Rodina protokolů TCP/IP
verze 3

Téma 4: Adresy a adresování v TCP/IP, IP adresy verze 4

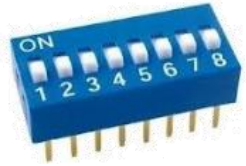
Jiří Peterka

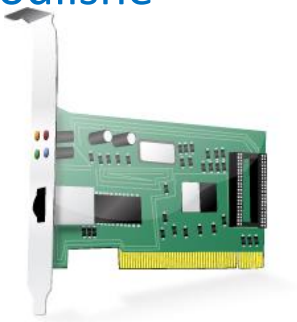
potřeba adresování

- adresování: možnost přidělit konkrétnímu objektu (vhodnou) adresu
  - za účelem jeho identifikace
- týká se objektů na různých úrovních abstrakce (vrstvách)
  - příklad: *když klikneme na nějaký (hypertextový) odkaz ….*
    - potřebujeme identifikovat cílový uzel, na kterém běží server
      - získat IP adresu
    - potřebujeme k němu dopravit svůj požadavek, přes mezilehlé uzly
      - IP adresy, linkové adresy uzlů
    - v rámci uzlu musíme identifikovat entitu, fungující jako (HTTP) server
      - přes odpovídající port
    - potřebujeme identifikovat požadovaný prvek
      - jméno a příponu objektu, jeho typ, ...
- důsledek: adresování se týká všech vrstev
  - vrstva síťového rozhraní
    - technologicky závislé linkové adresy
      - např. 48-bitové Ethernetové adresy)
      - identifikují uzel jako celek
  - síťová vrstva
    - abstraktní IP adresy
      - identifikují uzel jako celek
  - transportní vrstva
    - relativní adresy: čísla portů
      - identifikují entity v rámci uzlů
  - aplikační vrstva
    - URI/URL odkazy na konkrétní objekty

adresování na úrovni vrstvy síťového rozhraní

- obecně:
  - TCP/IP tuto vrstvu nepokrývá, na zde používaných adresách mu nezáleží
    - adresy, používané na vrstvě síťového rozhraní nepředepisuje ani neomezuje
      - někdy jsou tyto adresy označovány jako **HW adresy** (též: **linkové adresy**)
  - adresy vyšších vrstev (IP adresy na síťové vrstvě) jsou na těchto HW adresách nezávislé
    - platí pro IPv4, pro IPv6 vše trochu jinak
      - výhoda: vyšší vrstvy nepotřebují vědět, jak je řešena vrstva síťového rozhraní
    - alternativa: v IPX/SPX byly síťové adresy přímo závislé na linkových adresách
      - ukázalo se jako nepříliš šťastné řešení, následně změněno
- v praxi:
  - HW (linkové) adresy různých technologií mohou být diametrálně odlišné
    - Ethernet: 48-bitové adresy
      - celosvětově unikátní, každému rozhraní přiřazuje již výrobce
    - ARCNET: 8-bitové adresy
      - nastavuje uživatel na DIP switch-i
        - za unikátnost takto nastavených adres (v dané síti) odpovídá uživatel

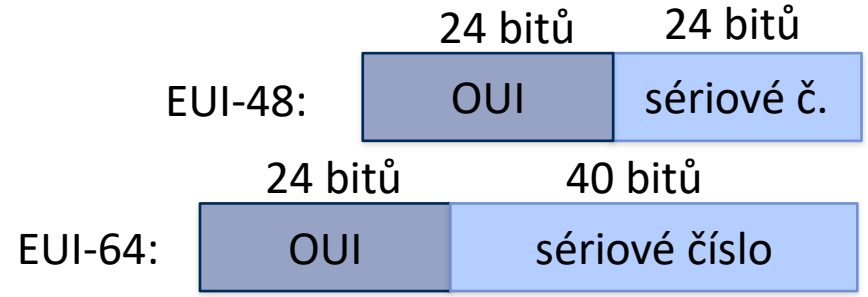




Ethernetové (MAC) adresy

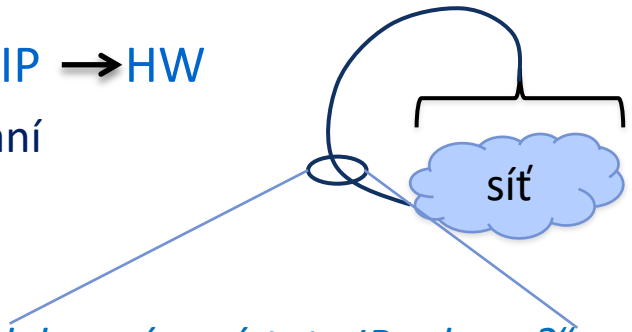
- mají 48 bitů
  - vnitřně jsou 2-složkové:
    - vyšší 3 byty představují identifikaci výrobce (OUI, Organizationally Unique Identifier)
      - jednotlivým výrobcům je přiděluje je organizace IEEE (Institute of Electrical and Electronics Engineers)
    - nižší 3 byty představují sériové číslo konkrétního síťového rozhraní
      - „výrobní číslo“
- jsou pevně nastaveny výrobcem a nedají se měnit
  - až na výjimky, kdy je změna možná
  - jsou (měly by být) celosvětově unikátní
- nevypovídají o umístění či příslušnosti uzlu do sítě !!!!

- používají se již na úrovni MAC podvrstvy
  - proto původní označení: **MAC 48**
    - správně by ale „MAC adresa“ měla být jakákoli adresa, používaná na podvrstvě MAC
  - nové označení: adresy **EUI-48**
- existují také 64-bitové Ethernetové adresy: EUI-64
  - mají větší část pro „sériové číslo“
    - a stejné (3-bytové) OUI
  - používají se např. v IPv6



převod mezi HW a IP adresami

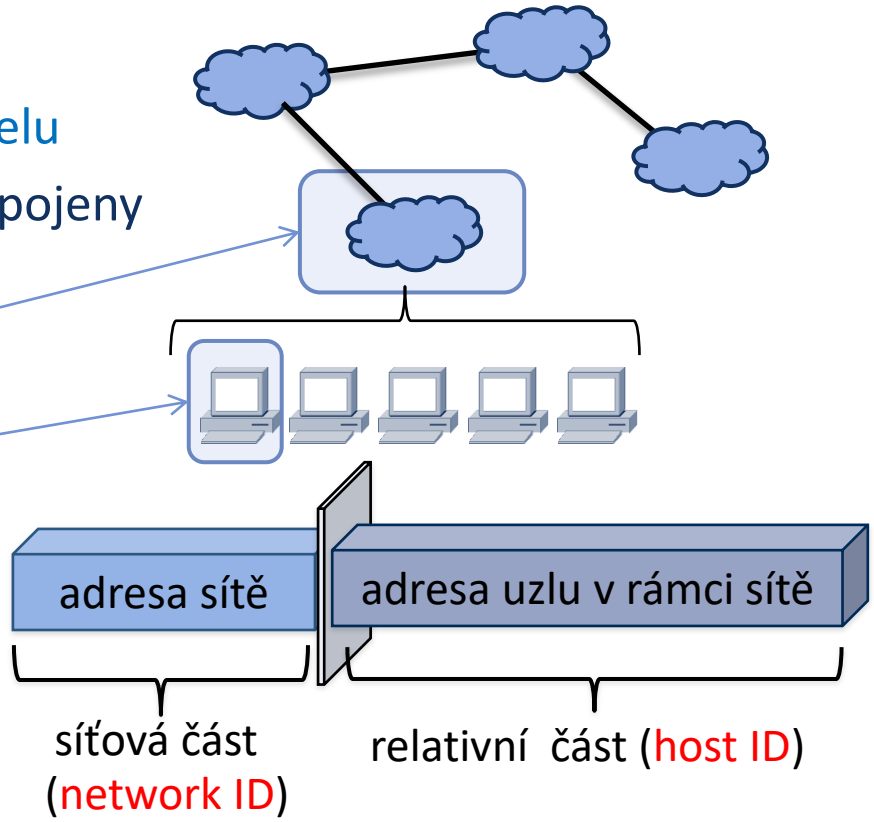
- TCP/IP potřebuje možnost převodu mezi HW adresami a IP adresami
- možnosti:
  - protokol ARP (Address Resolution Protocol): převod IP → HW
    - vyžaduje možnost broadcastu na vrstvě síťového rozhraní
      - nabízí například Ethernet
    - princip fungování:
      - ten, kdo zná IP adresu, rozešle broadcastem dotaz: *„kdo z vás má tuto IP adresu?“*
      - ten, kdo ji používá, odpoví a sdělí svou HW adresu

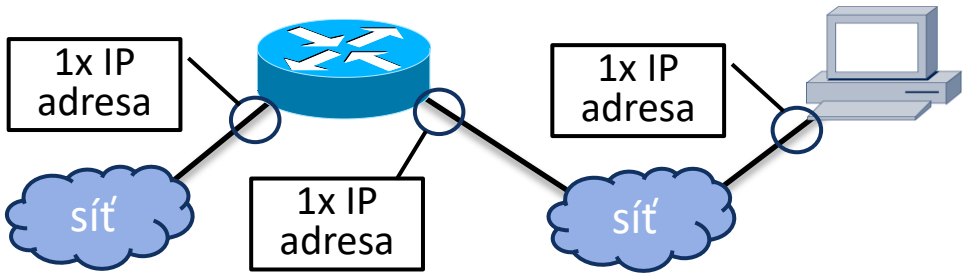


  - opačný převod (HW → IP) je vlastně přidělování IP adresy na základě HW adresy
    - používají se k tomu protokoly RARP (Reverse ARP), BootP, DHCP, ……..
- problém:
  - co když není k dispozici broadcast (na úrovni vrstvy síťového rozhraní)?
  - převod přes tabulku
    - „někdo“ (vhodný server) spravuje převodní tabulku, odpovídá na dotazy ohledně převodu
      - používá se u IP nad ATM (ATMARP)
  - převod přímým výpočtem
    - nelze např. u Ethernetu
    - podmínkou je možnost nastavení HW adresy
      - jako např. v ARCNETu

adresování na síťové vrstvě

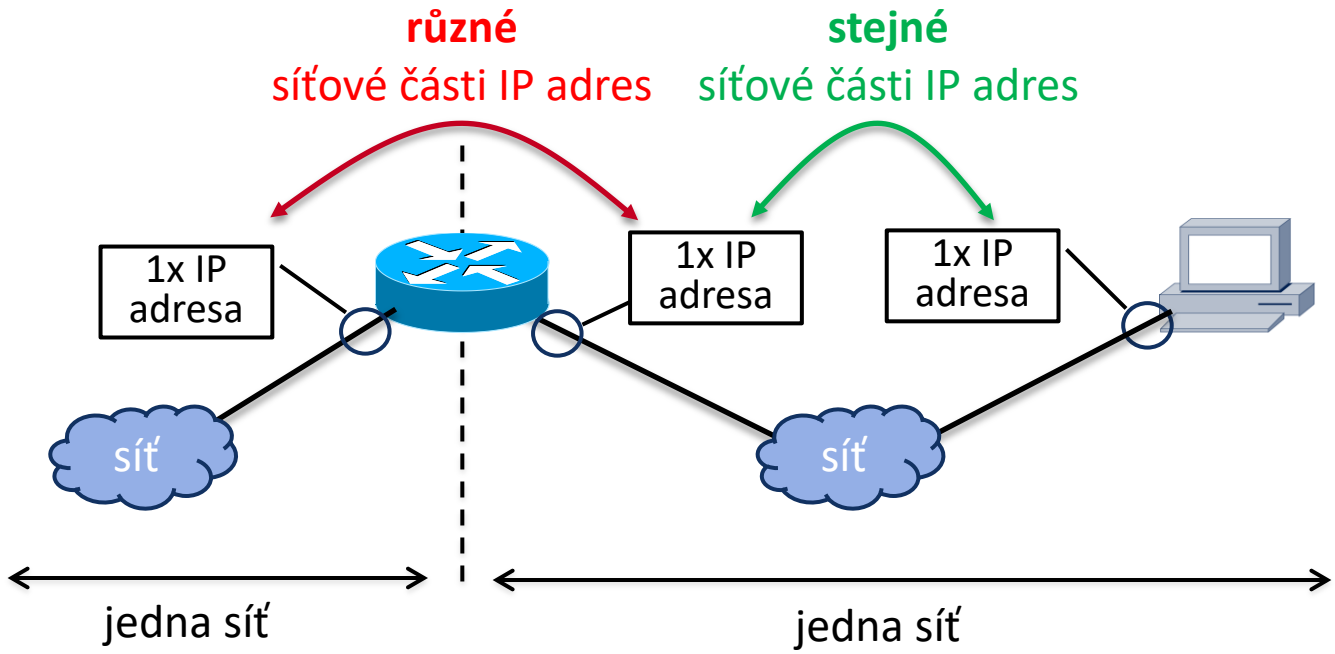
- adresují se celé uzly (uzly jako celky)
  - vychází se z představy katenetového modelu
    - že existuje více sítí a jsou vzájemně propojeny
- síťové adresy musí vyjadřovat:
  - příslušnost ke konkrétní síti
  - relativní adresu uzlu v rámci dané sítě
- proto:
  - síťové adresy (**IP adresy**) jsou 2-složkové
- přesněji:
  - IP adresy se přidělují síťovým rozhraním
    - koncové uzly (hosts) mají 1 rozhraní (1 IP adresu)
    - směrovače mají více síťových rozhraní (a na každém 1 IP adresu)
      - podobně multihomed uzly
  - IP adresy verze 4: celkem 32 bitů
  - IP adresy verze 6: celkem 128 bitů





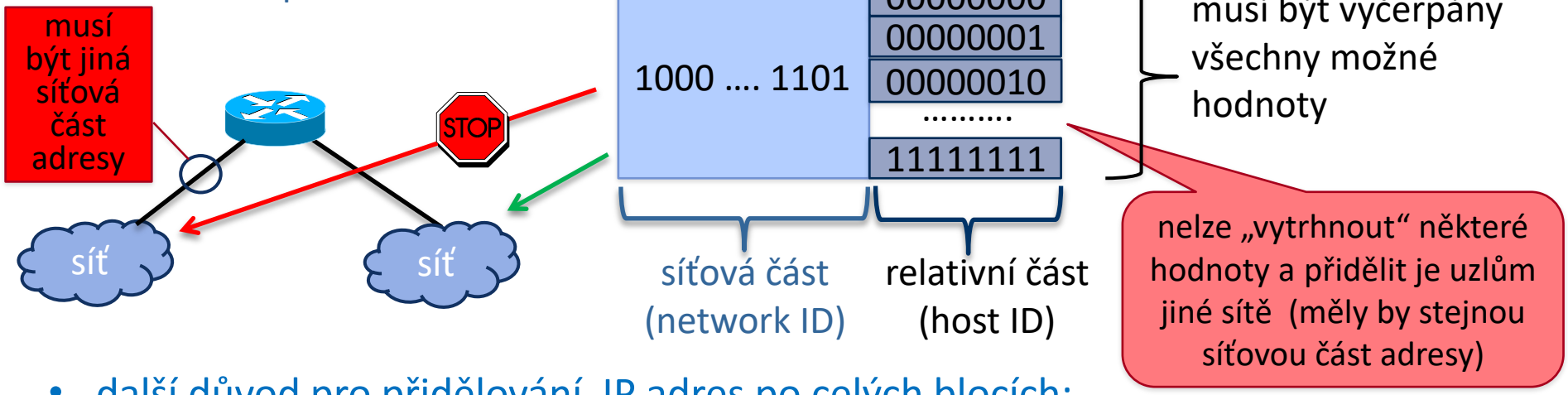
pravidla přidělování IP adres

- při přidělování IP adres konkrétním uzlům (jejich rozhraním) musí být dodržen význam obou jejich složek:
  - pokud se dvě síťová rozhraní nachází ve stejné síti, musí mít jejich IP adresy stejnou síťovou část (network ID)
    - a naopak různé relativní části (host ID)
  - pokud se dvě síťová rozhraní nachází v různých sítích, musí mít jejich IP adresy různou síťovou část svých adres (network ID)
    - zatímco jejich relativní části (host ID) mohou, ale nemusí být stejné.

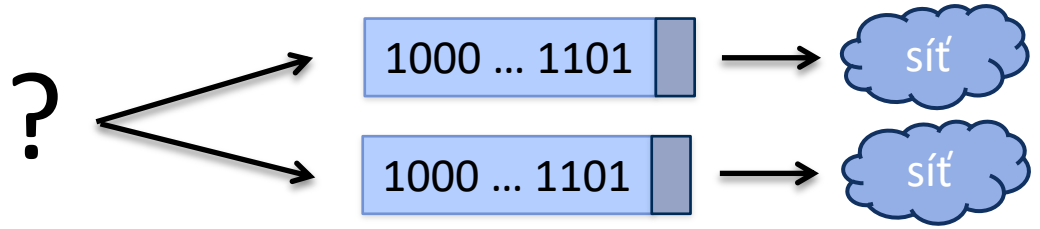


přidělování IP adres po blocích

- pravidla přidělování IP adres mají významný důsledek:
  - IP adresy se musí přidělovat po celých blocích !!!!
    - blok ve smyslu: všechny konkrétní IP adresy se stejnou síťovou částí
      - představa bloku:

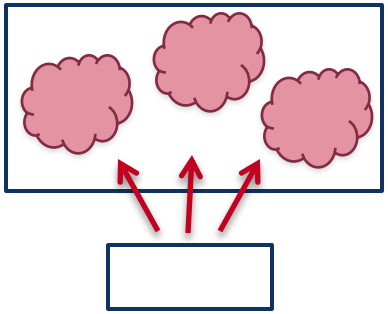


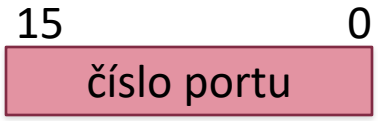
  - další důvod pro přidělování IP adres po celých blocích:
    - algoritmy směrování v TCP/IP se rozhodují pouze podle síťové části adresy
      - a teprve uvnitř cílové sítě berou v úvahu i relativní část IP adresy
    - pokud by různé sítě měly stejnou síťovou část adresy, algoritmy směrování by to zmátlo ….



adresování na transportní vrstvě

- připomenutí:
  - na síťové vrstvě (i na vrstvě síťového rozhraní) se adresují uzly jako celky
    - příslušné adresy nedokáží rozlišit různé entity v rámci téhož uzlu
- transportní vrstva:
  - adresování již potřebuje rozlišit různé entity v rámci daného uzlu
    - ale zase nepotřebuje identifikovat uzel jako celek
  - proto na transportní vrstvě stačí jen relativní adresy: **porty**
    - jde o celá čísla v rozsahu 0 až 65535 (tj. 16 bitů)

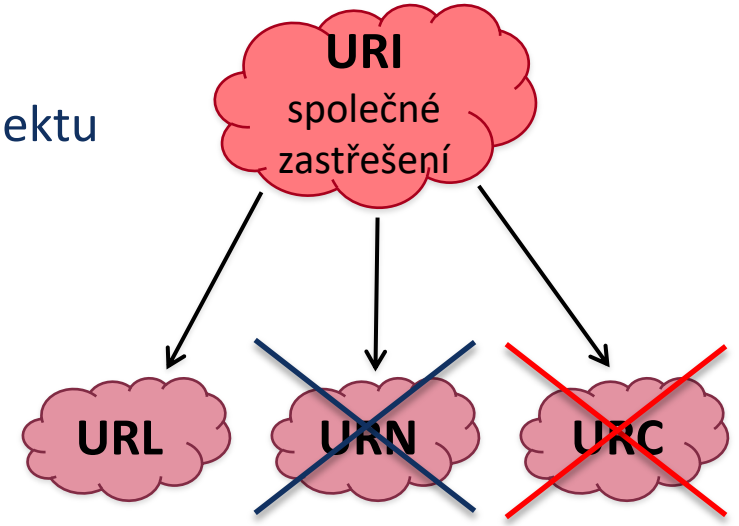




- porty jsou abstraktní (logické) adresy
  - zatímco „fyzické“ entity se musí „asociovat“ (bind) s konkrétními porty
- konvence o „dobře známých portech“ (well known ports)
  - porty 0 až 1023, s pevně daným významem
    - konvenci o dobře známých portech udržuje organizace IANA
    - publikuje ji na webu: `http://www.iana.org/assignments/port-numbers`

adresování na aplikační vrstvě

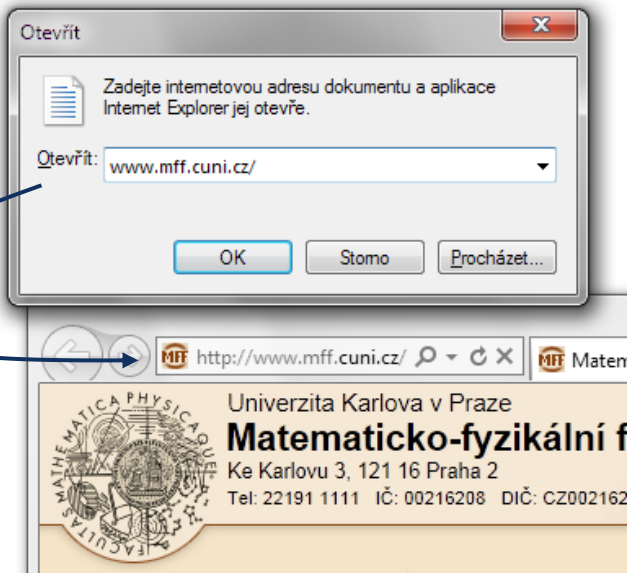
- je třeba identifikovat různé typy objektů (texty, obrázky, videa, ….)
  - které se mohou nacházet na různých místech v síti (uzlech)
    - mají obvykle formu souborů (případně proudů / streamů)
- možnosti adresování:
  - nezávisle na umístění objektu
    - identifikuje se objekt jako takový, nezávisle na jeho umístění
      - v praxi: např. ISBN (Internation Standard Book Number) pro identifikaci knih
      - v TCP/IP: **URN** (Uniform Resource Name)
  - v závislosti na umístění objektu
    - součástí identifikace (adresy) je i umístění objektu
      - v TCP/IP: **URL** (Uniform Resource Locator)
  - „nějak jinak“
    - například přes metadata, přes citace apod.
      - v TCP/IP: **URC** (Uniform Resouce Citation)
        - nebylo ale nikdy standardizováno



neformálně: URI = URL
formálně: správně je URI (URL je neformální, ale používá se více)

URI schémata

- identifikátory URI (URL) mají obecnou strukturu
  - ta vychází z tzv. **schémat** (angl: scheme)
    - schéma = varianta zápisu URI identifikátoru
      - vždy začíná jménem schématu a pokračuje specifickou částí
  - obecný tvar je `<schéma>:<specifická část>`
- příklady:
  - HTTP schéma: http://www.earchiv.cz
  - FTP schéma: ftp://sunsite.mff.cuni.cz/Network/RFCs/rfc-index-latest.txt
  - mailto schéma: mailto:jiri@peterka.cz



browser si sám doplnil (předpokládané) schéma

schéma	využití pro:	definuje:
file	Host-specific file names	RFC1738
ftp	File Transfer Protocol	RFC1738
http	Hypertext Transfer Protocol	RFC2616
https	Hypertext Transfer Protocol Secure	RFC2818
Imap	internet message access protocol	RFC5092
mailto	Electronic mail address	RFC2368
news	USENET news	
sip	Session Initiation Protocol	RFC3261
snmp	Simple Network Management Protocol	

HTTP a mailto schéma

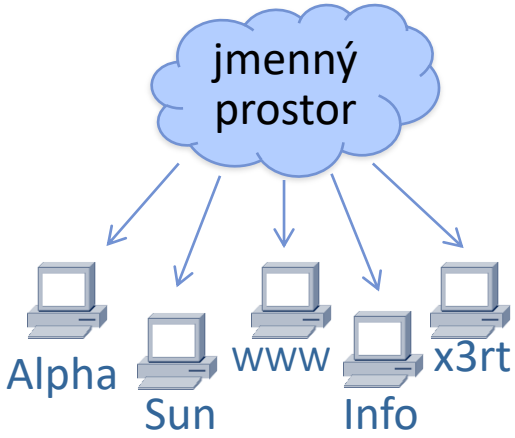
- HTTP schéma: obecný tvar specifické části je:

  `//<user>:<password>@<host>:<port>/<url-path>?<query>#<bookmark>`

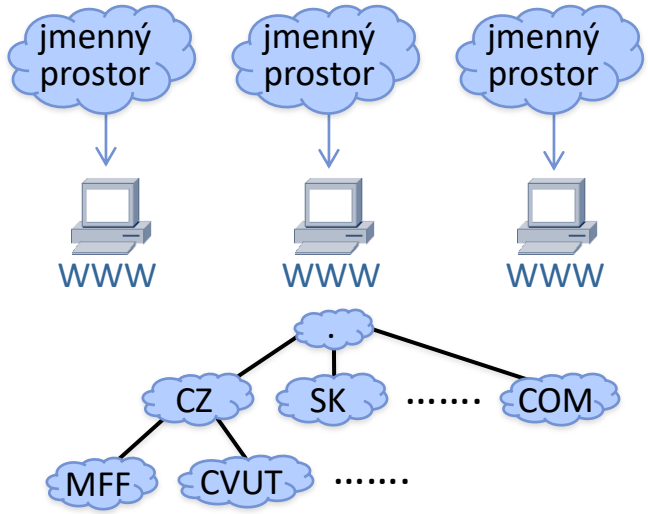
  - *umožňuje vyjádřit například:*
    - *jméno a heslo (pokud jsou zapotřebí)*
      - *například: http://**jmeno:heslo**@www.nic.cz/chranena_stranka*
    - *číslo portu (nutné uvádět jen tehdy, pokud se liší od „dobře známého“ portu 80)*
      - *například: http://novyweb.nic.cz**:8080***
    - *dotaz (například: vyhledávací)*
      - *například: http://www.google.cz**?q=MFF***
    - *záložku (bookmark) v rámci WWW stránky*
      - *například: http://www.earchiv.cz/a98/a816k180.php3#tld*
- *mailto schéma (RFC 2368):*
  - *v nejjednodušší podobě: mailto:jiri@peterka.cz*
  - *ale může obsahovat i předmět a tělo zprávy, stejně jako výčet dalších příjemců*
    - *například:* `mailto:jan@novak.cz?cc=josef@novotny.cz&subject=Pozdrav z prednasky&body=Dobry den, posilam pozdrav z prednasky o TCP/IP`

symbolická jména vs. číselné adresy

- nejjednodušší varianta URI (URL) odkazu je
  - <schema>://<host>
    - *kde <host> může být vyjádřen:*
      - *číselnou IP adresou, nebo*
      - *symbolickým (doménovým) jménem*

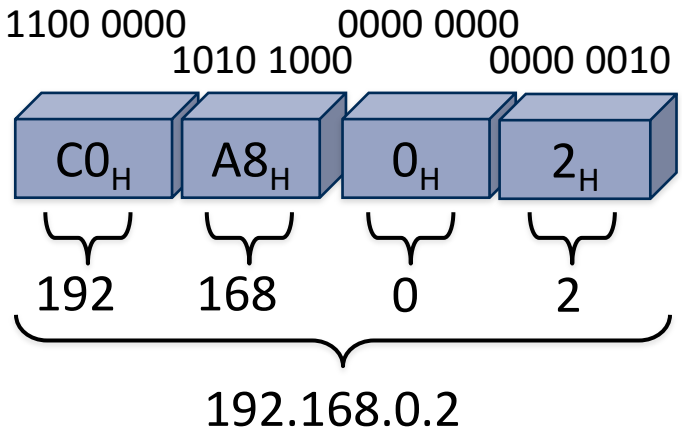


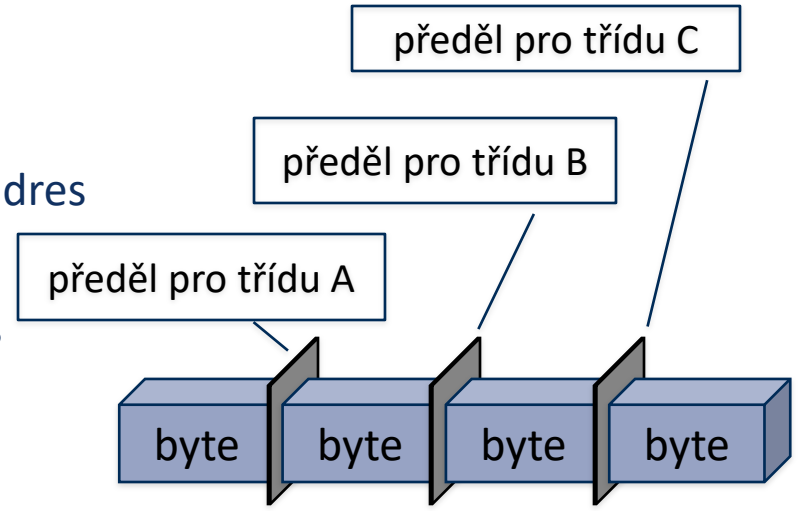
- symbolická jména mohou být vybírána:
  - z jednoho (plochého) jmenného prostoru
    - původně použito v ARPANETu, později v síti EARN/Bitnet
      - dnes v ČR: u datových schránek
    - nevýhoda: jednoduchá a snadno zapamatovatelná jména se brzy vyčerpají
  - z více jmenných prostorů
    - výhoda: stejné jméno lze přidělit opakovaně
      - podmínkou je možnost odlišit stejná jména, přidělená v různých jmenných prostorech
    - v TCP/IP se řeší pomocí hierarchického uspořádání jmenných prostorů (domén)
      - v rámci systému DNS (Domain Name System)



IP adresy verze 4

- jsou abstraktní
  - nemají přímou vazbu na HW adresy (adresy vrstvy síťového rozhraní)
    - proto jsou nutné možnosti překladu – ARP, tabulka, výpočet …..
- mají rozsah 32 bitů
  - jejich hodnoty se zapisují dekadicky, po bytech
- bylo nutné určit velikost bloků
  - po kterých se budou IP adresy přidělovat
  - zvolené řešení:
    - budou 3 různé pozice „předělů“
      - mezi síťovou a relativní částí IPv4 adresy
    - odpovídá to 3 různým **třídám IPv4 adres**
      - třída A, pro největší sítě
        - blok má velikost 2^{3*8} (16 777 216) IPv4 adres
      - třída B, pro středně velké sítě
        - blok má velikost 2^{2*8} (65 536) IPv4 adres
      - třída C, pro malé sítě
        - blok má velikost 2^{1*8} (256) IPv4 adres

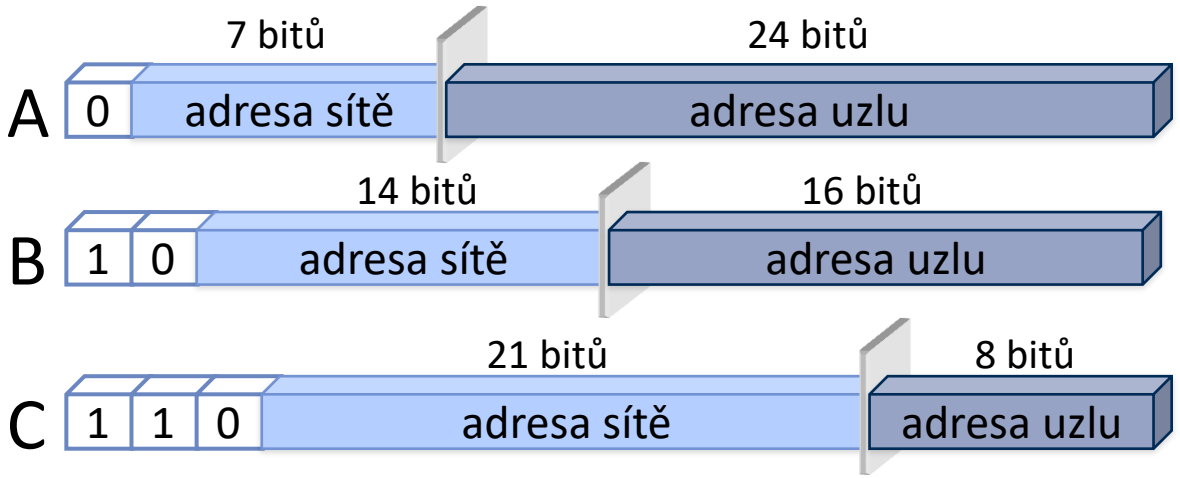




třídy IPv4 adres: A, B a C

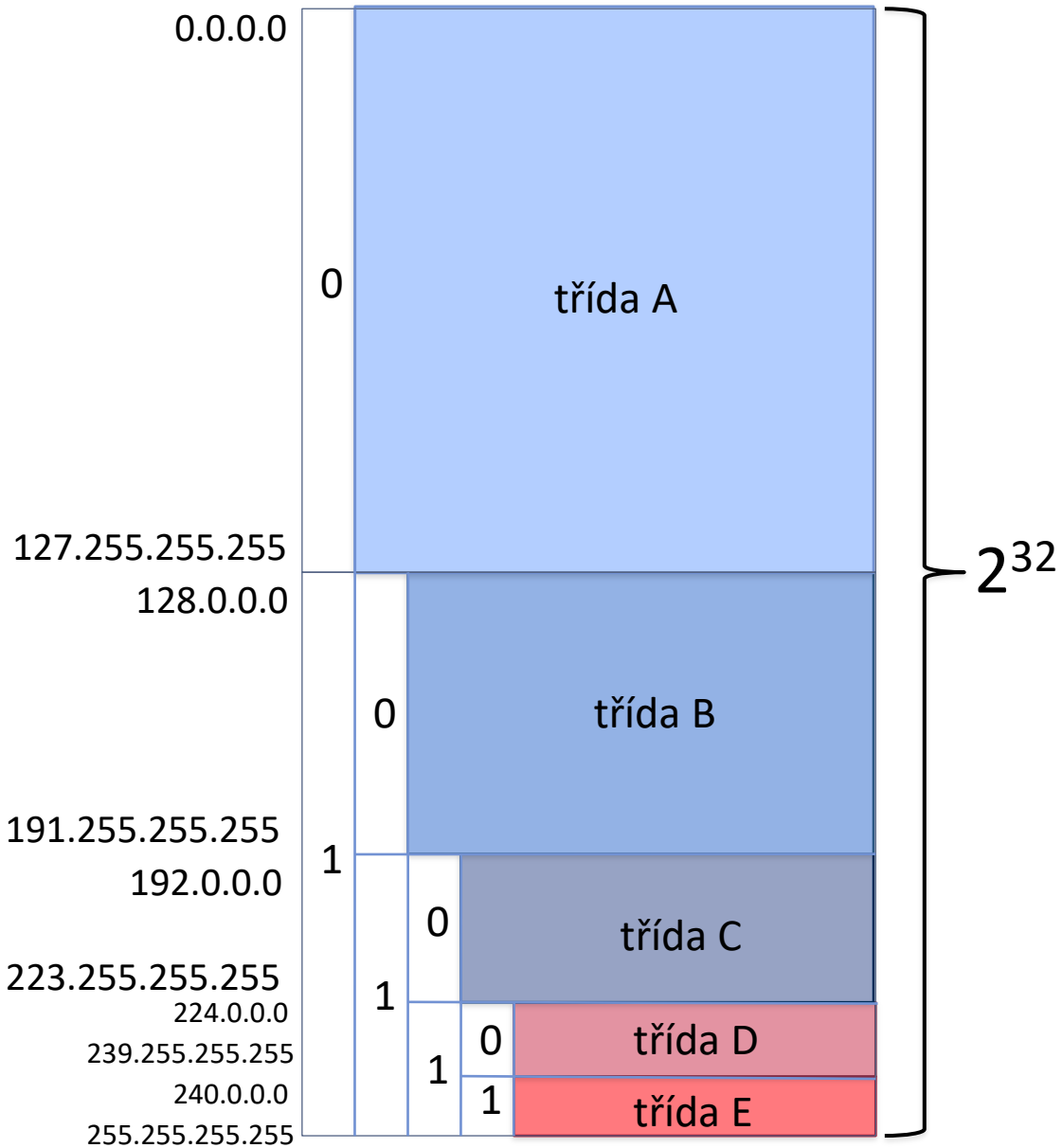
- rozlišení jednotlivých tříd IPv4 adres bylo původně pevně dáno:
  - třída A: nejvyšší bit je 0 (pro síťovou část adresy zbývá 7 bitů)
    - bloků je celkem $2^7 = 128$ (každý o velikosti $2^{24} = 16777216$ individuálních IPv4 adres)
      - ale oba „krajní“ mají speciální význam – reálně pouze 126 bloků IPv4 adres třídy A
      - jde konkrétně o IPv4 adresy v rozsahu 1.0.0.0 do 126.255.255.255
  - třída B: nejvyšší bity jsou 1 a 0
    - bloků je celkem $2^{14} = 16\,384$ (každý o velikosti $2^{16} = 65536$ individuálních IPv4 adres)
      - jde konkrétně o IPv4 adresy v rozsahu 128.0.0.0 do 191.255.255.255
  - třída C: nejvyšší bity jsou 1, 1 a 0
    - bloků je celkem $2^{21} = 2\,097\,152$ (každý o velikosti $2^8 = 256$ individuálních IPv4 adres)
      - jde konkrétně o IPv4 adresy v rozsahu 192.0.0.0 do 223.255.255.255

- tím ale není rozsah IPv4 adres úplně vyčerpán
  - ještě zbývá prostor pro adresy tříd D a E
    - se speciálním významem



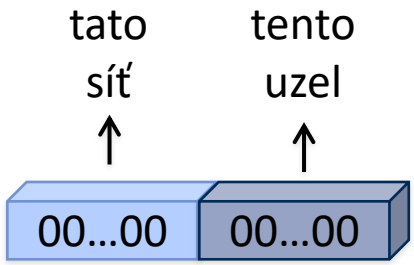
rozdělení prostoru IPv4 adres

- adresy třídy A: polovina
  - zabírají polovinu celého adresového prostoru IPv4
- adresy třídy B: čtvrtina
  - zabírají čtvrtinu celého adresového prostoru IPv4
- adresy třídy C: osmina
- adresy tříd D a E:
  - dělí se o zbývající osminu
    - třída D jsou adresy pro multicast
      - nemají 2 logické složky
    - třída E je určena pro budoucí rozšíření
      - ale nikdy tak nebyla využita

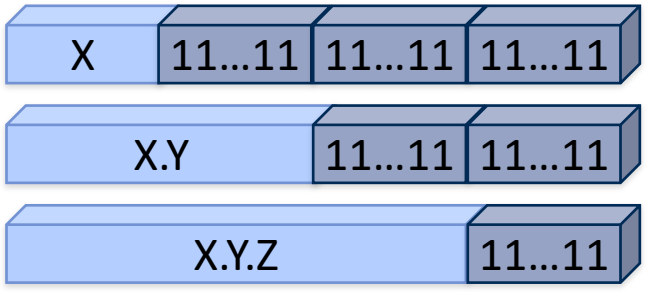


„speciální“ IPv4 adresy

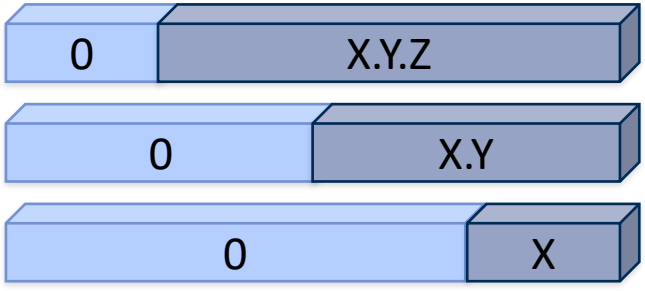
- některé IPv4 adresy mají speciální význam:
- základní princip:
  - samé 0 = „this“ (“tento”), též: the default, the current
    - vztahuje se k tomu, co je takto nahrazeno
  - samé 1 = „all“ („všechno“)
    - vztahuje se ke všem (existujícím) prvkům, které jsou takto nahrazeny



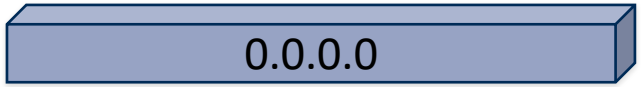
„tato“ konkrétní síť (X, X.Y, resp. X.Y.Z)



cílený broadcast (všechny uzly v sítích X, X.Y, resp. X.Y.Z)

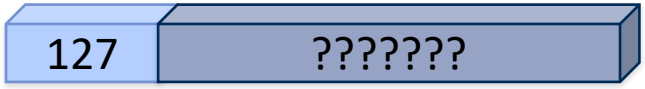


konkrétní uzel (X, X.Y, resp. X.Y.Z) v “této“ síti



„já“ (uzel který nezná svou IP adresu)

„omezený broadcast“

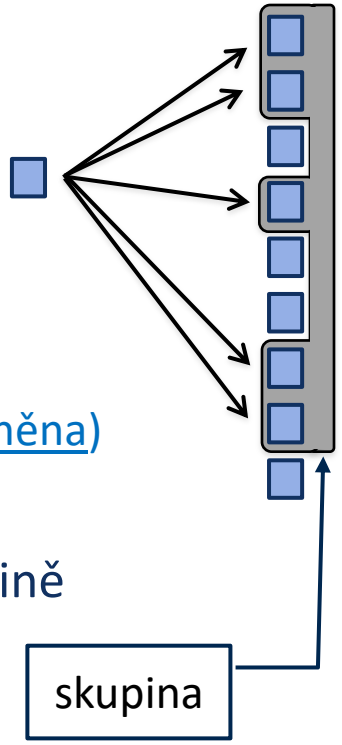


loopback

(broadcast „v této síti“ , je omezen jen na danou síť, směrovače jej nepropouští do dalších sítí)

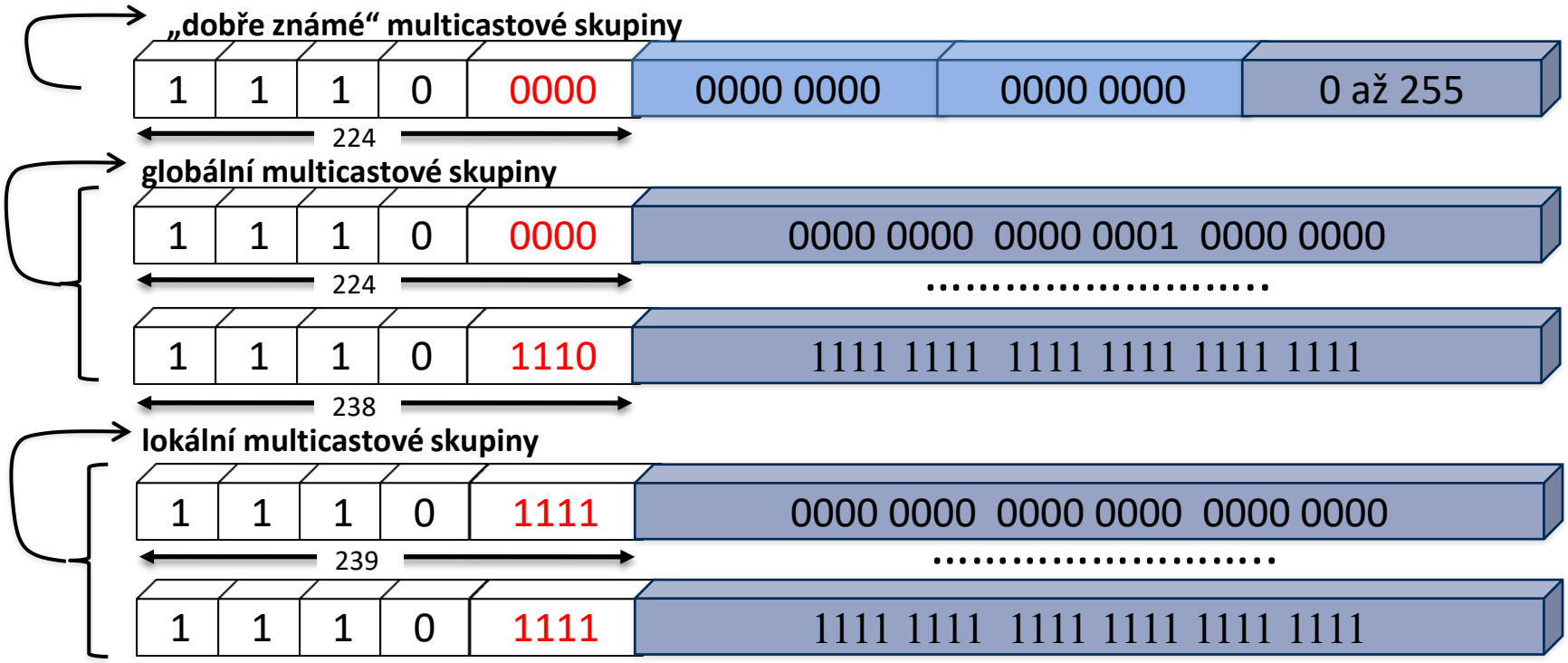
IPv4 adresy pro multicast (třída D)

- multicast = přenos od 1 zdroje k více příjemcům současně
  - unicast: od 1 k 1, broadcast: od 1 ke všem
- multicastová skupina:
  - *„ta skupina uzlů, která přijímá vysílání příslušného zdroje“*
    - tato skupina je adresována (jednou) multicastovou IP adresou
      - která nepotřebuje dělení na síťovou a relativní část (není logicky členěna)
  - multicastová skupina může být vymezena:
    - staticky: je dopředu a pevně definováno, které uzly jsou ve skupině
      - též: „dobře známé“ multicastové skupiny
      - příklady:
        - **224.0.0.1** („all hosts“, neboli: všechny uzly v dané síti)
        - **224.0.0.2** („all routers“, neboli: všechny směrovače v dané síti)
    - dynamicky: uzly se zařazují / vyjímají ze skupiny dle potřeby
      - musí existovat nástroje pro takovéto zařazování/vyjímání (správu skupin)
      - dělí se dále na:
        - lokální (dynamické) multicastové skupiny: jen pro uzly ze stejné sítě
        - globální (dynamické) multicastové skupiny: i pro uzly z různých sítí



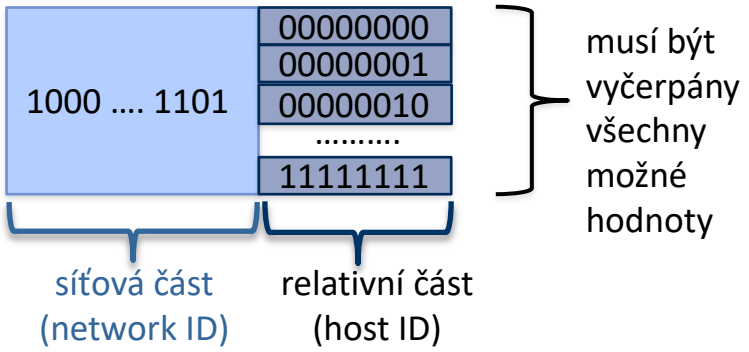
IPv4 adresy pro multicast (třída D)

- třída D IPv4 adres je vnitřně členěna:
  - prvních 256 adres (224.0.0.x) je vyhrazeno pro „dobře známé“ skupiny, viz:
    - **224.0.0.1** („all hosts“, neboli: všechny uzly v dané síti)
    - **224.0.0.2** („all routers“, neboli: všechny směrovače v dané síti)
  - posledních 2^{24} adres (239.x.x.x) je určeno pro lokální multicastové skupiny
  - ostatní adresy (224.0.1.0 až 238.255.255.255) jsou určeny pro globální multicastové skupiny



přidělování IPv4 adres

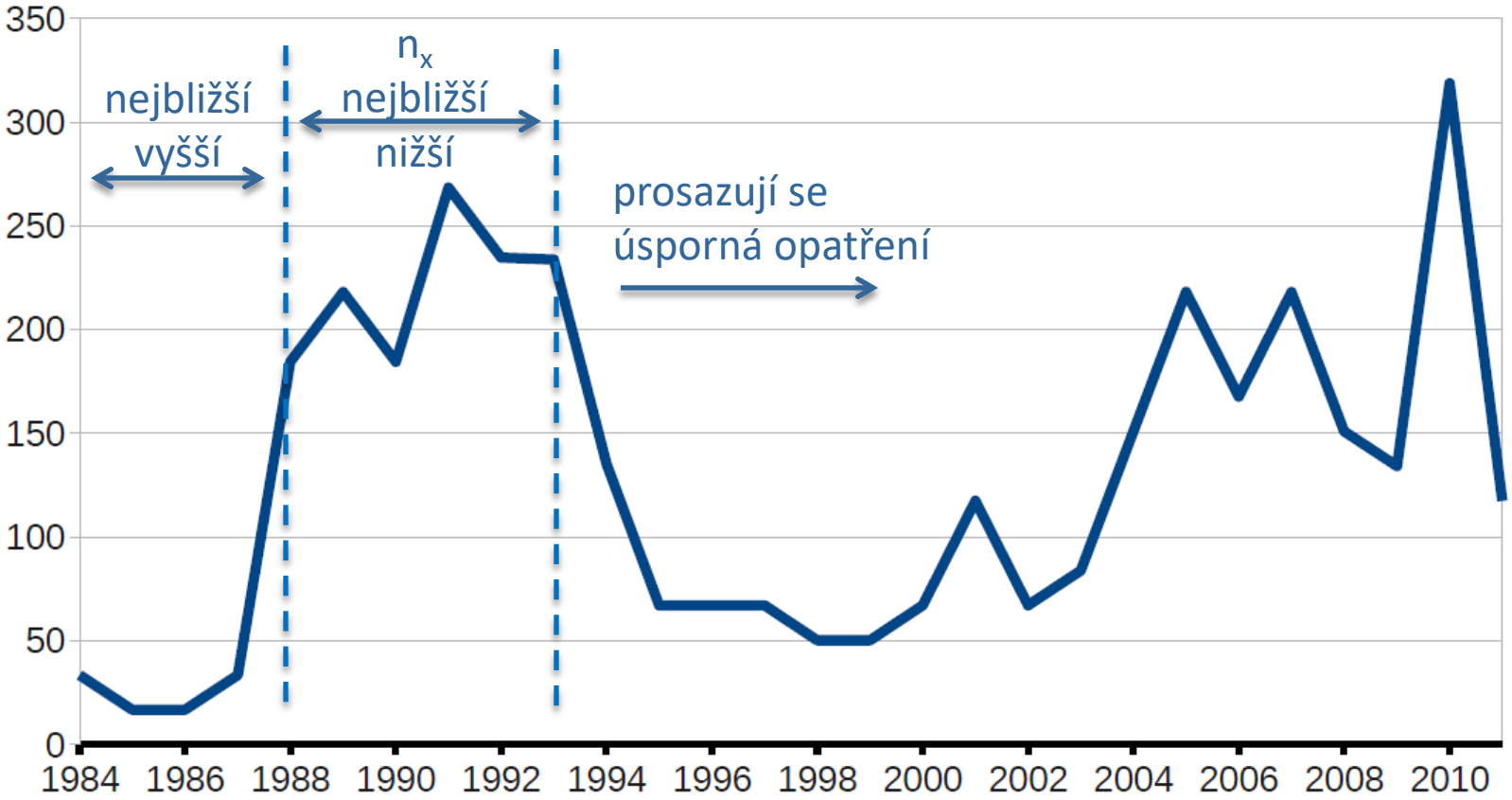
- připomenutí:
  - IP adresy se vždy přidělují po celých blocích
  - blok = všechny IP adresy se stejnou síťovou částí



- třídy IPv4 adres: A, B a C
  - blokem je vždy „celá síťová adresa“ příslušné třídy, jsou jen 3 možnosti:
    - 1 síťová adresa třídy C = 256 (2^8) individuálních IPv4 adres (1 tzv. C-čko)
    - 1 síťová adresa třídy B = 65536 (2^{16}) individuálních IPv4 adres (1 tzv. B-čko)
    - 1 síťová adresa třídy A = 16777216 (2^{24}) individuálních IPv4 adres (1 tzv. A-čko)
- prvotní způsob přidělování IPv4 adres:
  - zájemce s potřebou X adres dostal **„nejbližší vyšší blok“**
    - například: když potřeboval 1000 adres, dostal 1x B-čko (tj. 65536 IPv4 adres)
      - fakticky tak využil jen 1,52% přidělených adres, zbytek zůstal nevyužit !!
  - představovalo to velké plýtvání
    - a začalo hrozit rychlým vyčerpáním celého 32-bitového prostoru IPv4 adres !!
  - proto se přešlo na princip **„více nejbližších menších bloků“**
    - při potřebě 1000 adres dostal zájemce 4 nebo 8 C-ček (4-8x 256 IPv4 adres)

rychlost vyčerpávání IPv4 adres

- způsob přidělování IPv4 adres měl zásadní vliv na rychlost vyčerpávání celého 32-bitového prostoru IPv4 adres
  - princip „nejbližší vyšší“ se používal zhruba do roku 1988 – rychlý úbytek
  - princip „vícekrát nejbližší nižší“ se používal zhruba do roku 1993/4 – zpomalení



- rychlost vyčerpávání IPv4 adres si vynutila výrazná úsporná opatření !!!

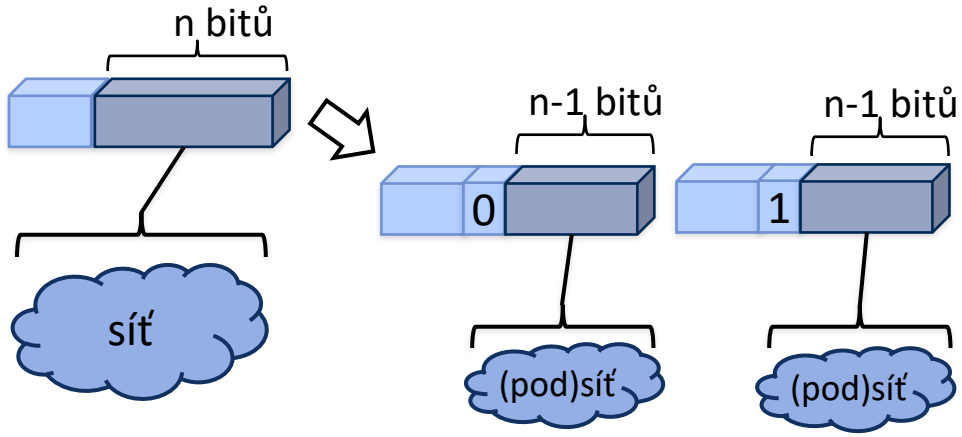
dočasné a trvalé řešení

- hrozba rychlého vyčerpání IPv4 adres si vynutila dvě různá řešení:
- dočasná řešení
  - pro zpomalení rychlého úbytku adres
  - konkrétně:
    - subnetting (RFC 950, 1985)
      - možnost rozdělit si blok IPv4 adres na menší části, pro více sítí
    - CIDR (RFC 1518, září 1993)
      - zrušení tříd (A,B a C)
      - možnost volit velikost přidělovaného bloku „libovolně“
    - privátní IPv4 adresy
      - možnost opakovaného použití stejných IPv4 adres (v privátních sítích)
      - využívá mechanismus překladu IP adres
        - NAT, Network Address Translation, RFC 1631, květen 1994
- trvalé řešení
  - zvětšení celého adresového prostoru
  - začal se hledat protokol IPng
    - IP next generation
    - práce zahájeny v H2 1992
      - nejprve 4 návrhy
      - 12/1992: další 3 návrhy
    - v červenci 1994 vybrán vítězný návrh
      - který se stává protokolem IP verze 6
    - začínají práce na „implementaci“ IPv6
      - řeší se otázky koexistence, přechodu na IPv6, distribuce IPv6 adres atd.
    - na přelomu 1995/1996 zveřejněna specifikace IPv6
      - RFC 1883 a další

IPv4 adresy byly vyčerpány (na úrovni IANA) v únoru 2011

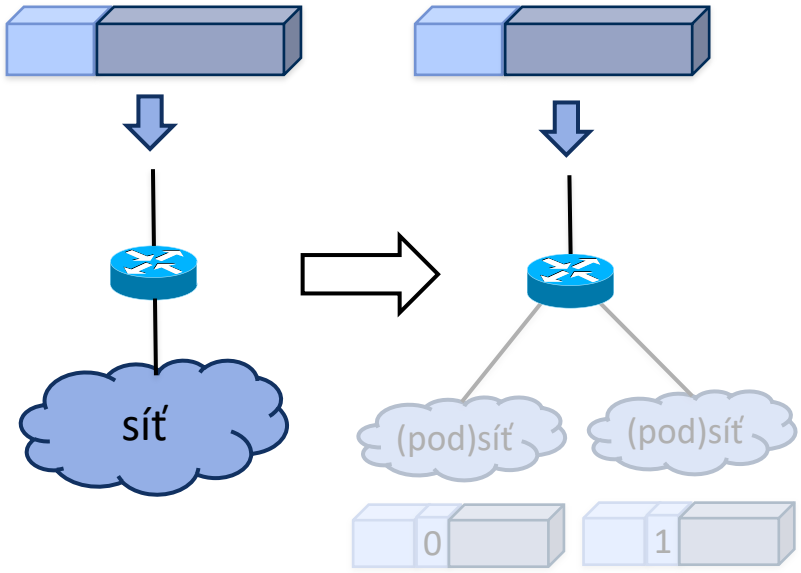
subnetting

- otázka:
  - co brání tomu, aby někdo dostal určitý blok IP adres a sám si ho rozdělil na menší bloky?
    - a tyto menší bloky přidělil různým (pod)sítím, alias: subnet-ům?



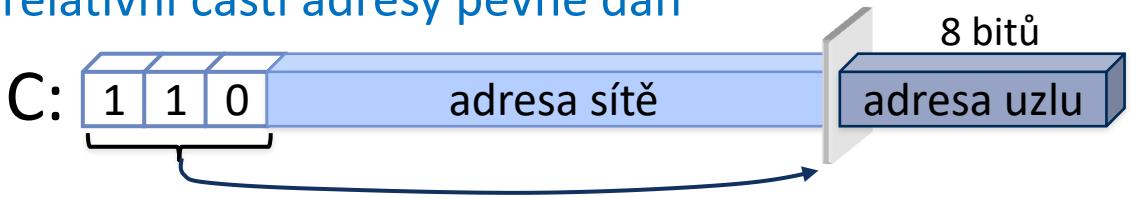
- odpověď:
  - ostatní nebudou vědět, jak to udělal!
    - a nebudou vědět, kudy směrovat data k jednotlivým (pod)sítím !

- řešení:
  - *můžete si sami rozdělit blok IP adres, podle svého uvážení – ale „zvenku“ to nesmí být vidět*
    - z pohledu ostatních sítí to nesmí být patrné
      - rozdělení lze udělat jen v soustavě (pod)sítí, které má pouze 1 vstupní bod

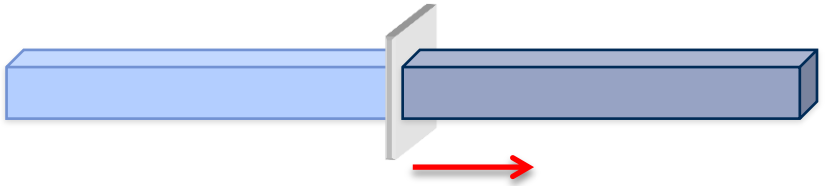


subnetting a síťové masky

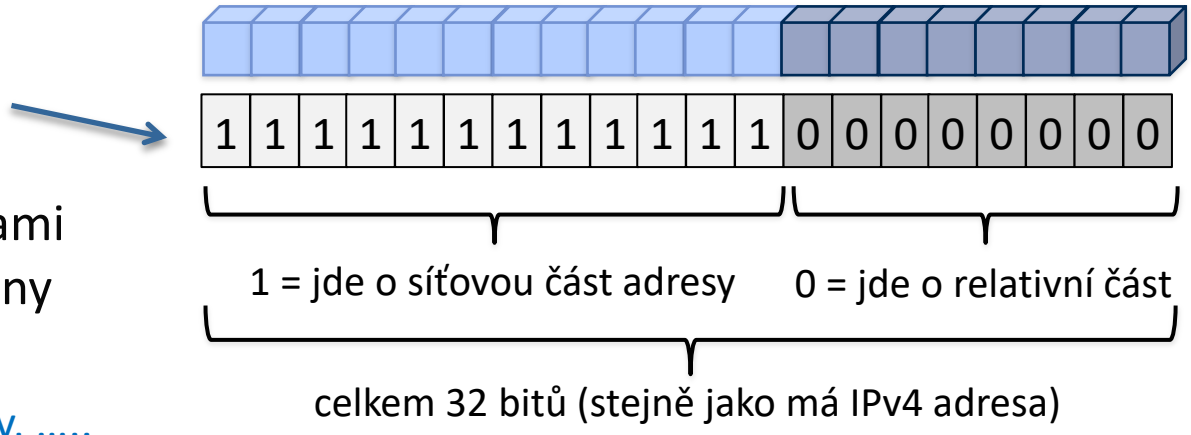
- subnetting jako první „popřel“ rozdělení IPv4 adres do tříd
  - kdy je předěl mezi síťovou a relativní částí adresy pevně dán
    - několika prvními bity



- podstata subnettingu:
  - předěl mezi síťovou a relativní částí IPv4 adresy se posouvá doprava
    - směrem k nižším bitům (ty se jakoby přidávají k síťové části adresy)
      - tj. už není pevně dán



  - problém:
    - jak se v rámci soustavy subnetů pozná, kam byl předěl posunut?
  - řešení:
    - zavedly se síťové masky
  - důsledek:
    - práci se síťovými maskami musí podporovat všechny síťové prvky i SW
      - směrovače, síťové karty, …..



příklad subnettingu

192.168.1	00000000-11111111

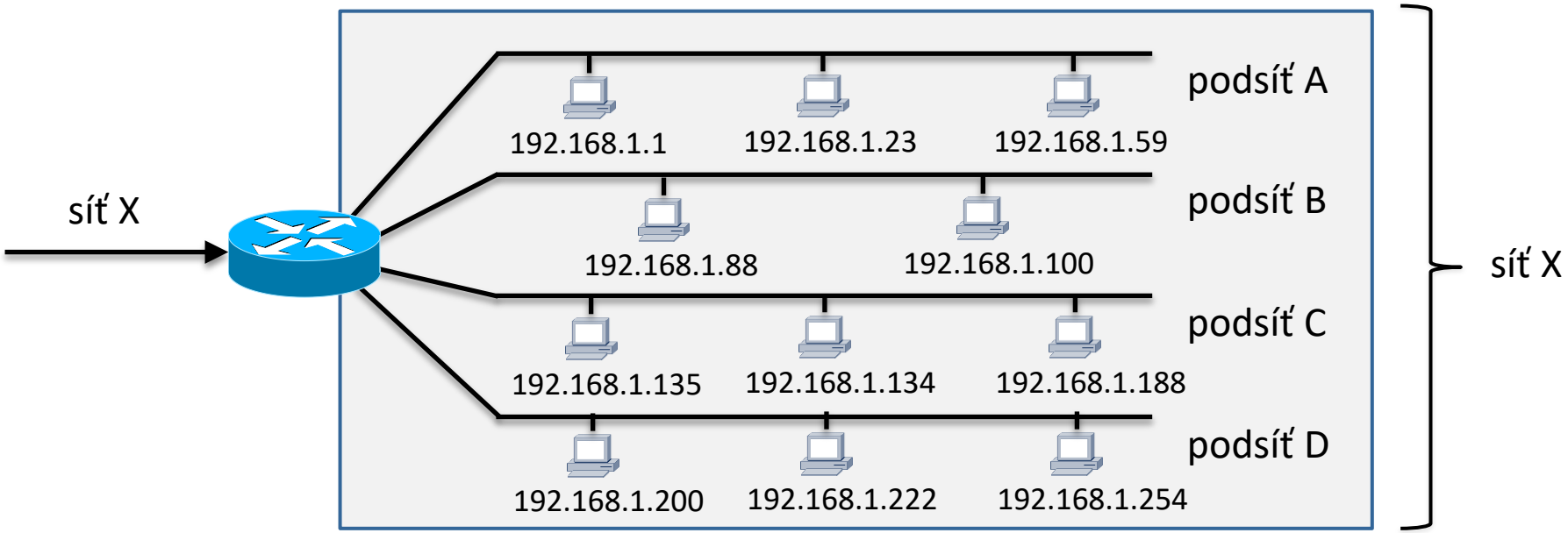
síť X: adresy 192.168.1.0 až 192.168.1.255

$255.255.255_D$ 00000000_B → síťová maska je: 255.255.255.0

192.168.1	00	000000-111111
	01	000000-111111
	10	000000-111111
	11	000000-111111

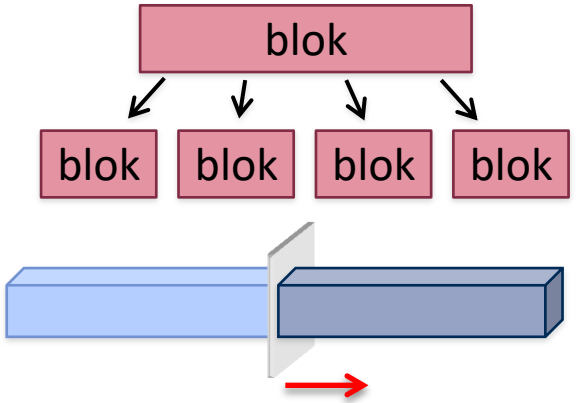
(pod)síť A: adresy 192.168.1.0 až 192.168.1.63
(pod)síť B: adresy 192.168.1.63 až 192.168.1.127
(pod)síť C: adresy 192.168.1.128 až 192.168.1.191
(pod)síť D: adresy 192.168.1.192 až 192.168.1.255

$255.255.255_D$ 11_B 000000_B → síťová maska je: 255.255.255.192 ($11\ldots1111000000_B$)

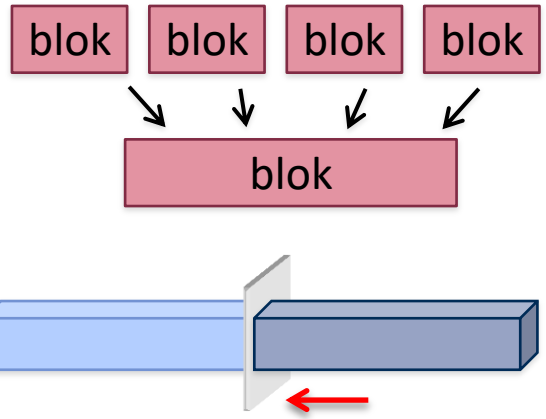


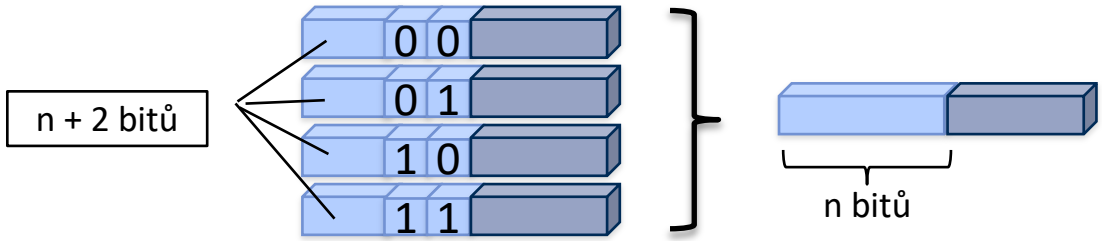
subnetting vs. supernetting

- subnetting je „rozdělování“ většího bloku na více menších bloků
  - posouvání předělu doprava, směrem k nižším bitům
  - lze využít jen „lokálně“
    - pouze v soustavách (pod)sítí, které se navenek chovají jako jedna (původní) síť
      - s původním blokem IP adres



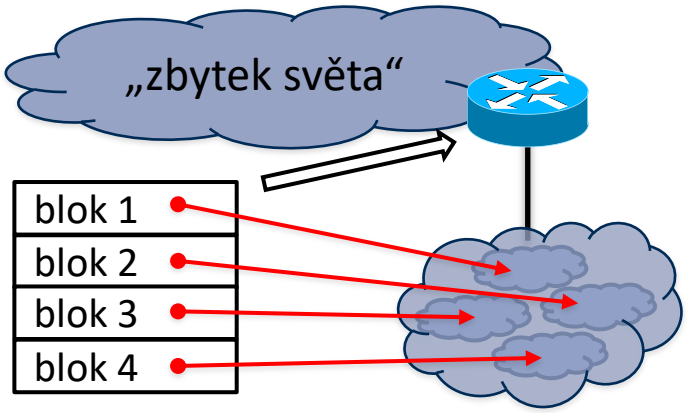
- opakem je supernetting (též: agregace)
  - aneb: *jak z několika menších bloků IP adres udělat jeden větší blok*
    - sloučit je (agregovat) do jednoho (většího) bloku
  - fakticky:
    - jde o posun předělu směrem doleva, k vyšším bitům
  - podmínka:
    - musí být vyčerpány všechny možné kombinace v těch bitech, přes které se předěl posouvá



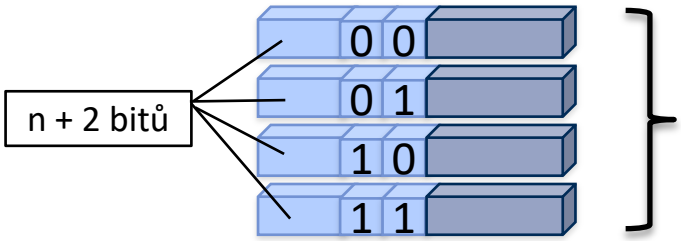


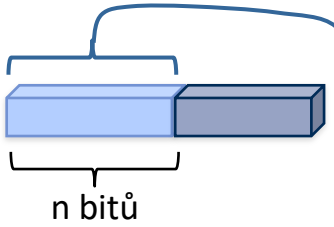
k čemu je supernetting?

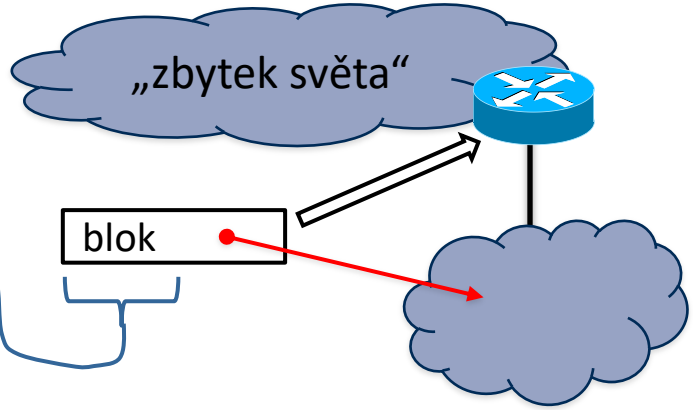
- supernetting je (obecná) technika, která se dá využít k různým účelům
- umožňuje mj. řešit problém nárůstu směrovacích tabulek
  - ten vzniká při přechodu na princip přidělování IPv4 adres „n* nejbližší nižší“
  - jedna síť dostala přiděleno několik samostatných bloků
    - např. 4 C-čkové IPv4 adresy
      - tj. 4 x 256 adres
  - ale: směrovače si (ve svých směrovacích tabulkách) musí pamatovat 4 položky
    - 4 různé „cesty“, vedoucí do stejné cílové sítě
      - nejen zbytečné, ale i drahé (zpomaluje to prohledávání směrovacích tabulek …)



  - supernetting umožňuje agregovat (sloučit) ony 4 samostatné položky do 1 položky

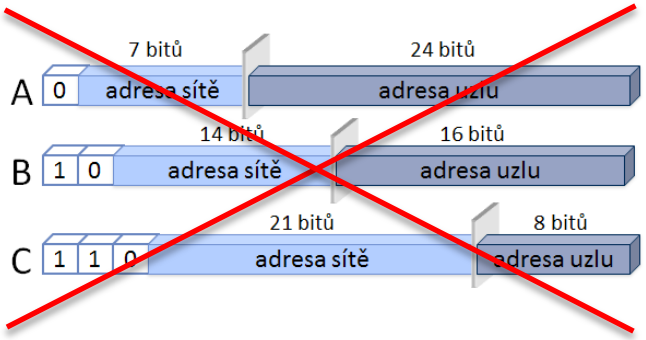


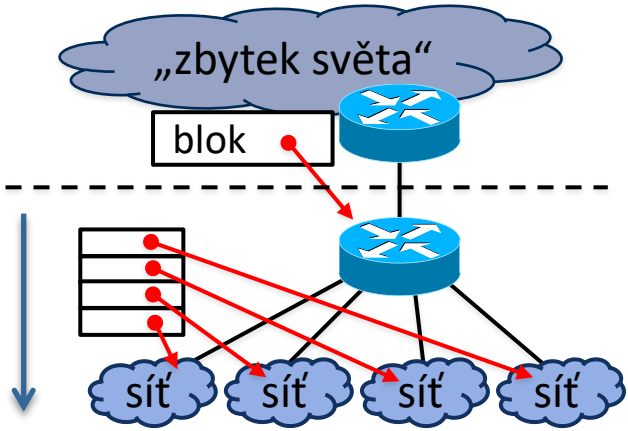




mechanismus CIDR

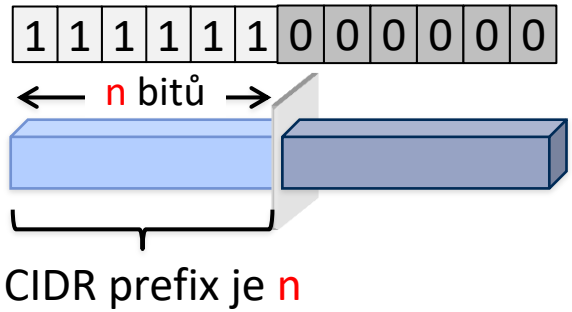
- CIDR: Classless Inter-Domain Routing
  - využívá technik supernettingu i subnettingu
  - (fakticky) ruší koncept tříd (A, B a C) u IPv4 adres
    - proto: Classless (doslova: *beztřídní* …)
  - místo toho:
    - umožňuje přidělovat libovolně velké bloky IPv4 adres: tzv. CIDR bloky
      - ve smyslu: s libovolným nastavením předělu mezi síťovou a relativní částí adresy
    - umožňuje dělit větší CIDR bloky na menší (subnetting)
      - využívá se při přidělování IPv4 adres, umožňuje vznik hierarchie přidělovatelů
        - „vyšší“ dostane větší CIDR blok, který rozdělí na menší CIDR bloky a sám je přidělí „nižším“ přidělovatelům (nebo přímo koncovým uživatelům)
    - umožňuje slučovat (agregovat) menší CIDR bloky do větších (supernetting)
      - využívá se při agregaci položek směrovacích tabulek
      - výrazně pomáhá s lokalizací směrovacích informací
        - umožňuje „ponechávat“ detailní směrovací informace jen tam, kde jsou skutečně zapotřebí, a nešířit je po celém Internetu





CIDR bloky a jejich prefixy

- mechanismus CIDR je „globální“
  - ve smyslu: na rozdíl od subnettingu není omezen jen na uzavřenou soustavu (pod)sítí s jedním vstupním bodem
    - ale je „vidět“ pro celý Internet
- musí ale řešit stejný problém jako subnetting:
  - určit, kde se nachází předěl mezi síťovou a relativní částí adresy
    - subnetting: řeší pomocí síťové masky (32 bitů, jako IP adresa)
    - CIDR: řeší pomocí tzv. (CIDR) **prefixu**
      - prefix je číslo, které udává počet bitů síťové části
- příklady:
  - CIDR blok s prefixem 8 odpovídá třídě A
    - jedné A-čkové síťové adrese
  - blok s prefixem 16 odpovídá třídě B
  - blok s prefixem 24 odpovídá třídě C
  - blok s prefixem 32 odpovídá 1 individuální IPv4 adrese
- zápis CIDR bloků: X.Y.X.W/prefix, například 192.168.0.0/16



čím menší je prefix,
tím větší je CIDR blok

počáteční adresa bloku

příklady CIDR bloků

192.168.1	00000000-11111111

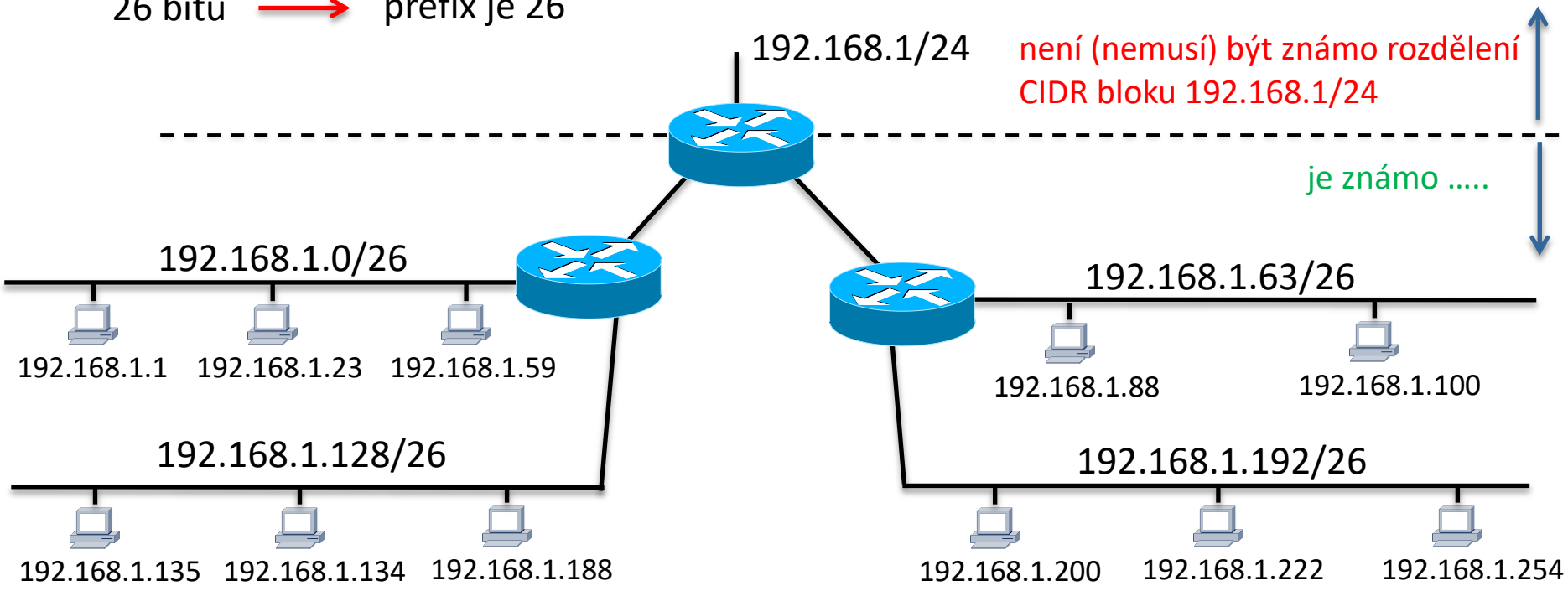
CIDR blok 192.168.1/24

24 bitů → prefix je 24

192.168.1	00	000000-111111
	01	000000-111111
	10	000000-111111
	11	000000-111111

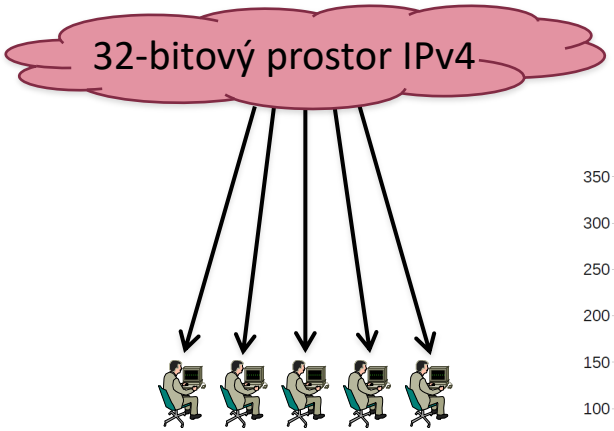
síť A: 192.168.1.0/26 (192.168.1.0 až 192.168.1.63)
síť B: 192.168.1.64/26 (192.168.1.64 až 192.168.1.127)
síť C: 192.168.1.128/26 (192.168.1.128 až 192.168.1.191)
síť D: 192.168.1.192/26 (192.168.1.192 až 192.168.1.255)

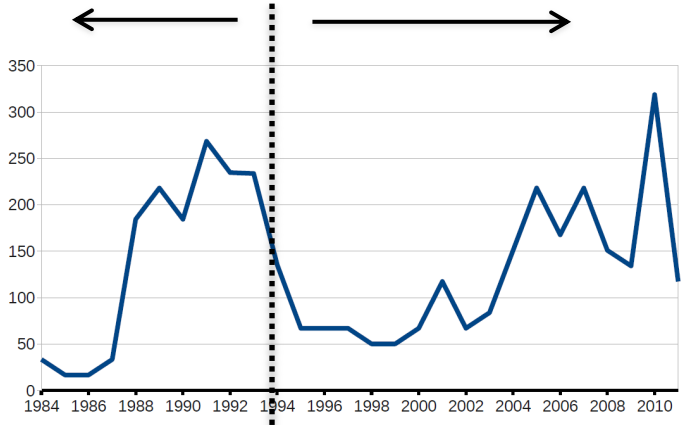
26 bitů → prefix je 26

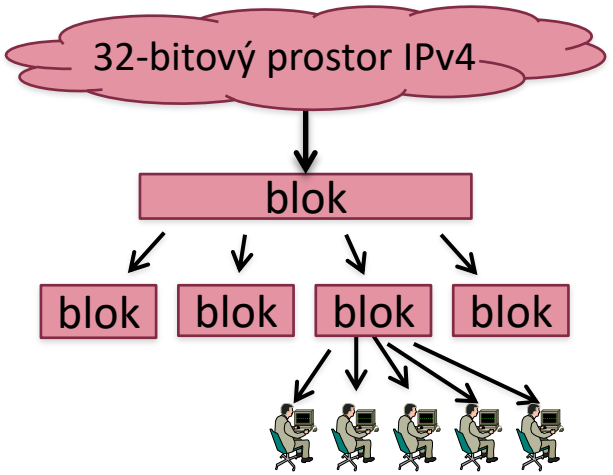


CIDR a přidělování IPv4 adres

- původně (před CIDR-em):
  - IPv4 adresy se přidělovaly „po třídách“
    - existovaly jen 3 velikosti přidělovaných bloků (síťové adresy třídy A, B nebo C)
      - dal se přidělit vždy jen určitý počet bloků jedné z těchto tří velikostí
      - na principu „nejbližší vyšší“ nebo „n_x nejbližší nižší“
    - bloky se daly dělit jen pomocí subnettingu
- později (CIDR, cca 1994)
  - IPv4 se přidělovaly po CIDR blocích
    - různě velkých, dle potřeby
  - možnost hierarchického přidělování
    - „vyšší přidělovatel“ dostal větší CIDR blok
      - ten rozdělil na menší CIDR bloky a ty dále přiděloval „nižším“ přidělovatelům
        - kteří postupovali obdobně

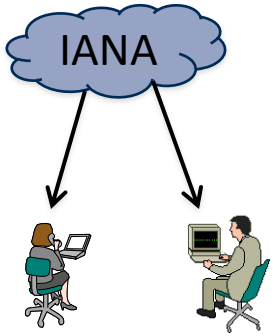


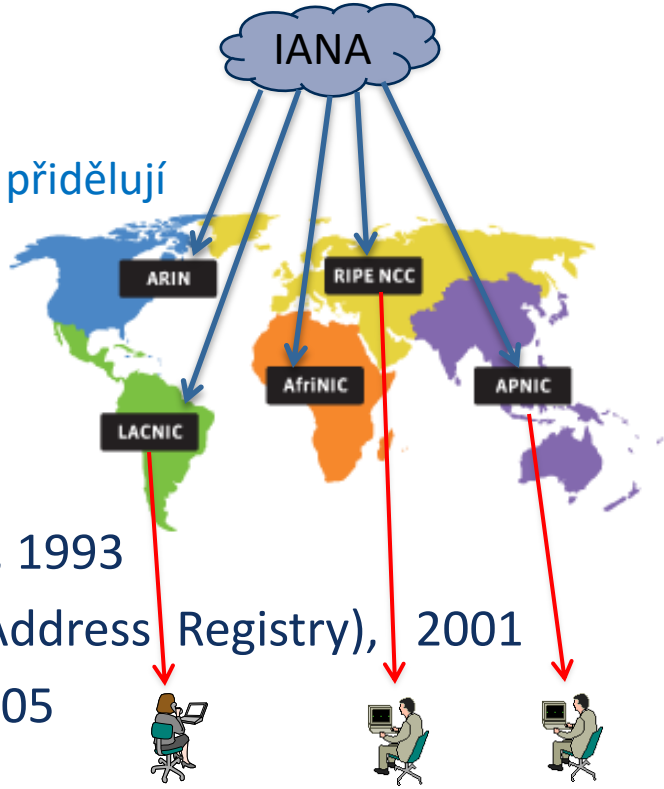




RIR, Regional Internet Registry

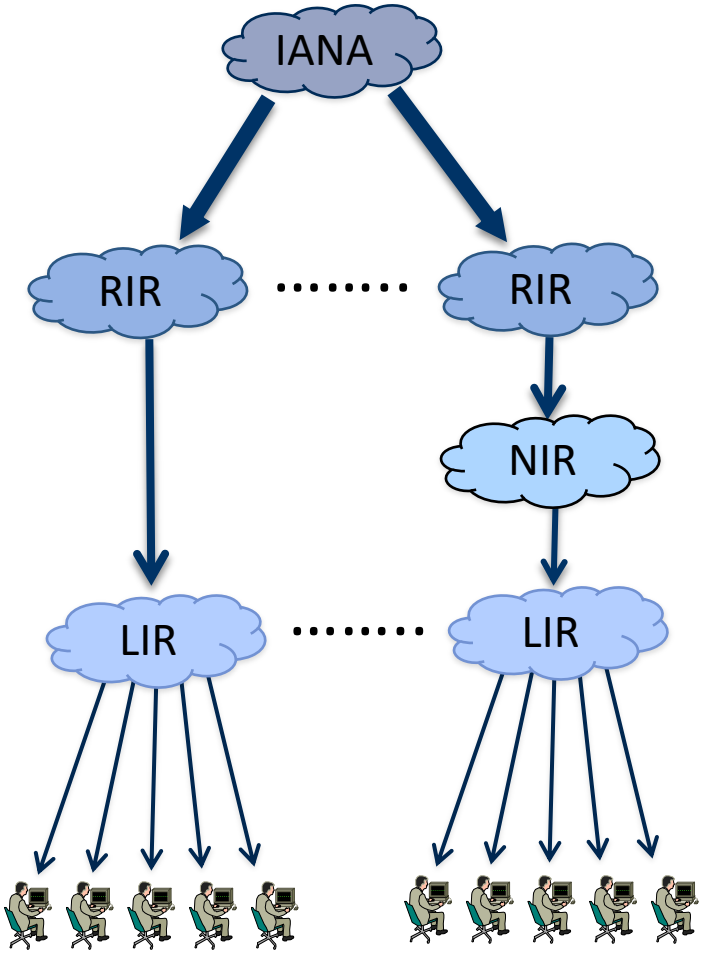
- správcem 32-bitového prostoru IPv4 adres byla organizace IANA
  - Internet Assigned Numbers Authority, www.iana.org
- původně:
  - celé bloky IPv4 adres přiděluje koncovým uživatelům přímo IANA
    - ale: agenda s tím spojená je stále rozsáhlejší a náročnější
- změna, cca 1991/92: — ještě přes nástupem CIDR
  - vznikají tzv. **RIR** (Regional Internet Registry)
    - velcí regionální „přidělovatelé“ IPv4 adres
      - dostávají od IANA větší počty bloků, které pak samy přidělují
  - dnes je po světě celkem 5 různých RIR
    - **ARIN** (American Registry for Internet Numbers)
      - založen 1997 ale fakticky fungovat již od roku 1991
    - **RIPE** (Reseaux IP Europeens), 1992
    - **APNIC** (Asia-Pacific Network Information Centre), 1993
    - **LACNIC** (Latin American and Caribbean Internet Address Registry), 2001
    - **AfriNIC** (African Network Information Centre), 2005





LIR, Local Internet Registry

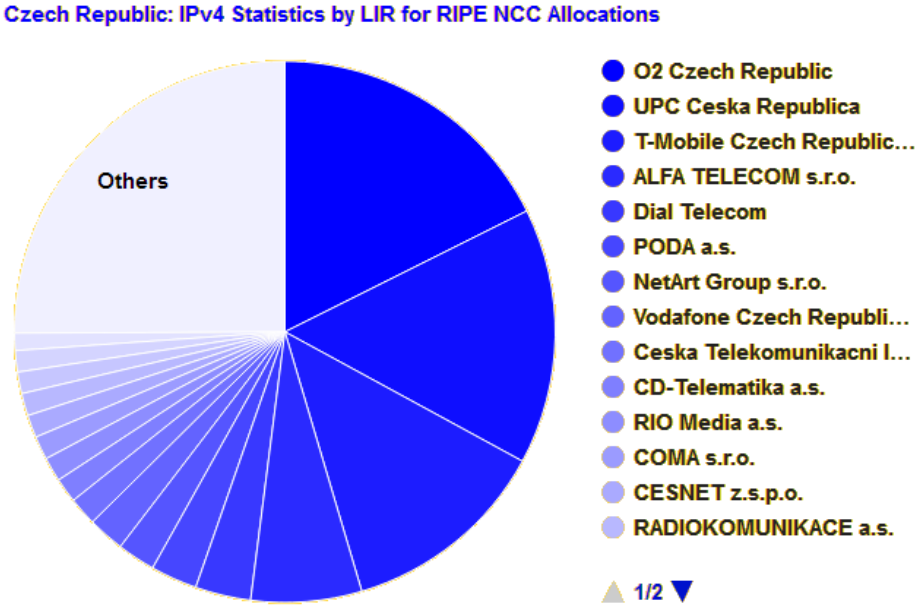
- hierarchie „přidělovatelů“ IP adres se dále rozrůstala
  - i v souvislosti s nástupem mechanismu CIDR a přidělování po CIDR blocích
    - RIR-y dostávají velké CIDR bloky od IANA
- další patra:
  - NIR: National Internet Registry
    - národní přidělovatelé (např. Čína, Japonsko, Jižní Korea, Taiwan, Vietnam, Mexiko, Brazílie, Chile):
      - od RIR dostávají větší CIDR bloky,
      - z nich přidělují menší CIDR bloky „nižším patrům“ (LIR)
  - LIR: Local Internet Registry
    - typicky: ISP (poskytovatelé přístupu)
      - dostávají větší CIDR bloky od RIR (NIR)
      - přidělují menší CIDR bloky koncovým zákazníkům
        - často jen jednotlivé IPv4 adresy



- všichni ISP v ČR jsou LIR-y, jsou členy RIPE (RIR-u), od něj dostávají CIDR bloky

počty IPv4 adres u ISP v ČR (6/2017)

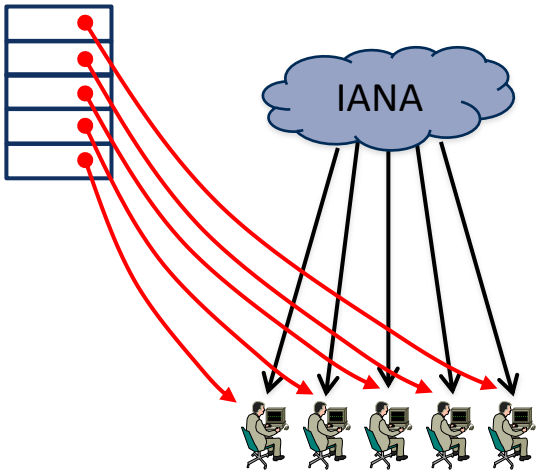
ISP	počet IPv4 adres	
O2 Czech Republic, a.s.	1401856	17.630 %
UPC Ceska Republica, s.r.o.	1216512	15.299 %
T-Mobile Czech Republic a.s.	992256	12.479 %
ALFA TELECOM s.r.o.	526336	6.619 %
Dial Telecom, a.s.	264192	3.322 %
PODA a.s.	221184	2.782 %
NetArt Group s.r.o.	181248	2.279 %
Vodafone Czech Republic a.s.	174080	2.189 %
Ceska Telekomunikacni Infrastruktura a.s	135168	1.700 %
CD-Telematika a.s.	117760	1.481 %
RIO Media a.s.	115712	1.455 %
COMA s.r.o.	111616	1.404 %
CESNET z.s.p.o.	107520	1.352 %
RADIOKOMUNIKACE a.s.	106496	1.339 %
Ipex Ltd.	102400	1.288 %
CoProSys a.s.	101376	1.275 %
Casablanca INT	81920	1.030 %
Internethome, s.r.o.	69632	0.876 %
Brno University of Technology	66560	0.837 %
SMART Comp. a.s.	59392	0.747 %
Fortech s.r.o.	58368	0.734 %
itself s.r.o.	54272	0.683 %
ha-vel internet s.r.o.	49152	0.618 %



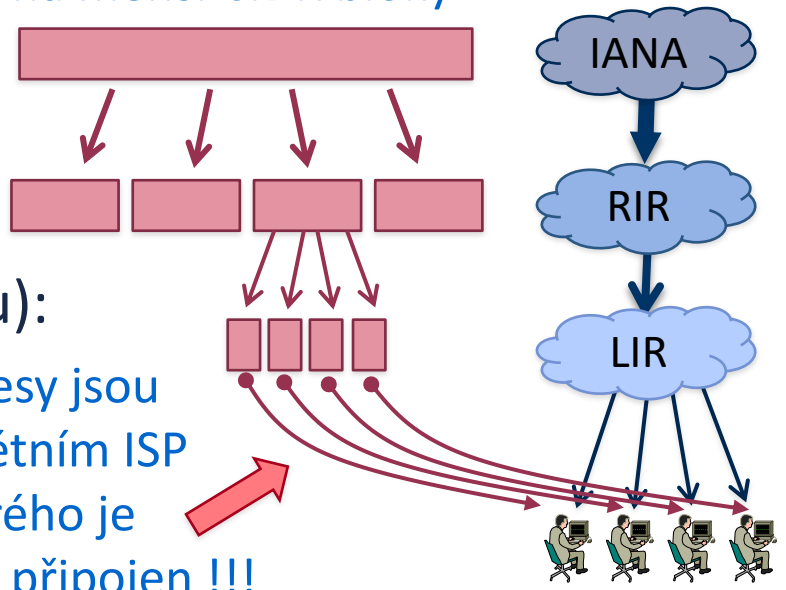
ISP	počet IPv4 adres	
Master Internet s.r.o.	41984	0.528 %
TS-Data s.r.o.	41984	0.528 %
ISP Alliance a.s.	34304	0.431 %
InterneXt 2000, s.r.o.	33792	0.425 %
SITKOM spol. s r.o.	33792	0.425 %
Nordic Telecom s.r.o	32768	0.412 %
SuperNetwork s.r.o.	29696	0.373 %
NWT a.s	27648	0.348 %
METRONET s.r.o.	23552	0.296 %
Emerald Real Group s.r.o.	21504	0.270 %
INTERNET CZ, a.s.	21504	0.270 %
.....		

závislost IP adres na ISP

- přidělování IPv4 adres pomocí mechanismu CIDR omezilo rozsah směrovacích informací
  - ale učinilo IP adresy závislé na způsobu připojení (ISP)
- původně:
  - všechny směrovače musely vědět, kde (ve kterém směru) leží každá jednotlivá síť třídy A, B či C
    - každý (páteřní) směrovač pro ni musel mít jednu položku ve své směrovací tabulce



- nově:
  - páteřní směrovače znají pouze cesty k největším CIDR blokům
  - „nižší“ směrovače již mají informace o rozdělení (některých) CIDR bloků na menší CIDR bloky



- důsledek (CIDR-u):
  - přidělené IP adresy jsou závislé na konkrétním ISP (LIR-u), přes kterého je koncový uživatel připojen !!!

(vy)čerpání IPv4 adres

- IANA přidělovala RIR-ům IPv4 adresy po CIDR blocích s prefixem 8
  - což by dříve odpovídalo 1 síťové adrese třídy A
    - tj. blok o velikosti $2^{32-8} = 2^{24} = 16777216$ individuálních IPv4 adres
- 1.2.2011
  - APNIC dostal 2 CIDR bloky /8 a organizaci IANA zbylo posledních 5 bloků /8
- 3.2.2011
  - IANA dala každému z 5 RIRů poslední blok /8

IPv4 adresy došly na úrovni IANA

- 15.4.2011
  - APNIC vyčerpal volné IPv4 adresy
    - zůstal jen poslední /8 blok, ze kterého může každý ISP dostat jen 1 blok /22

IPv4 adresy došly v regionu Asie/Pacifik

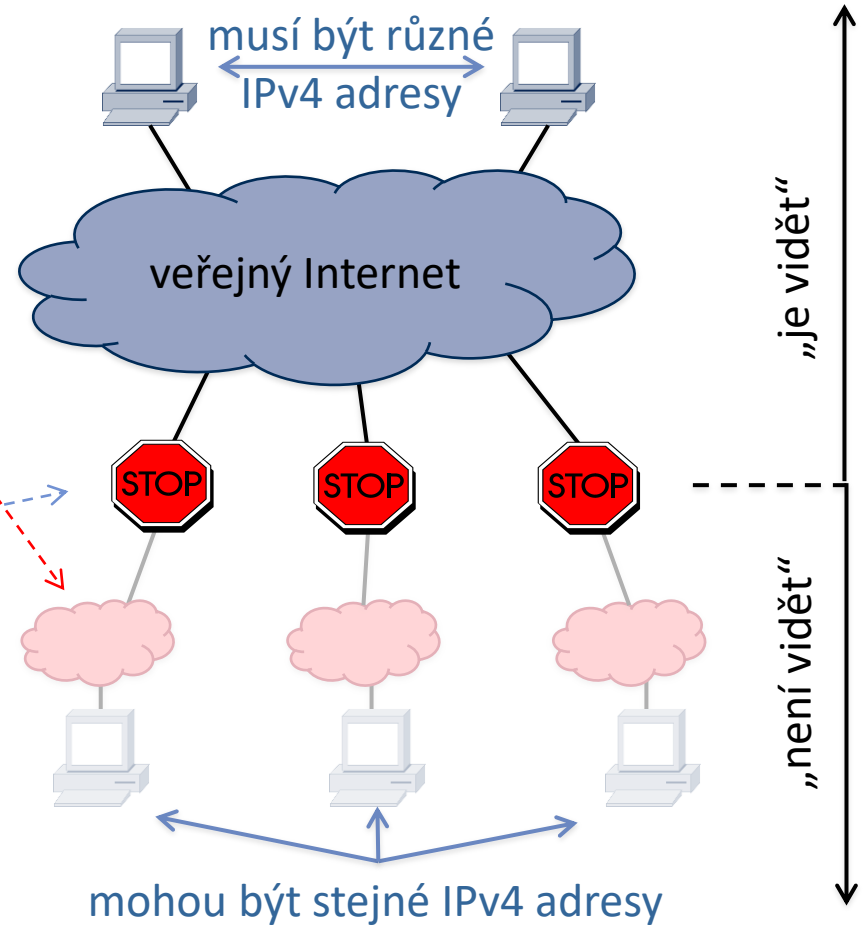
- 14.9.2012
  - RIPE vyčerpal volné IPv4 adresy
    - zůstal jen poslední (speciální) blok, ze kterého lze získat jen 1 blok /22
      - ale volné IPv4 jsou stále na úrovni LIRů (jednotlivých ISP)

IPv4 adresy došly v regionu Evropa/Rusko

privátní IP adresy

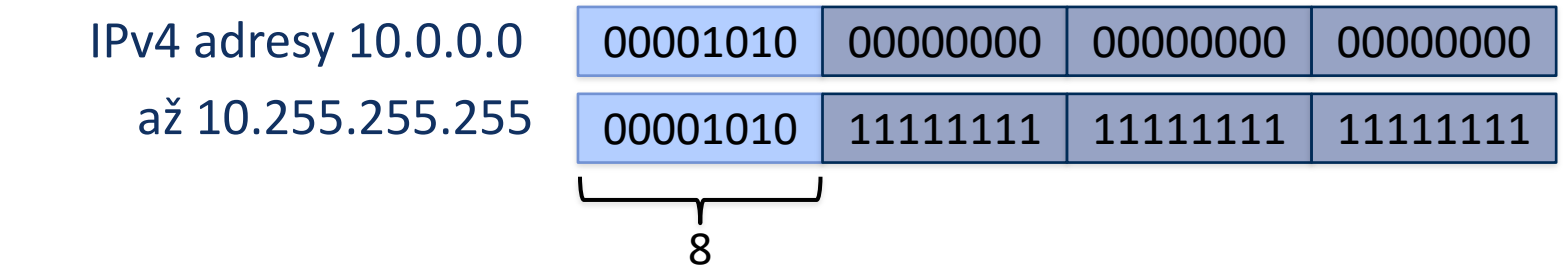
- jde o jedno z dočasných opatření na snížení úbytku IPv4 adres
- připomenutí:
  - ve veřejném Internetu musí mít každý uzel (rozhraní) unikátní IP adresu
    - jinak by nemohly korektně fungovat směrovací mechanismy
- výjimka:
  - stejné IP adresy lze využít opakovaně – ale jen v takových sítích, které „nejsou zvenku vidět“
    - v tzv. privátních sítích
      - proto: privátní IP adresy
  - podmínka:
    - privátní sítě musí být „schovány“ za něčím, co brání šíření informací o dostupnosti privátních adres
      - to může být:
        1. **překlad adres (NAT)**
        2. **firewall (proxy brána)**

velmi úspěšné

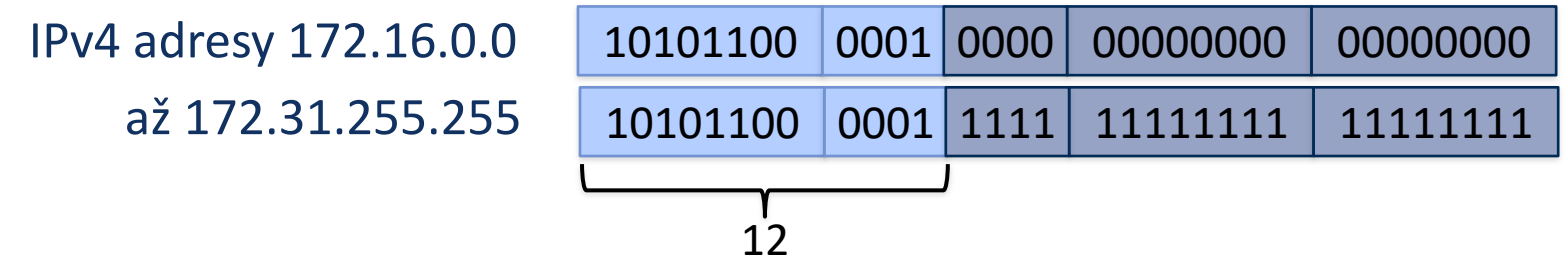


vyhrazené privátní IPv4 adresy

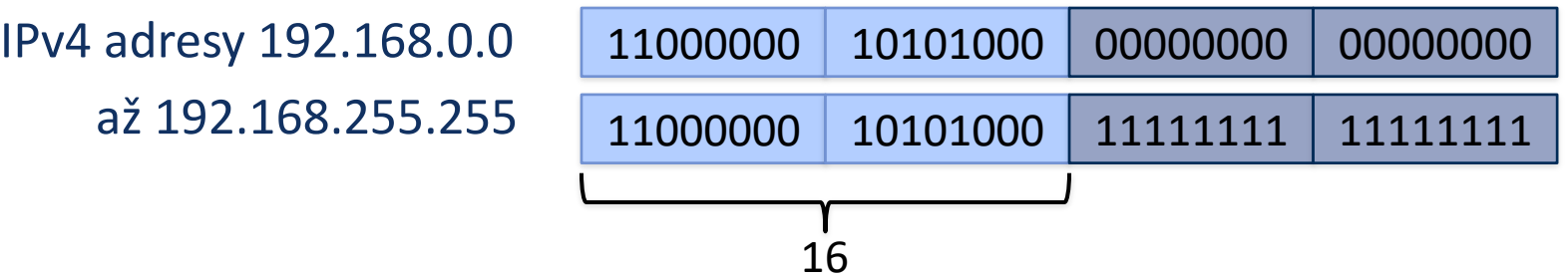
- v roli privátních IPv4 adres lze použít jakékoli IPv4 adresy
  - ale není to vhodné / správné !!!!
- existují IPv4 adresy, vyhrazené pro využití v roli privátních IPv4 adresy (dle RFC 1918)
  - 1 síťová adresa třídy A (10.x.x.x), resp. CIDR blok 10/8



  - 16 síťových adres třídy B (172.16.x.x až 172.31.x.x), resp. CIDR blok 172.16/12

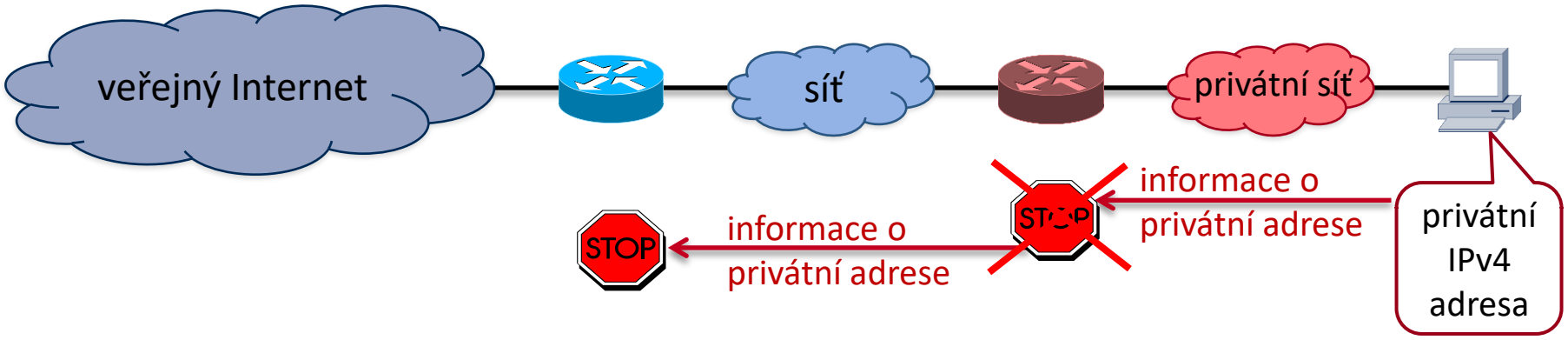


  - 256 síť. adres třídy C (192.168.0.x až 192.168.255.x), resp. CIDR blok 192.168/16



proč používat vyhrazené privátní adresy?

- souvisí to se „zneviditelněním“ privátní sítě a IP adres v těchto sítích
  - může být realizováno pomocí překladu adres (NAT) ve směrovačích
    - nebo pomocí proxy bran v rámci firewallů
  - princip fungování:
    - směrovač (proxy brána) nesmí propustit „ven“ informaci o privátní IP adrese
  - ale:
    - co když dojde k nějaké chybě a informace se (omylem) dostane ven?
  - řešení:
    - pokud byla použita vyhrazená privátní IPv4 adresa, pak nejbližší další směrovač či proxy brána chybu napraví (a informaci zastaví / nepustí dále)
      - pokud by byla použita jiná než vyhrazená IPv4 adres, další směrovač/brána nepozná, že by měl něco zastavit

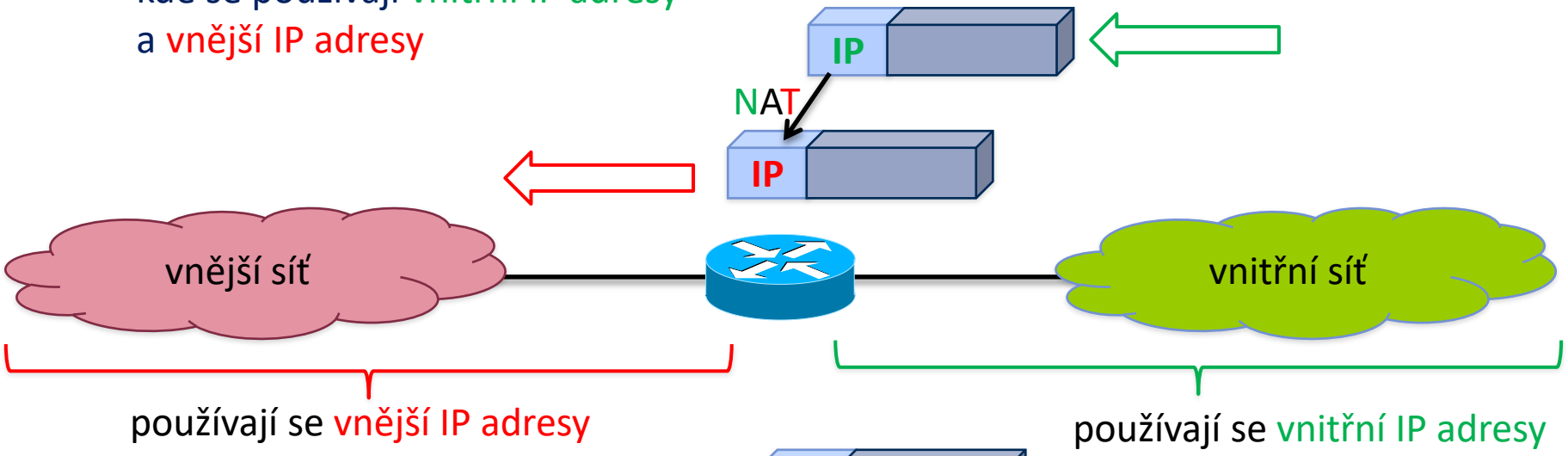


překlad adres (NAT)

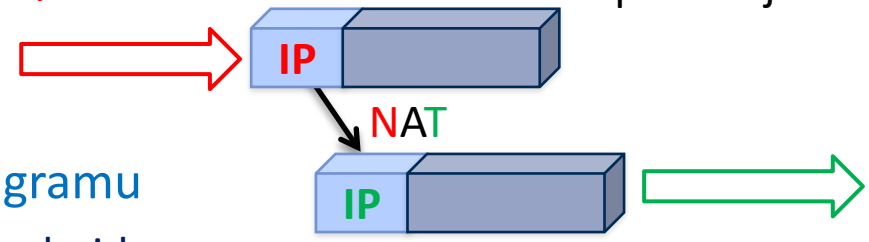
- NAT: Network Address Translation
  - jde o jeden z mechanismů, které umožňují používat privátní IP adresy
    - ale jde o obecnější řešení, které přináší i další výhody a možnosti využití:
      - „sdílení“ malého počtu veřejných IP adres větším počtem uzlů
      - větší volnost při přidělování IP adres ve vlastní síti
      - vyhnout se nutnosti přeadresování (změny IP adres) při změně ISP
      - snazší škálovatelnost vlastní sítě
      - …….
- vznikl kvůli potřebě šetřit IPv4 adresami
  - používá se hlavně v rámci IPv4 (NATv4)
  - lze jej implementovat i pro IPv6 (NATv6)
    - ale tam by neměl být/není tolik zapotřebí
- pomáhá i v oblasti bezpečnosti
  - představuje určitou formu firewallu
    - brání nevyžádanému přístupu k uzlům „za NAT-em“
- má ale i své problémy
  - s dostupností uzlů „za NAT-em“
  - zvyšuje složitost a spotřebu zdrojů
    - snižuje celkovou propustnost
    - často si existence NATu vynucuje dodatečná „nápravná“ řešení, která dostatečně nefungují
  - některé aplikace s NAT-em nefungují vůbec
    - takové, které pracují přímo s IP adresami
      - které NAT „nevidí“ a nepřekládá

princip fungování překladu adres

- NAT (překlad adres) funguje:
  - na síťové vrstvě
    - pracuje s IP datagramy
  - na rozhraní mezi vnitřní sítí a vnější sítí
    - kde se používají vnitřní IP adresy a vnější IP adresy
- princip fungování NATu
  - v hlavičkách datagramů přepisuje vnitřní IP adresy na vnější IP adresy
    - u odesilatele, resp. příjemce

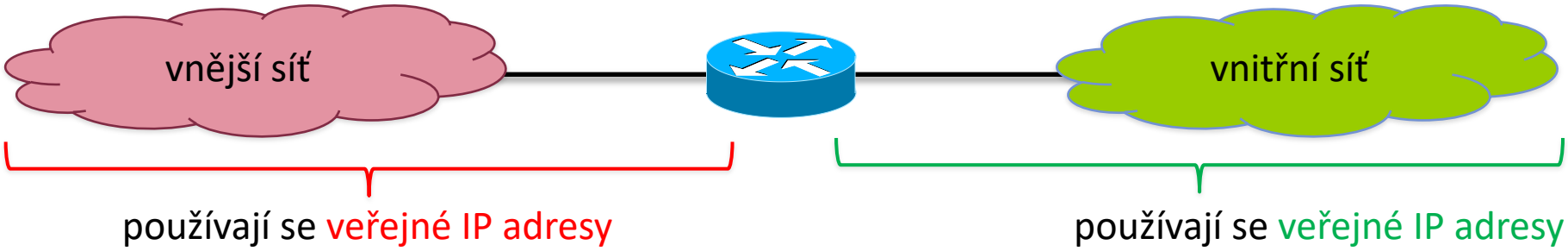


- NAT by (teoreticky):
  - neměl měnit tělo IP datagramu
    - ale v praxi se bez toho neobejde
      - musí přepočítávat i tzv. pseudohlavičky UDP datagramů i TCP segmentů !!!
        - protože ty jsou počítány z IP adres odesilatele a příjemce

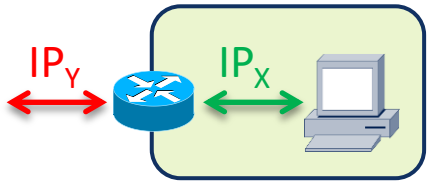


možné využití překladu adres

- řešení situace, kdy se chceme vyhnout přeadresování celé sítě
  - například: po změně ISP si chceme ponechat staré veřejné IP adresy od původního ISP, a neměnit je za nové veřejné IP adresy od nového ISP
    - staré veřejné IP adresy mají rozsah X (například: jeden konkrétní CIDR blok /24)
    - nové veřejné IP adresy mají rozsah Y (jde o jiný, ale stejně velký CIDR blok /24)



- vnější síť je veřejným Internetem
  - a používá veřejné IPv4 adresy
- vnitřní síť je privátní sítí
  - a používá veřejné IPv4 adresy z rozsahu X
- NAT překládá mezi:
  - veřejnými IP adresami z rozsahu Y a veřejnými IP adresami z rozsahu X
    - stylem 1:1
      - vazba mezi adresami z rozsahů X a Y může být statická
        - překlad proto může být „stejně dostupný“ v obou směrech



statický a dynamický NAT

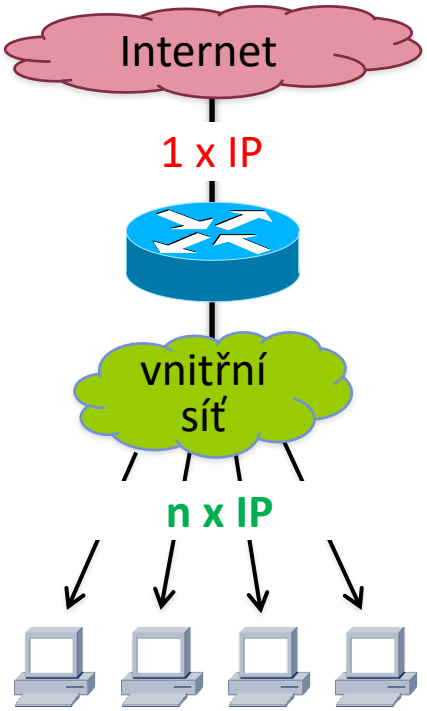
- vazba mezi vnitřními a vnějšími adresami při překladu může být:

- statická:
  - lze dopředu sestavit převodní tabulku
  - důsledky:
    - zařízení z vnitřní sítě (s vnitřní IP adresou) „vystupují navenek“ vždy pod stejnou vnější IP adresou
    - zařízení ve vnitřní síti může být kdykoli adresované (dostupné) z vnější sítě
      - pod svou vnější IP adresou
  - jinými slovy:
    - zařízení ve vnitřní síti jsou (bez problémů, vždy) dostupná přes „své“ (odpovídající) vnější IP adresy
  - nevýhoda:
    - rozsahy vnitřních a vnějších IP adres musí být stejné
      - nic se neušetří

- dynamická:
  - převodní tabulka se sestavuje až podle potřeby
    - ne všechna zařízení z vnitřní sítě musí mít „přidělenu“ vnější adresu
      - tj. rozsah Y může být menší než X
        - lze ušetřit vnější IP adresy
  - důsledky:
    - zařízení ve vnitřní síti nemusí být dostupná z vnější sítě
      - nejsou dostupná, pokud (ještě) nemají přidělenu žádnou vnější IP adresu
    - dostupná zařízení z vnitřní sítě „vystupují navenek“ pod různými vnějšími IP adresami
      - a dopředu není známo jakými

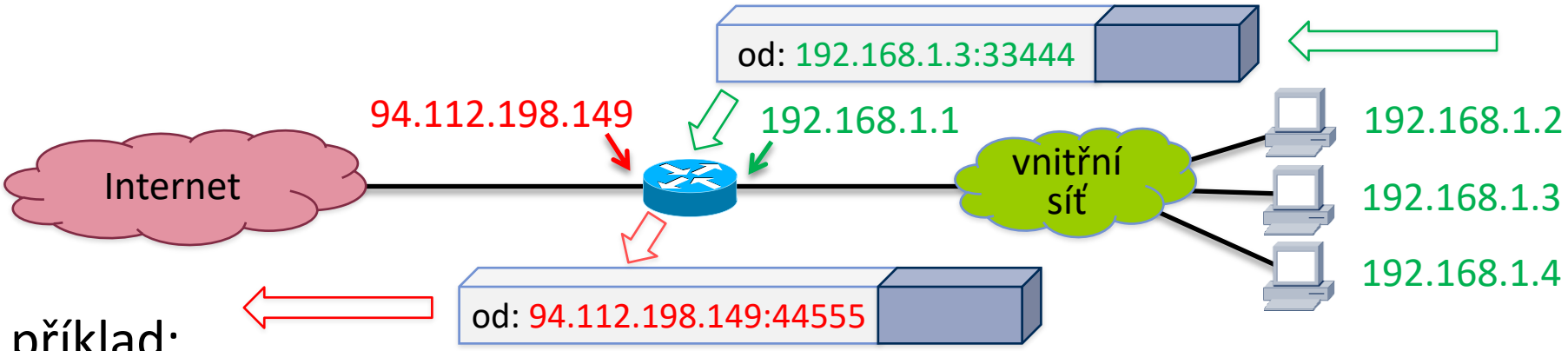
NAPT: Network Address and Port Translation

- je řešením pro situaci, kdy máme k dispozici méně vnějších IP adres než vnitřních IP adres
- typický případ: připojení k Internetu
  - máme (od ISP dostaneme přidělenu) jen 1 vnější IP adresu
  - ale ve vnitřní síti máme více koncových zařízení
    - kterým potřebujeme přidělit (vnitřní) IP adresu
      - správné je použít zde vyhrazené privátní IP adresy
        - například 192.168.xx.xx
  - vnitřní (privátní) síť oddělíme pomocí směrovače s NAT-em
    - který funguje v režimu „Address and Port Translation“



- princip fungování NAPT-u (s 1 vnější IP adresou):
  - všechny vnitřní IP adresy překládá do jedné (stejné) vnější IP adresy
  - rozlišení toho, o kterou vnitřní IP adresu jde, si uchovává v čísle portu
    - proto: Network Address and Port Translation, zkratkou NAPT
      - častěji se ale používá označení: Port Address Translation, zkratkou PAT
    - vazba mezi vnějšími a vnitřními IP adresami a porty má dynamický charakter !!

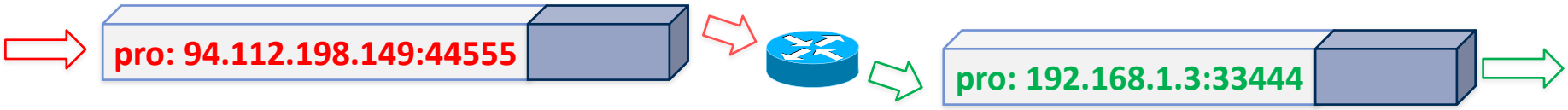
způsob fungování PATu: příklad



- příklad:
  - uzel ve vnitřní síti s adresou 192.168.1.3 odesílá IPv4 datagram do Internetu
    - ze svého portu č. 33444
  - směrovač s NATem provede překlad v IPv4 datagramu:
    - 192.168.1.3:33444 přepíše na 94.112.198.149:44555
      - tuto „vazbu“ si zanese do tabulky (tzv. mapping, binding)

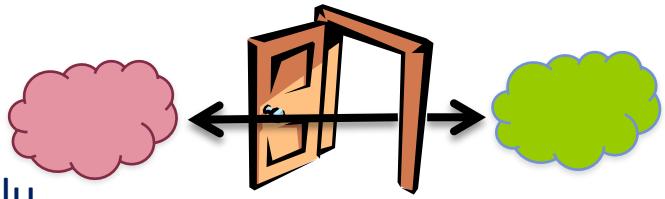
vnější	vnitřní
……	……
94.112.198.149: 44555	192.168.1.3: 33444
……	……

  - odpověď přijde na adresu 94.112.198.149 pro port 44555
    - směrovač s NATem podle své tabulky pozná, že odpověď „na port“ 44555 patří uzlu s vnitřní adresou 192.168.1.3, na jeho port 33444
      - v IP datagramu: 94.112.198.149:44555 přepíše na 192.168.1.3:33444



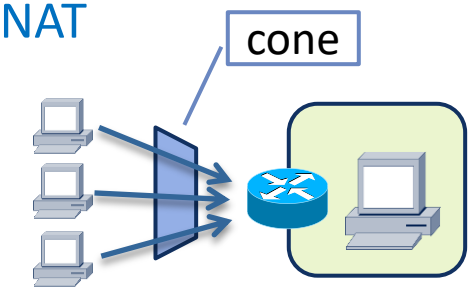
dynamický charakter PAT-u

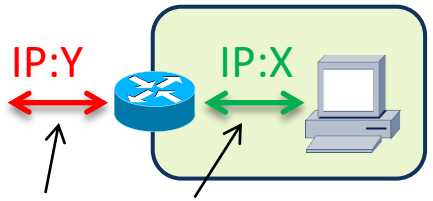
- vazba (mapping, binding) mezi vnitřní IP a port-em a vnější IP a port-em vzniká dynamicky a je pouze dočasná
  - její trvání je závislé na implementaci
    - v praxi se liší podle použitého transportního protokolu
      - UDP: 30 až 300 sekund, TCP: 30 až 60 minut !!!
  - poté vazba zaniká (expiruje)



vnější	vnitřní
.....	.....
94.112.198.149: 44555	192.168.1.3: 33444
.....	.....

- představa:
  - vytvořením vazby se jakoby dočasně otevírá brána skrze NAT
    - kterou lze procházet jak směrem „ven“, tak i směrem „dovnitř“
- podle toho, jak tato brána funguje, se rozlišuje:
  - Full Cone NAT
  - (IP) Restricted Cone NAT
  - Port Restricted NAT
  - Symmetric NAT

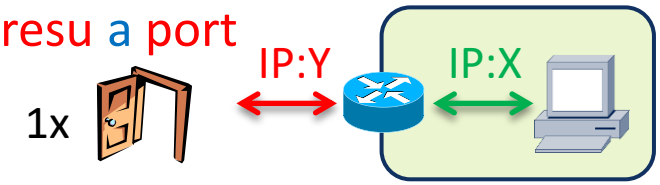




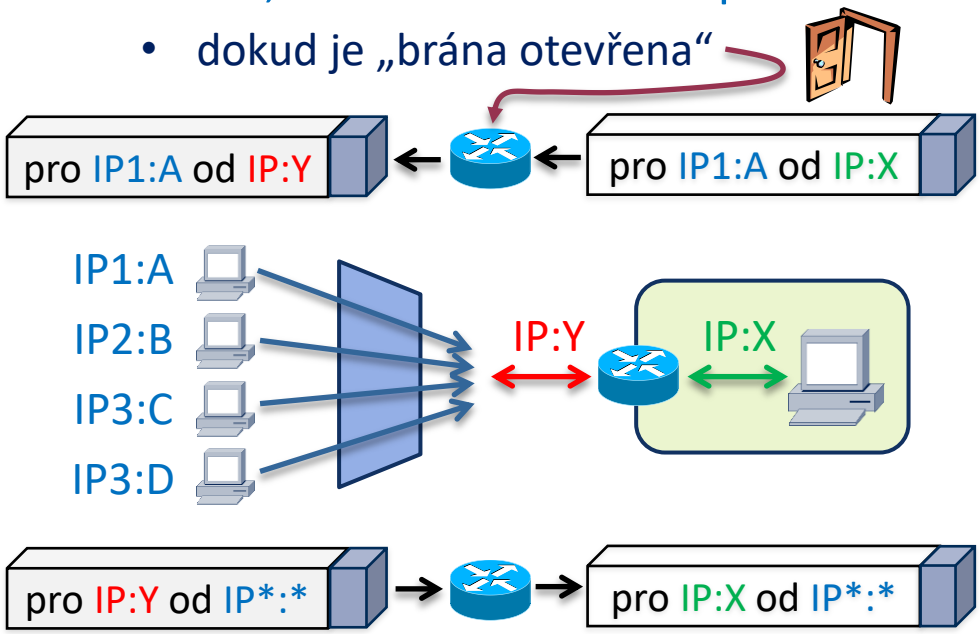
- liší se v tom:
  - jak se mapuje další komunikace ze stejné vnitřní IP adresy a portu
  - které vnější uzly mohou využít „brány“ pro komunikaci směrem do vnitřní sítě

varianty NAT-u (PAT-u)

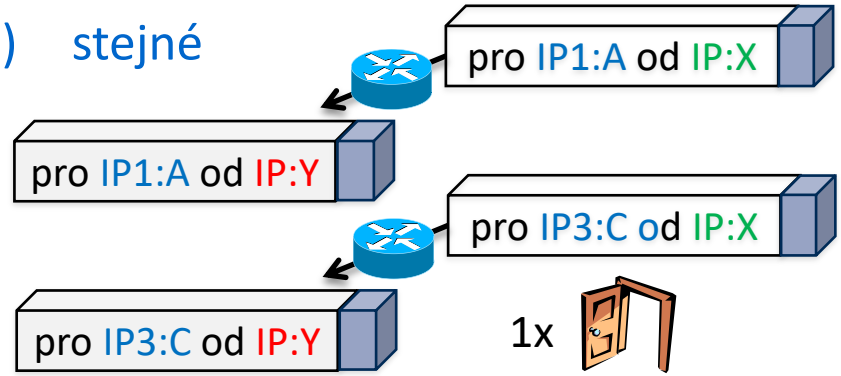
- varianta Full Cone NAT
  - a) stejná vnitřní IP adresa a port se překládají na stejnou vnější IP adresu a port



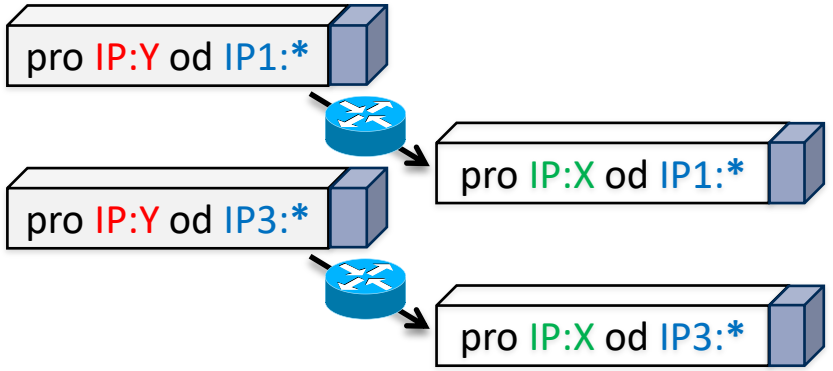
  - b) „odpovídat“ (přenášet data směrem „dovnitř“, na příslušnou vnější IP adresu a port) může kterýkoli vnější uzel, z kteréhokoli svého portu
    - dokud je „brána otevřena“



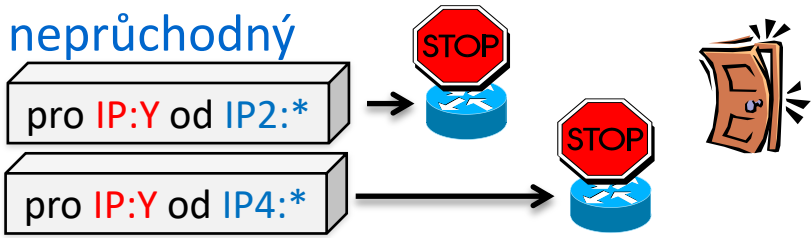
- varianta (IP) Restricted Cone NAT
  - a) stejné



  - b) „odpovídat“ mohou jen „oslovené“ vnější uzly, ale z libovolného portu

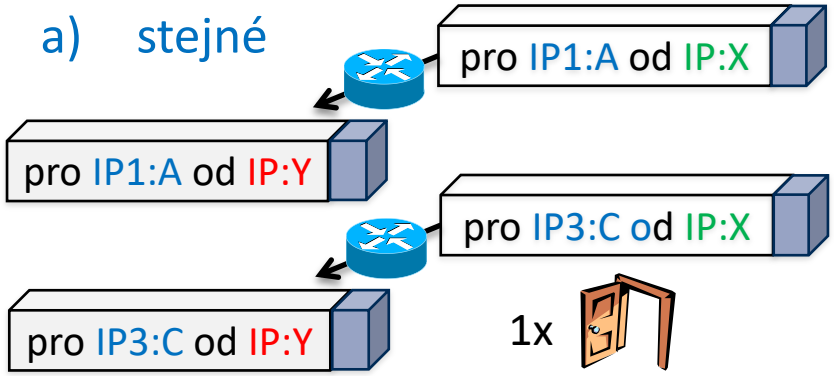


  - c) pro ostatní vnější uzly je NAT na IP:Y neprůchodný

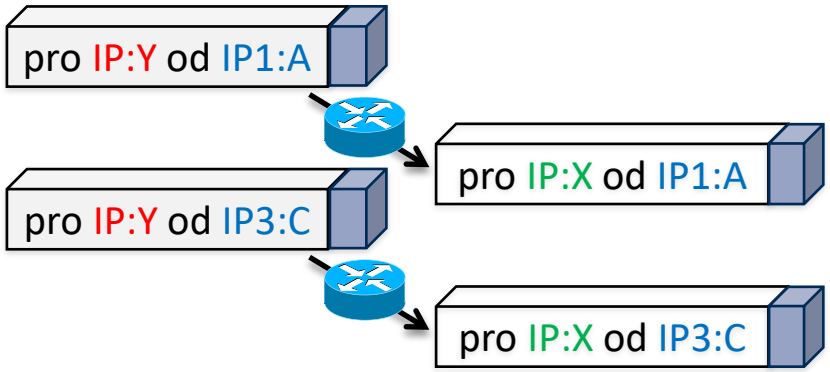


varianty NAT-u (PAT-u)

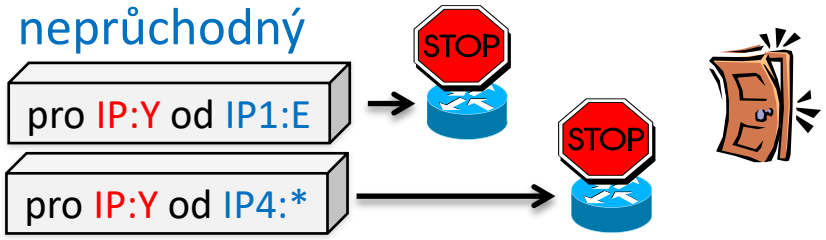
- varianta Port Restricted Cone
  - a) stejné



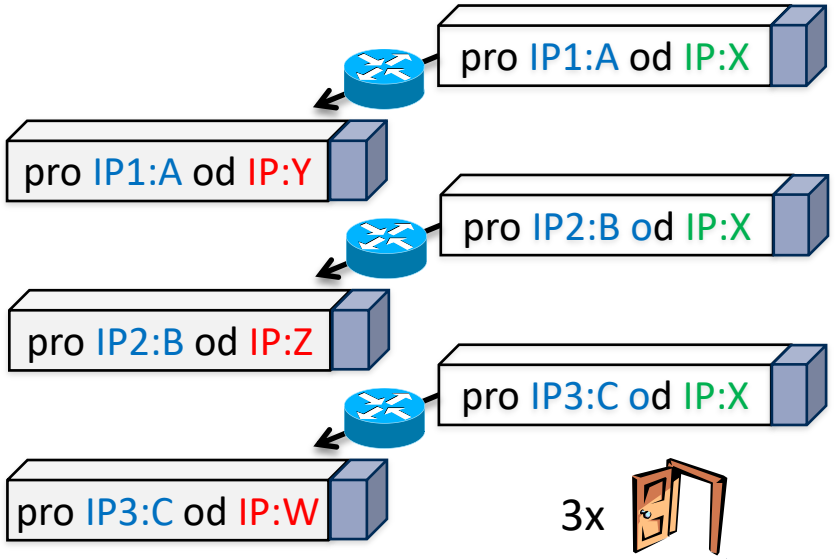
  - b) „odpovídat“ mohou jen „oslovené“ vnější uzly, a jen z „oslovených“ portů



  - c) pro ostatní uzly a porty je NAT na IP:Y neprůchodný



- varianta Symmetric NAT
  - a) stejná vnitřní IP adresa a port se překládají na stejnou vnější IP adresu ale (pokaždé) jiný port



  - b) „odpovídat“ mohou jen „oslovené“ vnější uzly, a jen z „oslovených“ portů
    - a jen na tu kombinaci IP:port, která byla pro ně „otevřena“

problémy NAT-u (PAT-u)

- nemůže fungovat v případě, kdy:
  - protokol vyšší vrstvy si „schovává“ IP adresy do těla IPv4 datagramu
    - protože pak NAT neví, že by je měl přeložit
      - jde například o protokol IPSEC
- řešení: tzv. inteligentní NAT/PAT
  - zná ty protokoly, které si ukládají IP adresy do těla IPv4 datagramu
    - a překládá i tyto adresy
      - protože ví, kde jsou a jak je má přeložit
- ALG: aplikační brány
  - výrobci se často snaží řešit sami, pomocí tzv. Application Level Gateway
    - orientovaných na konkrétní aplikace
    - obvykle ale nefungují, problém jen zhoršují (nejlépe je vypnout)
- nelze iniciovat komunikaci „zvenčí“, do vnitřní sítě
  - kromě statického NAT-u
  - „zvenčí“ nelze navázat spojení
    - ani začít komunikovat nespojovaně (datagramy)
  - již kvůli tomu, že není známa (dynamická) vazba vnější adresy na vnitřní IP adresu
- řešení: tzv. NAT Traversal
  - řada různých technik
    - např. STUN, TURN, …..
- jde současně o určitý prvek zabezpečení
  - brání to „průchodu“ nežádoucího provozu směrem do vnitřní sítě
    - ale současně brání i žádoucímu provozu