DNS
  - nemá certifikát IPv6 Ready
  - nepodporuje DHCPv6

Směrovače a aktivní prvky

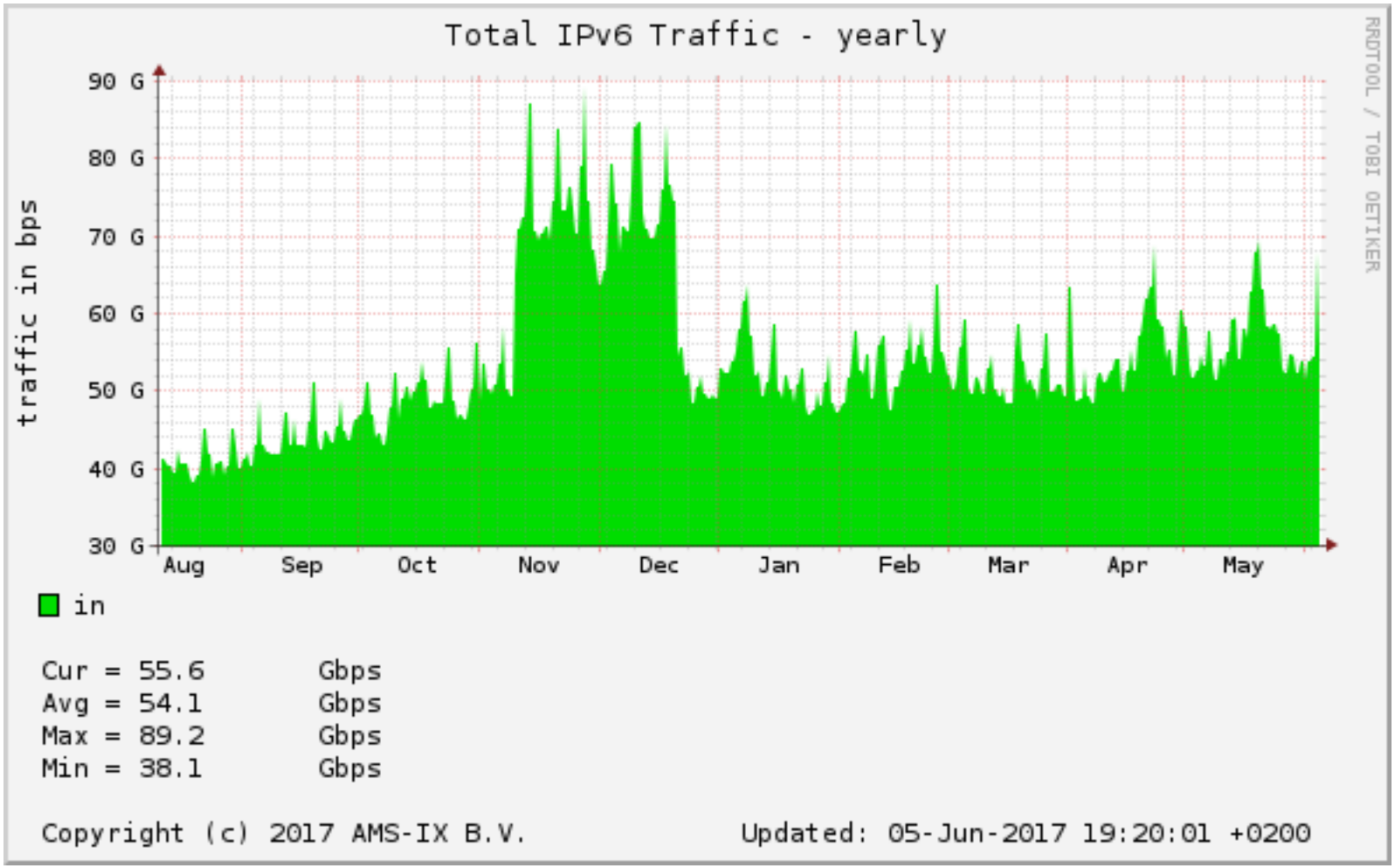
- IPv6 podporují snad všichni výrobci, ne nutně ve všech modelech
- doporučená kritéria pro výběr: **ripe-554**
  - https://www.ripe.net/ripe/docs/ripe-554
  - míněno jako pomůcka pro posuzování výrobků při výběru zařízení
  - které konkrétní IPv6 schopnosti požadovat od různých typů zařízení

IPv6 dnes

Nasazení IPv6

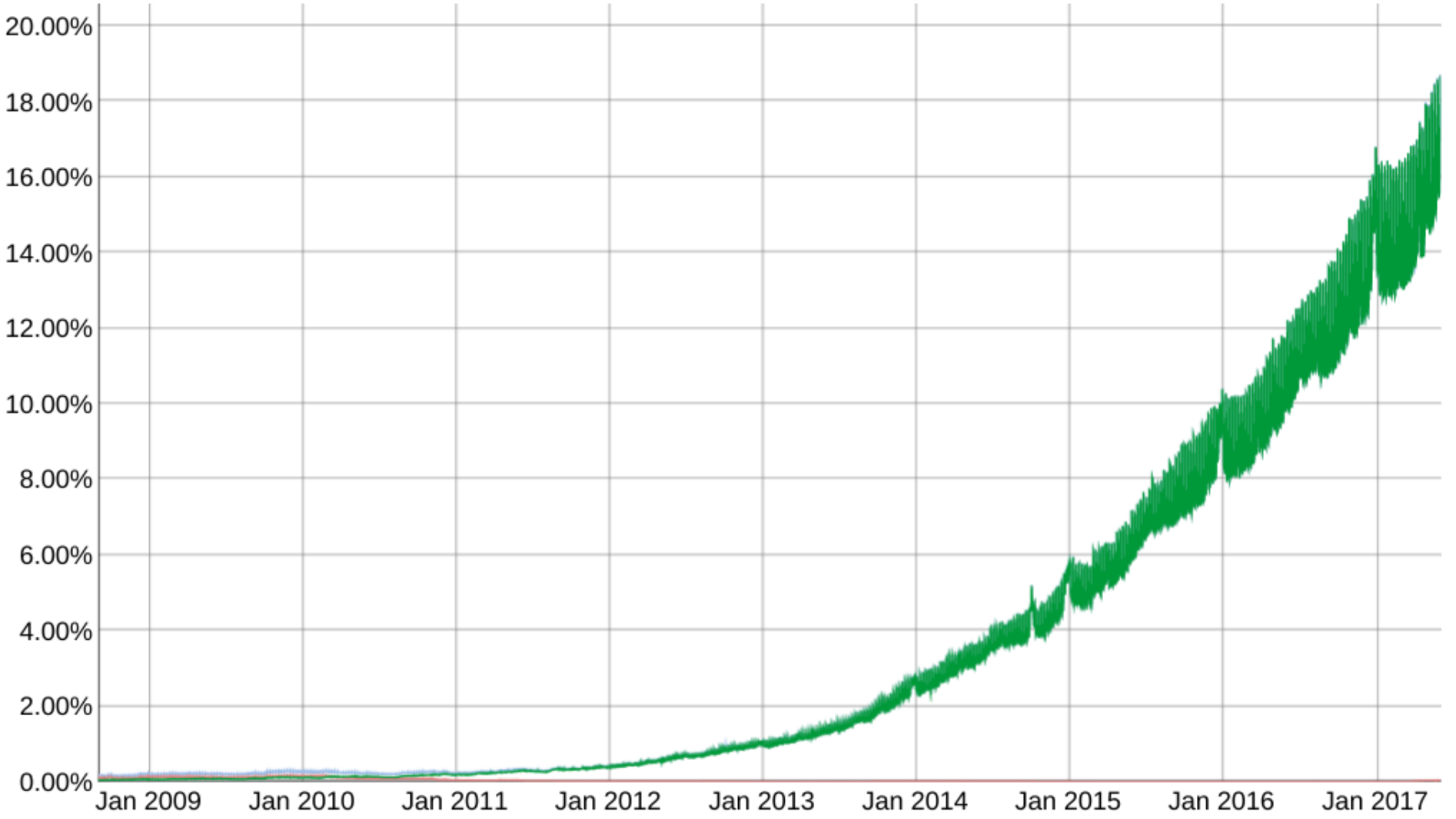
- už jsme dávno měli mít IPv6 Internet
- různé obavy + problém slepice vs vejce
  - kdo by nabízel služby, když nejsou uživatelé?
  - kdo by nabízel uživatelům, když nejsou služby?
- **World IPv6 Launch**
  - 6. 6. 2012
  - velké servery (Google, Facebook, Seznam) nasadily IPv6 do rutinního provozu

Statistika AMS-IX

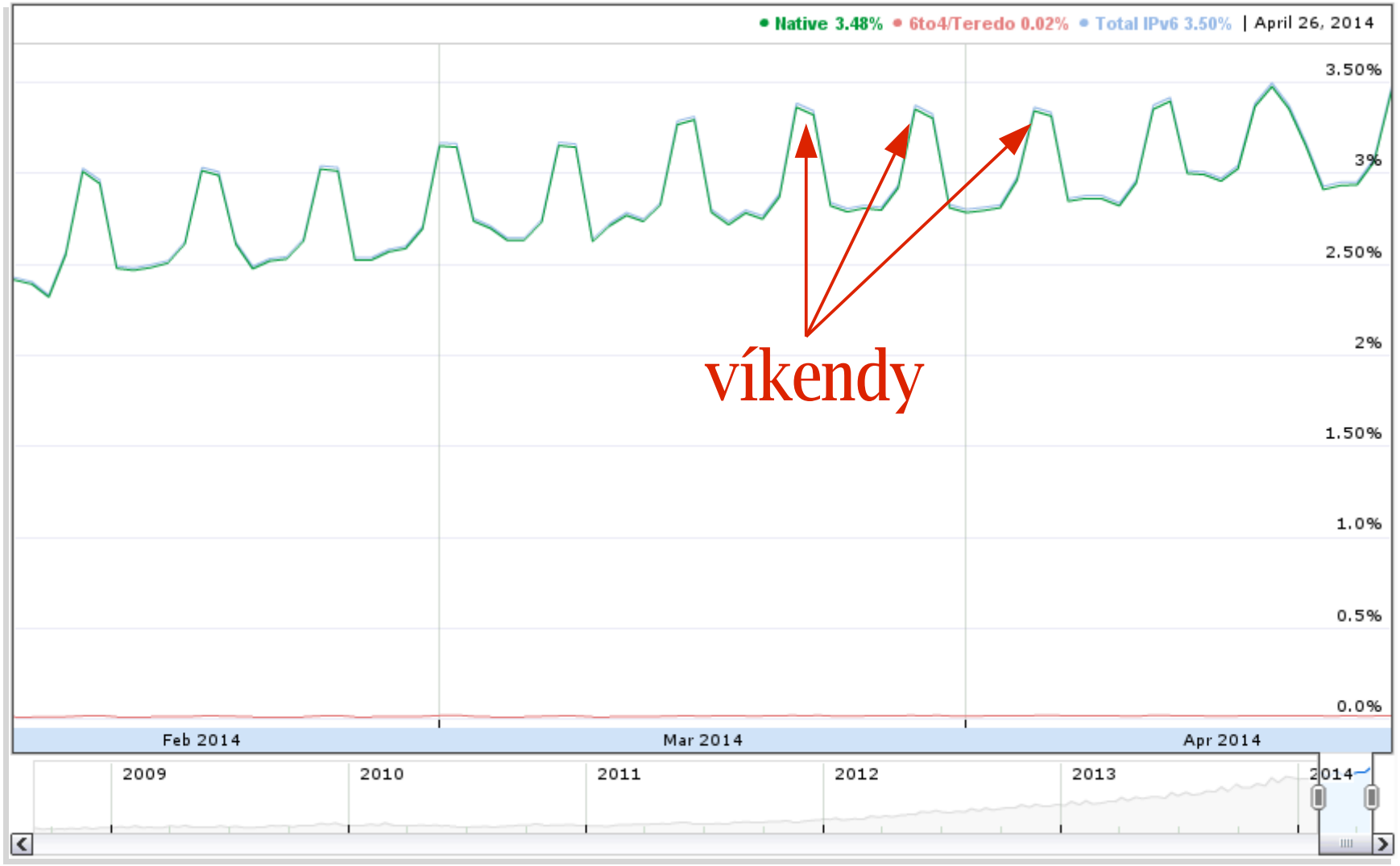


- IPv6 55 Gb/s, cca 1,7 % provozu

Statistika Google

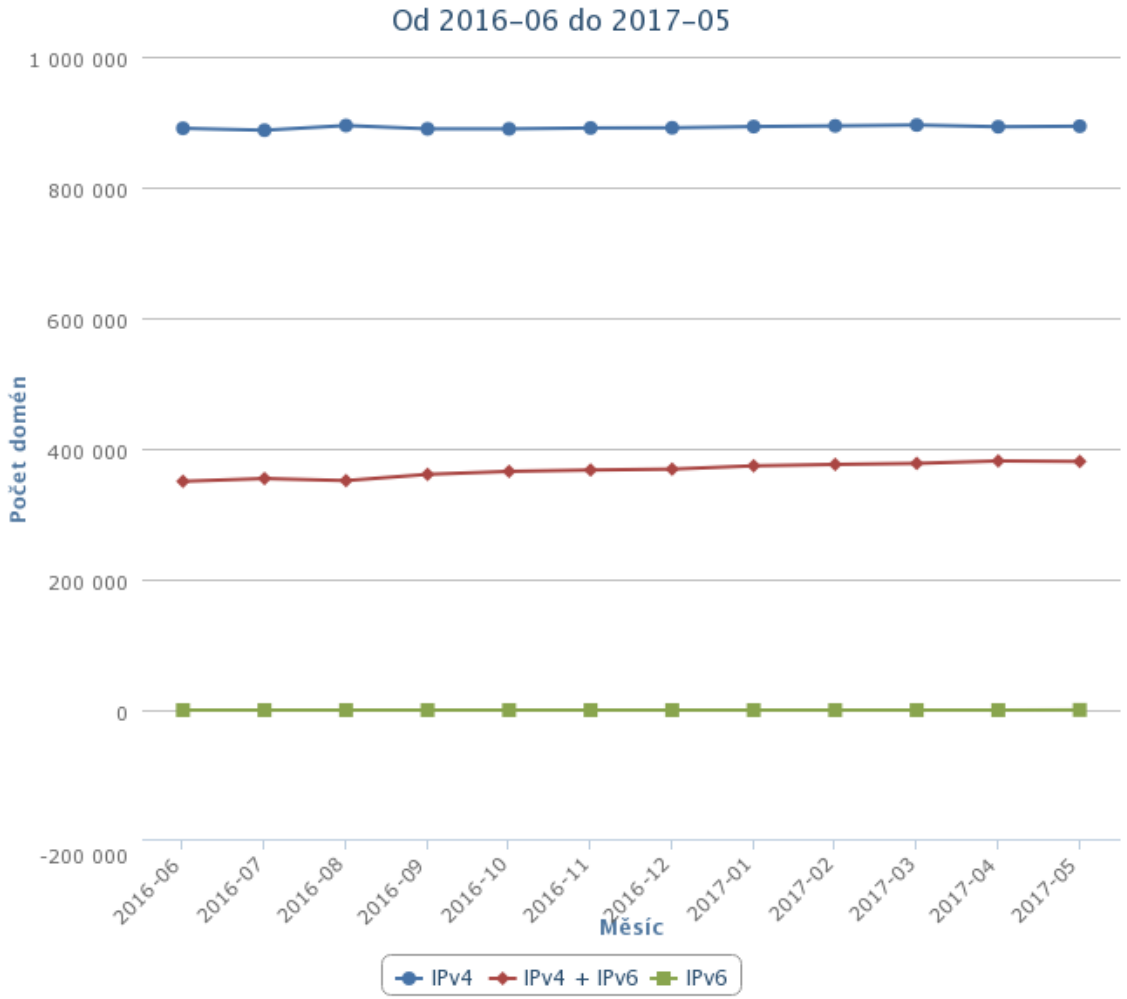


Google – detail



Statistiky CZ.NIC

- https://stats.nic.cz/stats/ipv6_domains/

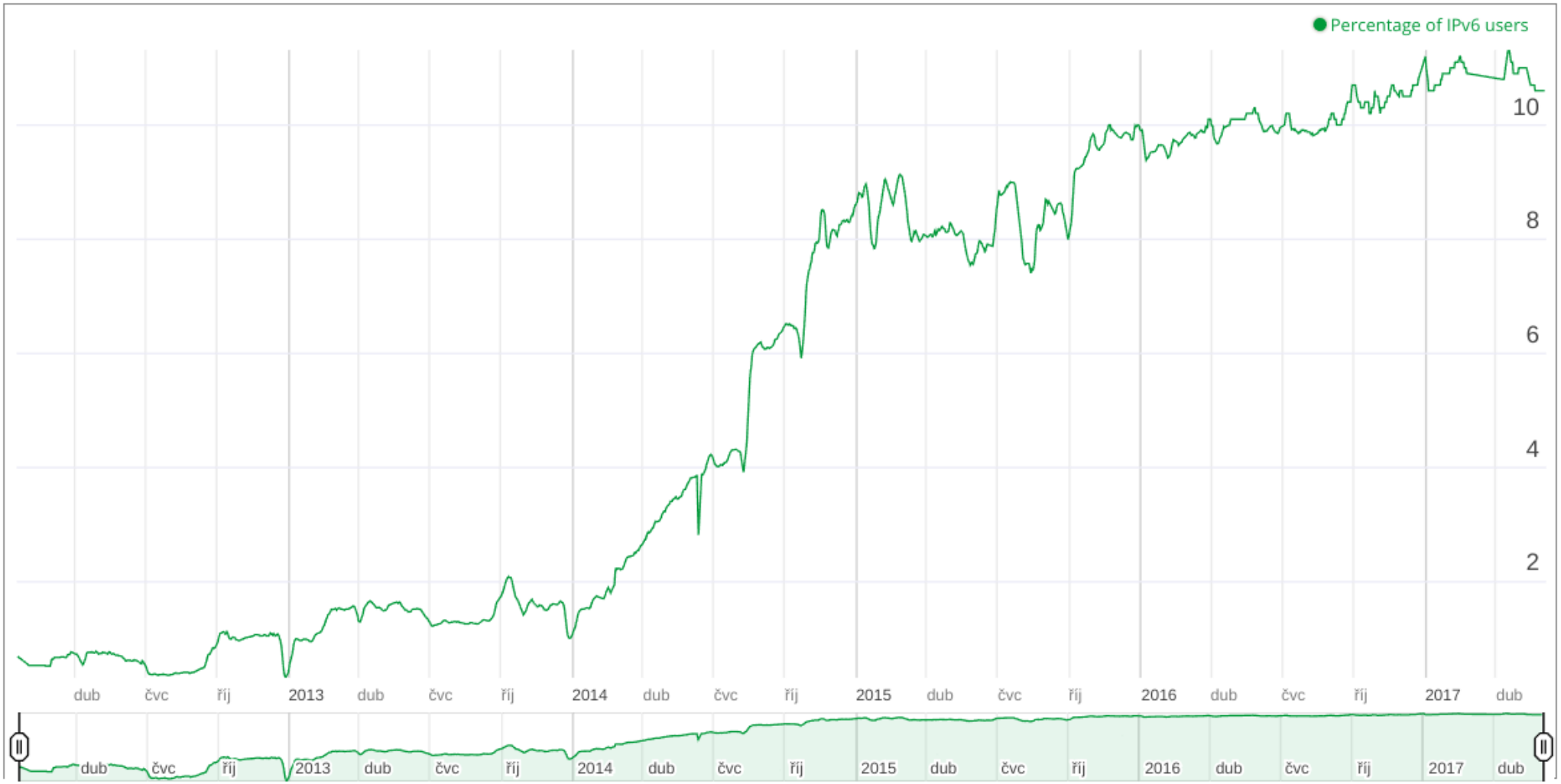


1,27 mil. domén .cz

381 tisíc (30 %)
podporuje IPv6

228 tisíc (18 %)
má IPv6 mail

Uživatelé v ČR



- 6lab.cisco.com

Děkuji za pozornost.

Dotazy?