



Katedra softwarového inženýrství,
Matematicko-fyzikální fakulta,
Univerzita Karlova, Praha



Rodina protokolů TCP/IP, verze 2.4

Část 6: IP směrování

Jiří Peterka, 2007

co je směrování (routing)?

- striktně vzato:
 - **volba směru pro další předání paketu/datagramu**
- ve skutečnosti zahrnuje:
 - **výpočet optimální cesty**
 - je to kombinatorický problém hledání nejkratší cesty v grafu
 - výsledkem jsou "podklady pro volbu směru"
 - **uchovávání směrovacích informací ("podkladů")**
 - vedení směrovacích tabulek
 - **předávání paketů (forwarding)**
 - používání výsledků výpočtů ("podkladů")
 - **udržování směrovacích informací**
 - aktualizace údajů pro výpočty cest, reakce na změny
- co všechno s tím dále souvisí?
 - celková koncepce směrování
 - celková koncepce internetu
 - katenetový model
 - které uzly se účastní
 - historický vývoj
 - **přímé a nepřímé doručování**
 - **metody optimalizace směrovacích tabulek**
 - **řešení směrování v opravdu velkých systémech**
 - autonomní systémy
 - **směrovací politiky**
 -

celková koncepce směrování

- **statické směrování**

- obsah směrovacích tabulek má statický charakter a nemění se
 - vyžaduje to ruční konfiguraci směrovačů (jejich směrovacích tabulek)
 - což je pracné a náchylné k chybám
 - nereaguje to na změny v síti
 - dostupnost nějaké sítě není závislá na stavu spojení
- používá se jen výjimečně:
 - pro definování tzv. implicitních cest
 - default route
 - pro zavedení směrů které nejsou inzerovány
 - například v rámci firewallů
 - pro implementaci speciálních směrovacích politik
 - kdy je záměrem reagovat na směrovací informace jinak než obvykle
 - jako obrana proti nekorektním směrovacím informacím
 -

- **dynamické směrování**

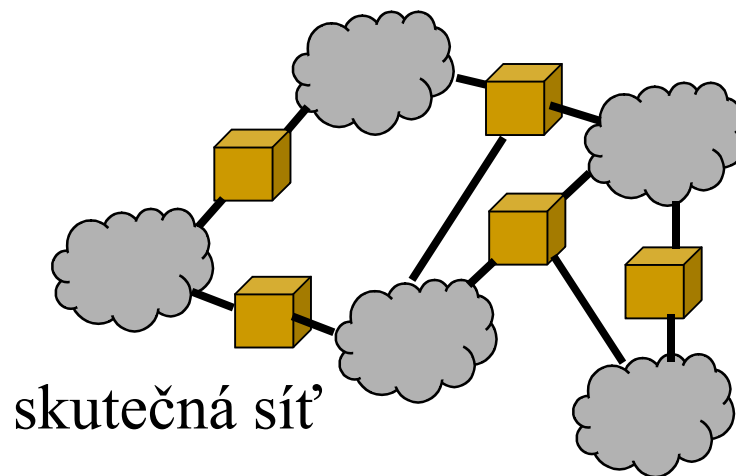
- obsah směrovacích tabulek má dynamický charakter a mění se
 - často je základ konfigurace vytvářen staticky, ruční konfigurací směrovačů
 - např. implicitní cesty
 - ostatní údaje se průběžně aktualizují
- existují dvě základní varianty dynamického směrování
 - **vector-distance routing**
 - sousední směrovače si předávají celé své směrovací tabulky (obsahující "vzdálenostní vektory")
 - je hůře škálovatelné a méně stabilní, přestává se používat
 - **link-state routing**
 - směrovače si předávají jen údaje o průchodnosti cest k sousedům
 - lépe škálovatelné, používá se

připomenutí: koncepce internetu

- Internet je budován na principu katenetu
 - je soustavou vzájemně propojených sítí
 - jednotlivé sítě jsou odděleny směrovači
- "předávání" paketů (forwarding):
 - má na starosti protokol IP
- aktualizaci směrovacích informací, výpočty cest:
 - zajišťují specializované protokoly jako RIP, OSPF, ..



představa katenetu



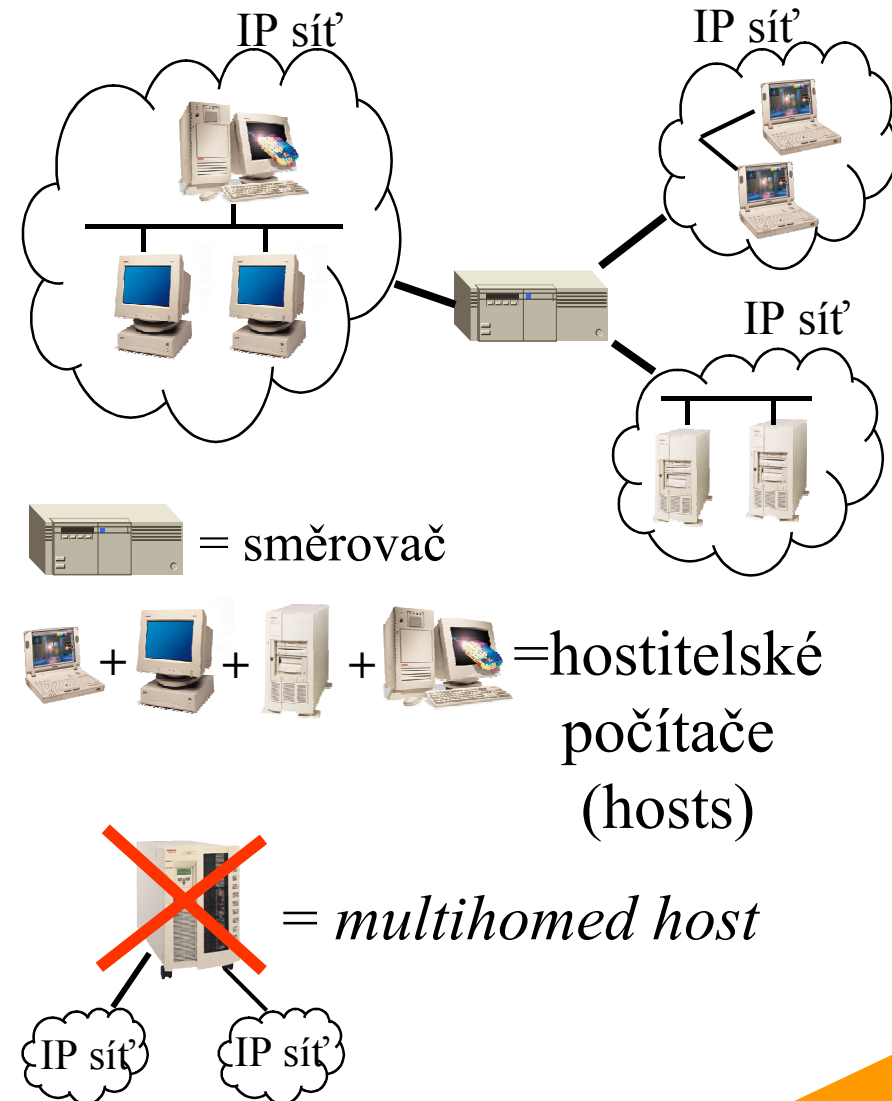
skutečná síť



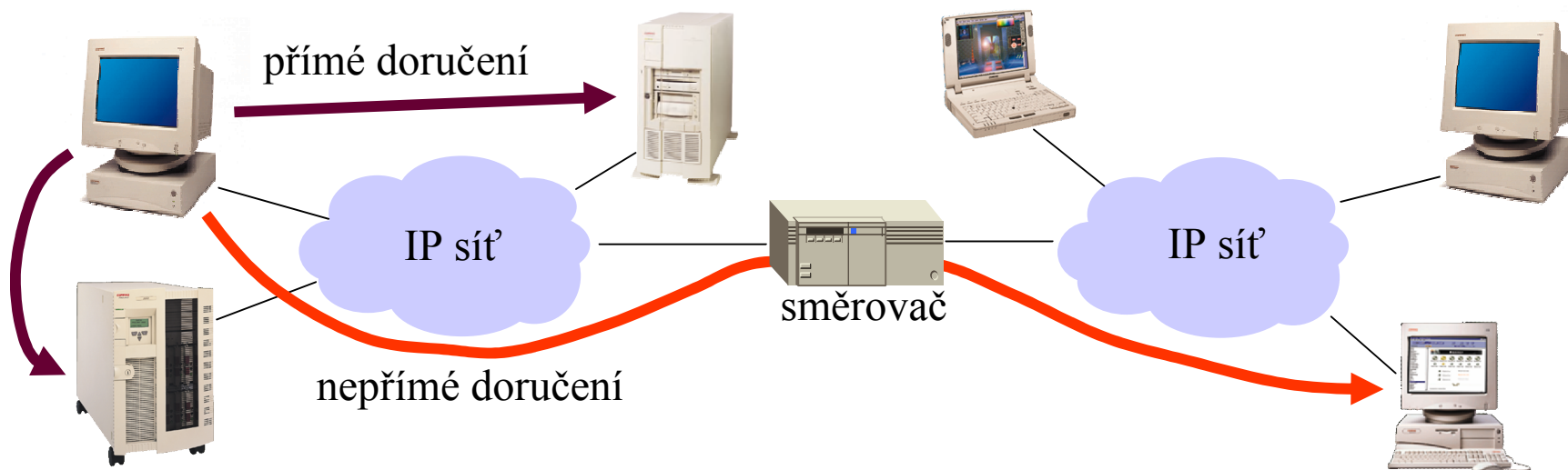
připomenutí:

hostitelské počítače vs. směrovače

- TCP/IP předpokládá, dva typy uzlů v síti:
 - **hostitelské počítače (host computers)**
 - tj. koncové uzly, např. servery, pracovní stanice, PC, různá zařízení (tiskárny, ...)
 - jsou připojeny jen do jedné IP sítě (mají jen jednu síťovou adresu)
 - **směrovače (IP Routers)**
 - jsou připojeny nejméně do dvou IP sítí
 - zajišťují "přestup" (směrování)
- teze:
 - **oba typy uzlů by se neměly prolínat**
 - směrovače by neměly plnit další funkce
 - hostitelské počítače by neměly fungovat jako směrovače
 - v podobě tzv. multihomed-hosts, kdy jsou připojeny do více sítí současně



přímé a nepřímé doručování



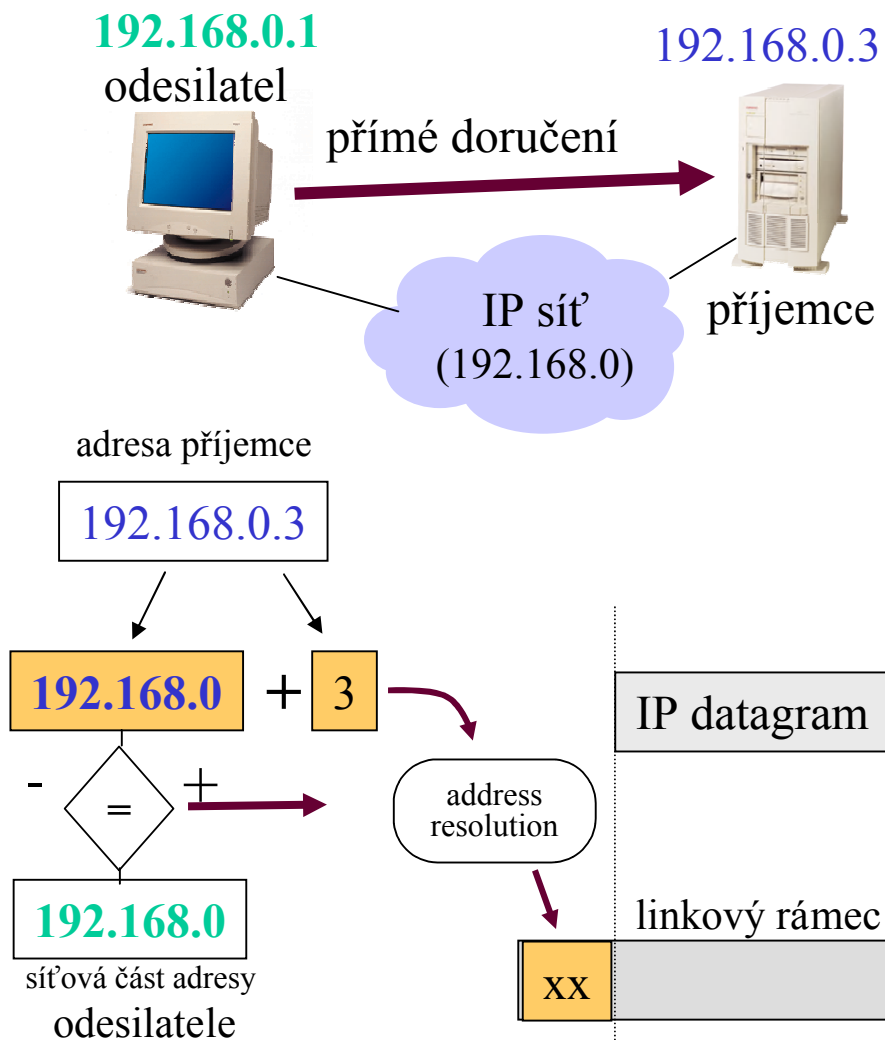
- **přímé doručování:**

- odesílatel a příjemce se nachází ve stejné IP síti
 - pozná se podle toho, že mají stejnou síťovou část své IP adresy
- odpadá rozhodování o volbě směru, o doručení se dokáže postarat linková vrstva (vrstva síťového rozhraní)
 - odesílatel pošle datagram "přímo" koncovému příjemci

- **nepřímé doručování**

- odesílatel a příjemce se nachází v různých IP sítích
- odesílatel musí určit nejvhodnější odchozí směr (resp. směrovač ležící v tomto směru)
 - odesílatel pošle datagramu směrovači ve zvoleném odchozím směru

představa přímého doručování



- odesílatel rozdělí cílovou adresu na její síťovou část a relativní část
 - použije dělení, platné pro jeho vlastní síť
 - masku sítě, CIDR prefix, event. vyjde z třídy adresy
 - získá síťovou část adresy příjemce
- pokud se síťová část adresy příjemce shoduje se síťovou částí vlastní adresy, jde o přímé doručování
- odesílatel převede IP adresu příjemce na jeho linkovou adresu
 - provede "Address Resolution", např. pomocí protokolu ARP
 - získá linkovou adresu XX
- odesílatel (jeho síťová vrstva) předá datagram vrstvě síťového rozhraní s požadavkem na doručení na adresu XX

představa nepřímého doručování

192.168.0.1

192.168.1.3

nepřímé doručení

IP síť
(192.168.0)

IP síť
(192.168.1)

192.168.0.2

adresa příjemce

192.168.1.3

192.168.1

+

3

- = +

192.168.0

síťová část adresy
odesílatele

volba odchozího směru

přes směrovač 192.168.0.2

přímé doručování

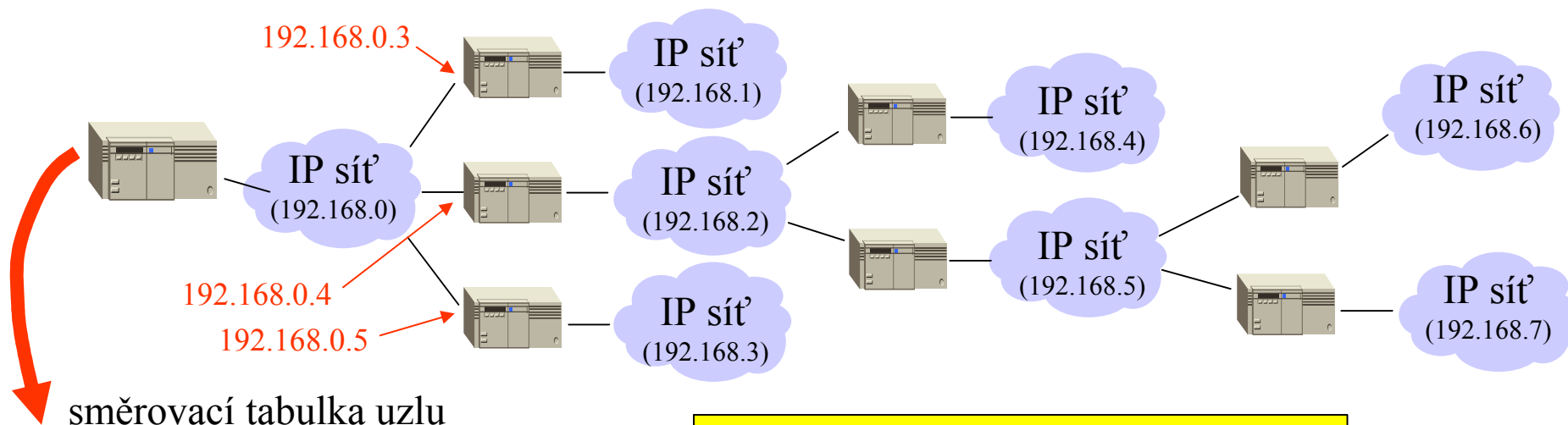
předchozí případ

směrovací tabulka

pro síť	posílej přes
192.168.1	192.168.0.2
.....	

- porovnáním síťových částí adres odesílatele zjistí, že se příjemce nachází v jiné síti
- odesílatel se "podívá" do své směrovací tabulky a podle ní zvolí odchozí směr
 - směrovač v odchozím směru
- odešle datagram zvolenému směrovači
 - již se jedná o přímé doručení

představa směrovacích tabulek

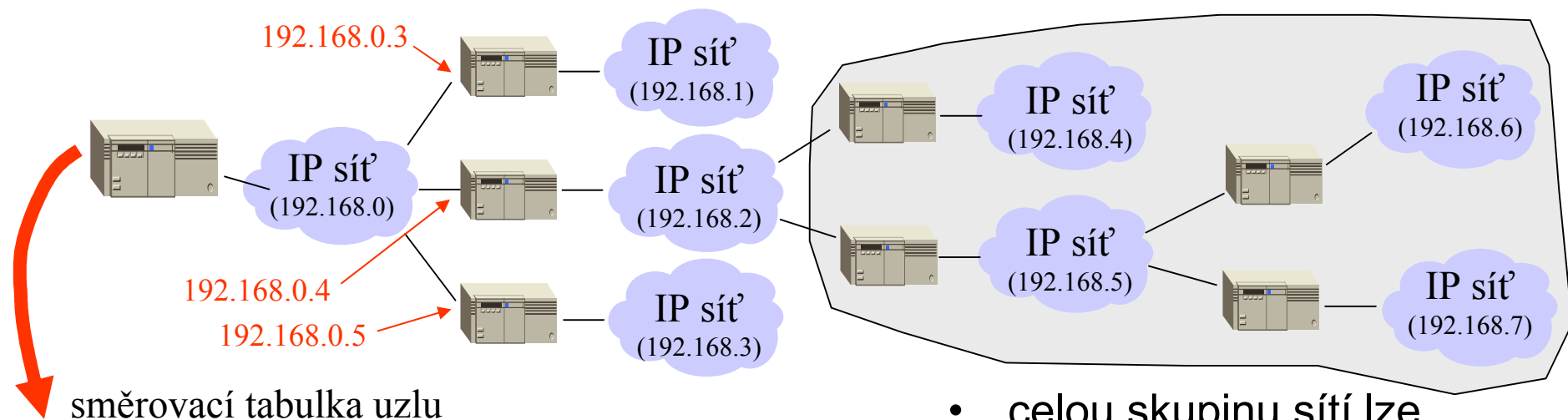


cílová síť/prefix	posílej přes
192.168.0/24	směřuj přímo
192.168.1/24	192.168.0.3
192.168.2/24	192.168.0.4
192.168.3/24	192.168.0.5
192.168.4/24	192.168.0.4
192.168.5/24	192.168.0.4
192.168.6/24	192.168.0.4
192.168.7/24	192.168.0.4

jsou to adresy nejbližšího přeskoku
(next hop)

- ve směrovací tabulce se nenachází úplná cesta k cíli, ale pouze "next hop"
 - **adresa nejbližšího směrovače**
- prefix v adrese cílové sítě odpovídá masce
 - "CIDR prefix" vyjadřuje počet jedničkových bitů masky

optimalizace směrovacích tabulek (agregace položek)



cílová síť/prefix	posílej přes
192.168.0/24	směřuj přímo
192.168.1/24	192.168.0.3
192.168.2/24	192.168.0.4
192.168.3/24	192.168.0.5
192.168.4/24	192.168.0.4
192.168.5/24	192.168.0.4
192.168.6/24	192.168.0.4
192.168.7/24	192.168.0.4

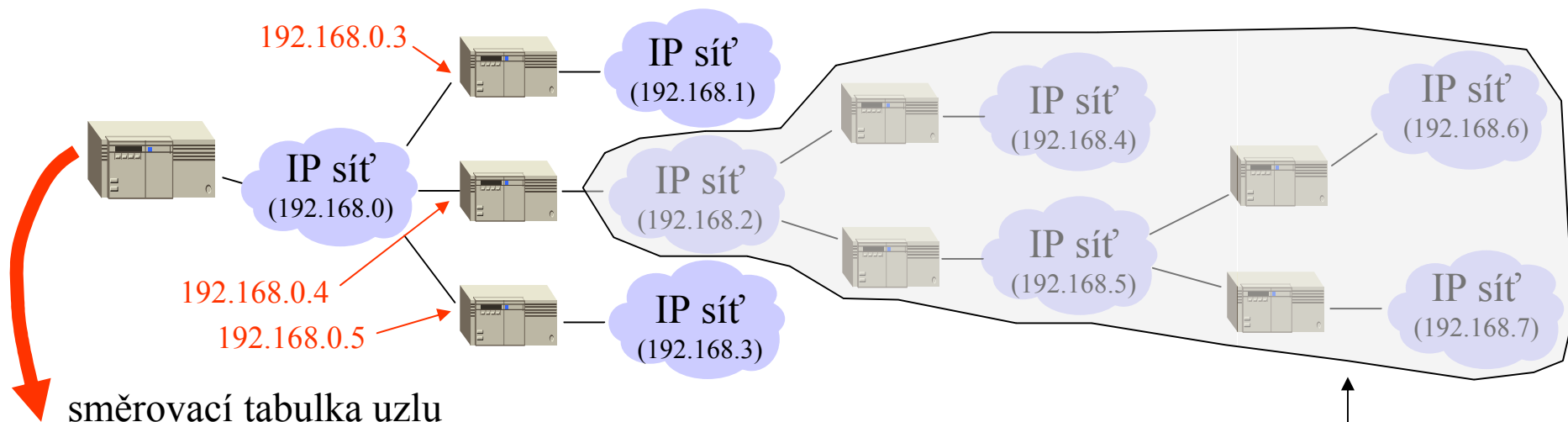
192.168.00000100
192.168.00000101
192.168.00000110
192.168.00000111
192.168.000001xx

22 bitů
24 bitů

- celou skupinu sítí lze sloučit (**agregovat**) do většího CIDR bloku

cílová síť/prefix	posílej přes
192.168.0/24	směřuj přímo
192.168.1/24	192.168.0.3
192.168.2/24	192.168.0.4
192.168.3/24	192.168.0.5
192.168.4/22	192.168.0.4

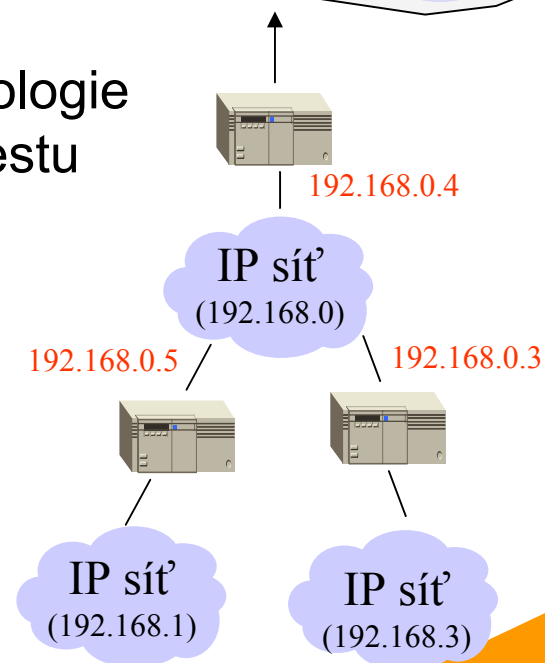
optimalizace směrovacích tabulek (implicitní cesty – default route)



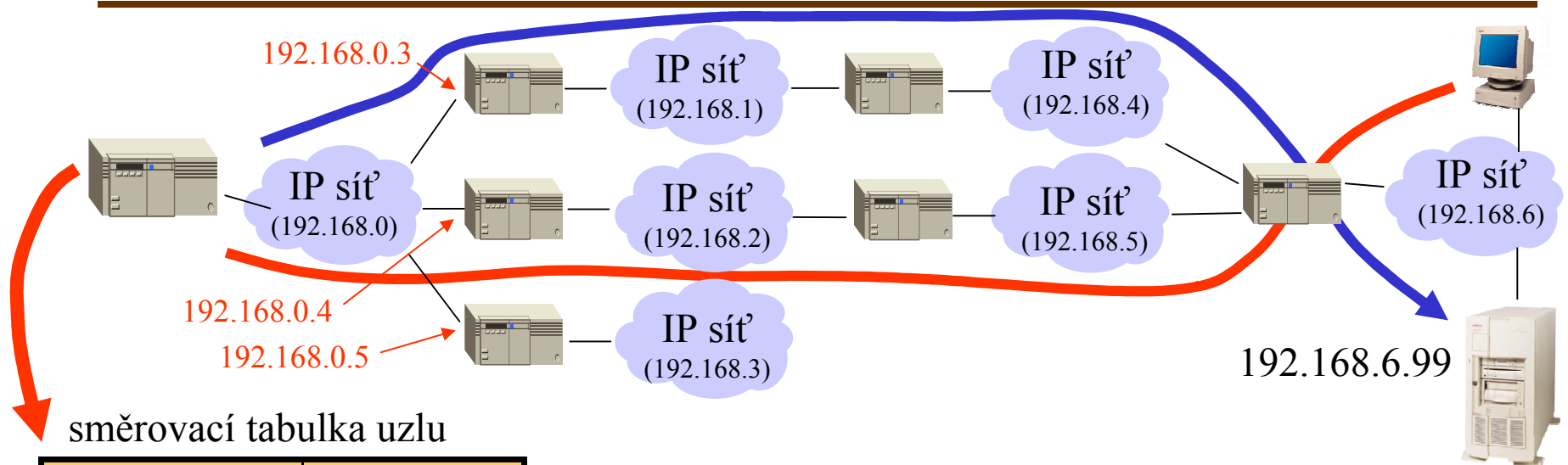
cílová síť/prefix	posílej přes
192.168.0/24	doruč přímo
192.168.1/24	192.168.0.3
192.168.2/24	192.168.0.4
192.168.3/24	192.168.0.5
192.168.4/ 22	192.168.0.4

- v případě stromovité topologie lze definovat implicitní cestu (default route), vedoucí "nahoru"

cílová síť/prefix	posílej přes
192.168.0/24	doruč přímo
192.168.1/24	192.168.0.3
192.168.3/24	192.168.0.5
všechno ostatní	192.168.0.4



host-specific route



směrovací tabulka uzlu

cílová síť/prefix	posílej přes
192.168.0/24	doruč přímo
192.168.1/24	192.168.0.3
192.168.2/24	192.168.0.4
192.168.3/24	192.168.0.5
....	
192.168.6/24	192.168.0.4
192.168.6.99/32	192.168.0.3

- pomocí masky (prefixu) lze do směrovacích tabulek zavést i specifické směrovací informace, týkající se jednotlivých uzlů
 - tzv. **host-specific route**
- lze to využít při redundantním připojení například pro odlišné směrování dat směřujících k nějakému serveru

"host-specific route" k uzlu 192.168.6.99

pravidla směrování

- "host-specific route" by měly být používány jen výjimečně
 - velmi zvětšují objemy směrovacích tabulek
 - musí se vyhodnocovat jako první
- snaha je rozhodovat při směrování podle příslušnosti cílového uzlu k určité síti
 - pak postačuje menší počet položek směrovacích tabulek
- agregace položek pomáhá snižovat objem směrovacích tabulek
 - pomáhá i používání "default route"
 - default route odpovídá prefixu 0
- pravidlo pro prohledávání směrovacích tabulek:
 - postupuj od nejvíce konkrétního k nejméně konkrétnímu
 - tedy: nejprve hledej položku s největším prefixem, postupuj k menším prefixům

příklad:

cílová síť/prefix	posílej přes
192.168.6.99/32	192.168.0.3
192.168.3/24	192.168.0.5
192.168.4/22	192.168.0.4
x/0 (ostatní)	192.168.0.1

← host-specific route

← default route

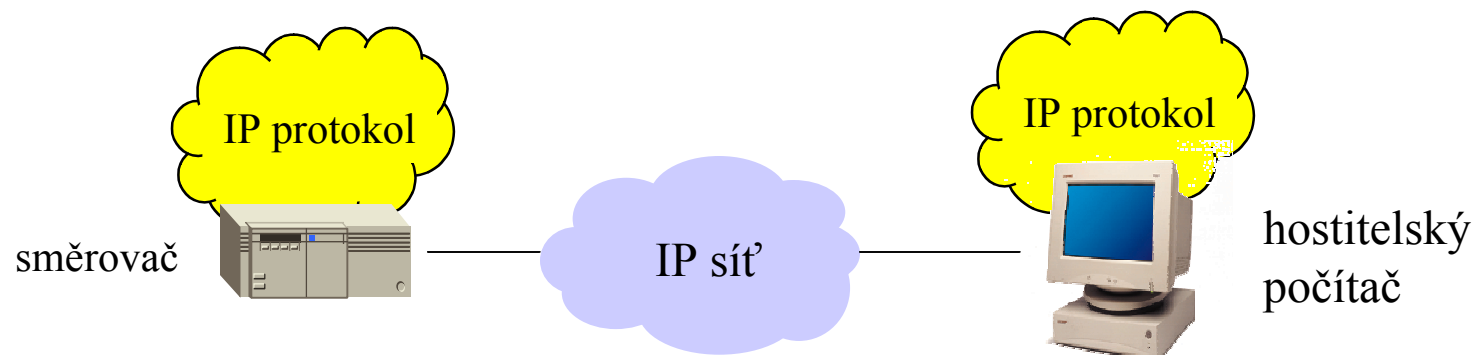


postup prohledávání

základní algoritmus směrování

- vezmi I_d (“plnou” IP adresu příjemce), a zjisti zda se příjemce nachází ve stejné síti jako ty ...
 - pokud ano, použij přímé doručování. Jinak
- začni prohledávat směrovací tabulku postupně podle velikosti prefixu
 - pokud se hodnota v prefixu právě prohledávané položky shoduje se stejnohlou částí I_d (příslušným počtem vyšších bitů), doručuj nepřímo dle této položky. Jinak pokračuj další položkou, pokud existuje ...
- existuje-li implicitní cesta (default route)
 - použij tuto cestu. Jinak ...
- skonči chybou
 - generuj ICMP zprávu "Destination Unreachable"

role hostitelských počítačů a směrovačů



- směrovače:

- účastní se všech činností v rámci směrování
 - včetně aktualizace směrovacích informací

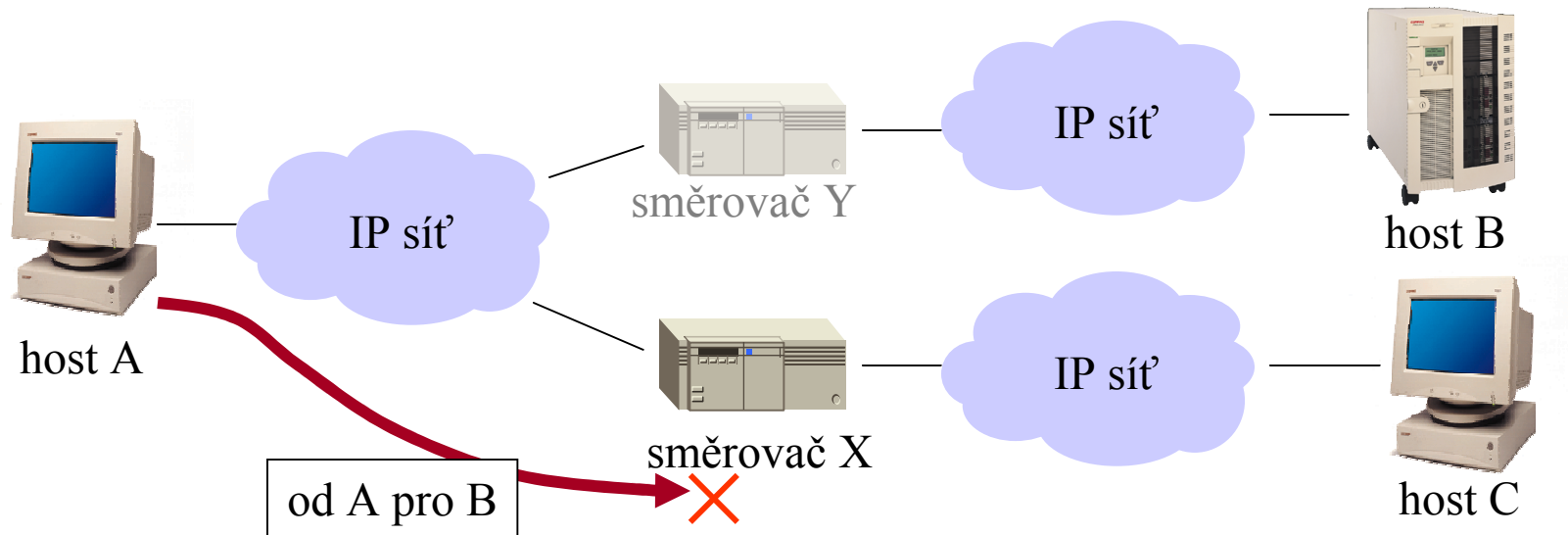
když se hostitelský počítač "chová špatně", směrovač jej "poučí" (poskytne mu správné směrovací informace)

- hostitelské počítače:

- také musí volit směr přenosu paketu
- vedou si směrovací informace ve svých směrovacích tabulkách
 - a využívají je – aplikují základní algoritmus směrování
- **ale neúčastní se aktualizace směrovacích informací !!!**

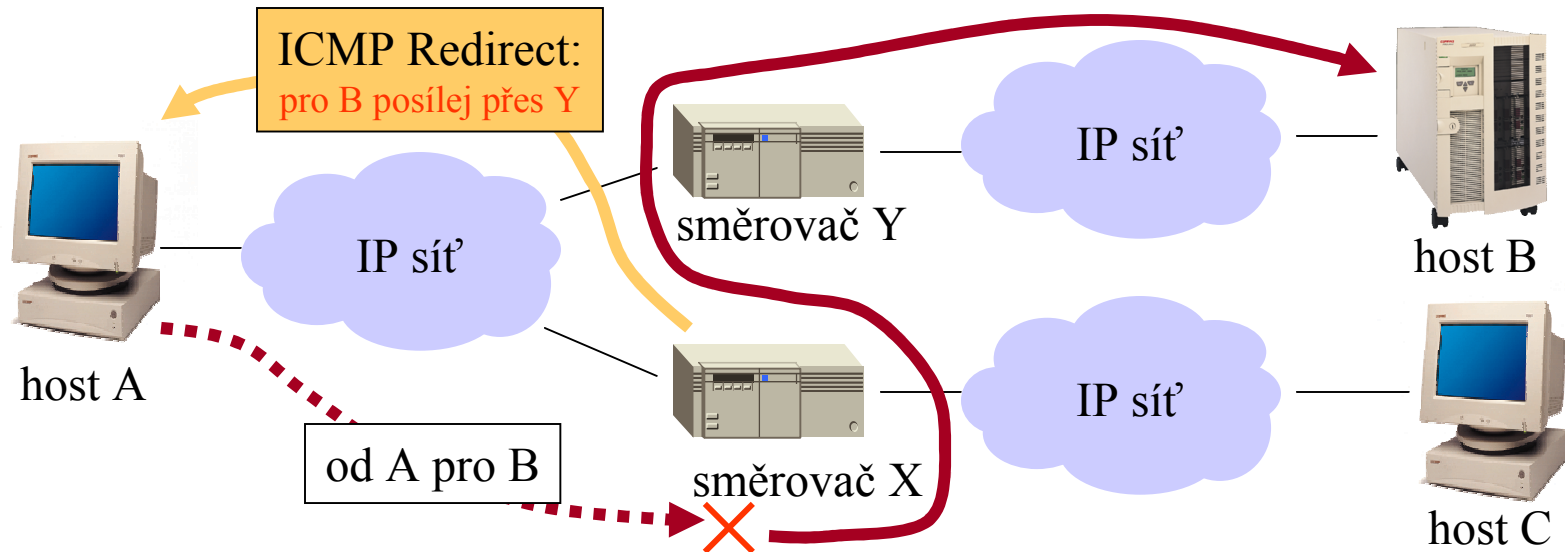


příklad



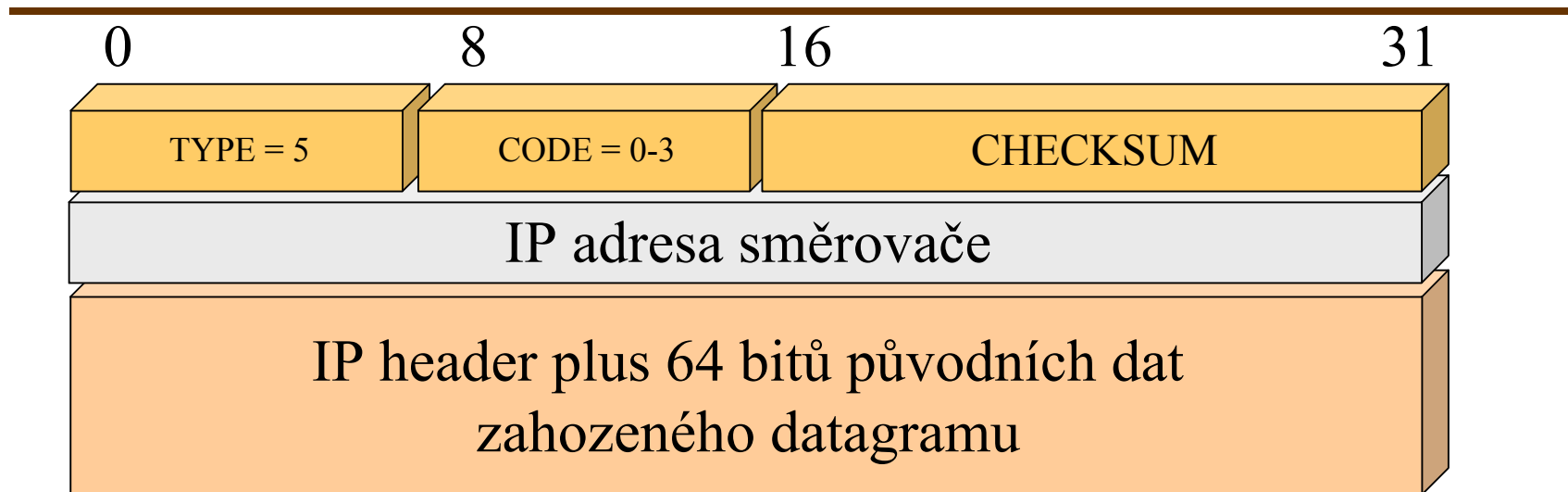
- "na počátku" musí každý hostitelský počítač znát alespoň jeden směrovač "vedoucí ven" ze sítě ve které se nachází
 - nechť host A zná směrovač X (ale nikoli směrovač Y)
- potřebuje-li host A poslat něco hostu B, pozná že jde o nepřímé doručování a pošle to směrovači X
- směrovač X pozná, že neleží na nejvhodnější cestě mezi hostem A a hostem B
 - upozorní hosta A na vhodnější (kratší) cestu – na existenci směrovače Y

ICMP Redirect



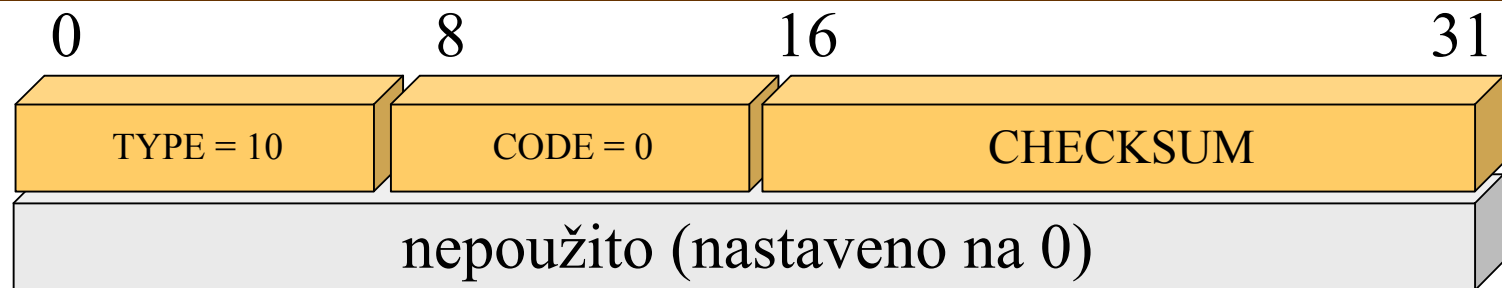
- směrovač X se postará o správné doručení IP datagramu k uzlu B
 - sám pošle data směrovači Y, ten se postará o doručení
- směrovač X pošle hostu A zprávu "ICMP Redirect" ve smyslu:
 - "datagramy pro uzel B příště posílej přes směrovač Y"
 - host A by se měl poučit
 - měl by si zanést směrovač Y do své směrovací tabulky a příště jej použít

ICMP Redirect



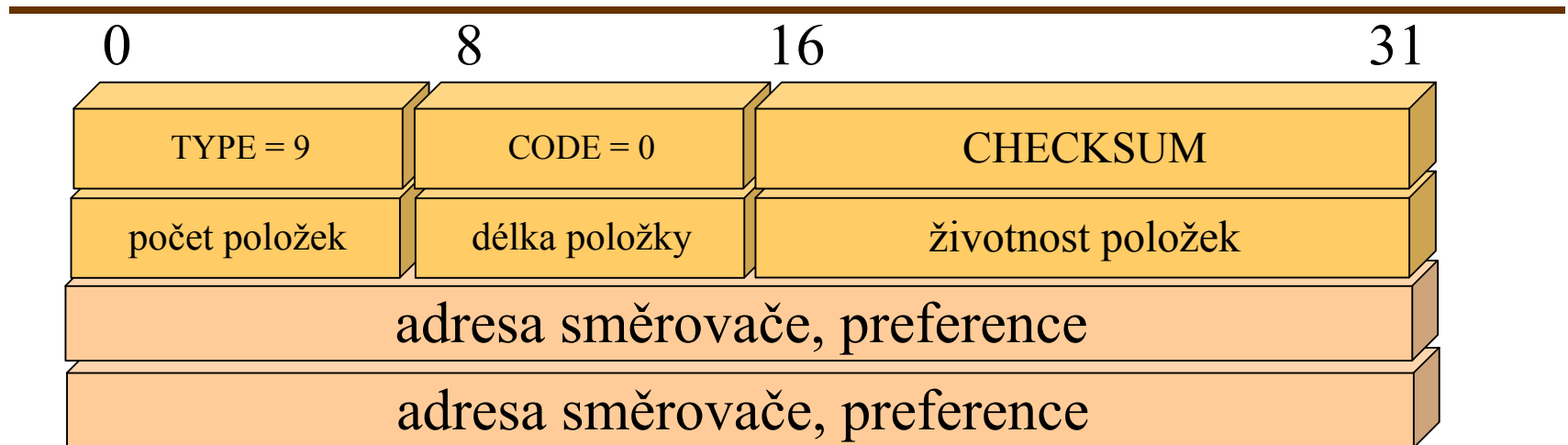
- jde o hlášení od směrovače, že existuje lepší cesta pro doručení IP datagramu a vede přes jiný směrovač
 - jeho IP adresa je uvedena
 - CODE=0: změň směrování pro síť, 1: změň směrování pro uzel
 - CODE=2: změň směrování pro síť pro daný typ služby, 3: dtto, uzel&služba
- odesílatel (hostitelský počítač) by měl zareagovat zanesením nového směrovače do své směrovací tabulky
 - pokud tak neučiní, nesprávně oslovený směrovač má právo jej znovu upozornit, ale nesmí jej odmítnout (musí vždy předat data správným směrem)

ICMP Router Solicitation



- jde o "dotaz do pléna": **jaké jsou tedy směrovače?**
 - ICMP zpráva je rozesílána pomocí IP broadcastu všem uzlům dané sítě
- odpověď přináší informaci o dostupných směrovačích v síti
 - (zřejmě) je brána první odpověď která dorazí, eventuelní neúplnost je řešena pomocí ICMP Redirect
- umožňuje to, aby hostitelské počítače nemusely "na počátku" znát žádný směrovač
- směrovače odpovídají na dotazy
 - ale samy také generují odpovědi (advertisement) v náhodném intervalu mezi 450 až 600 vteřinami

ICMP Router Advertisement

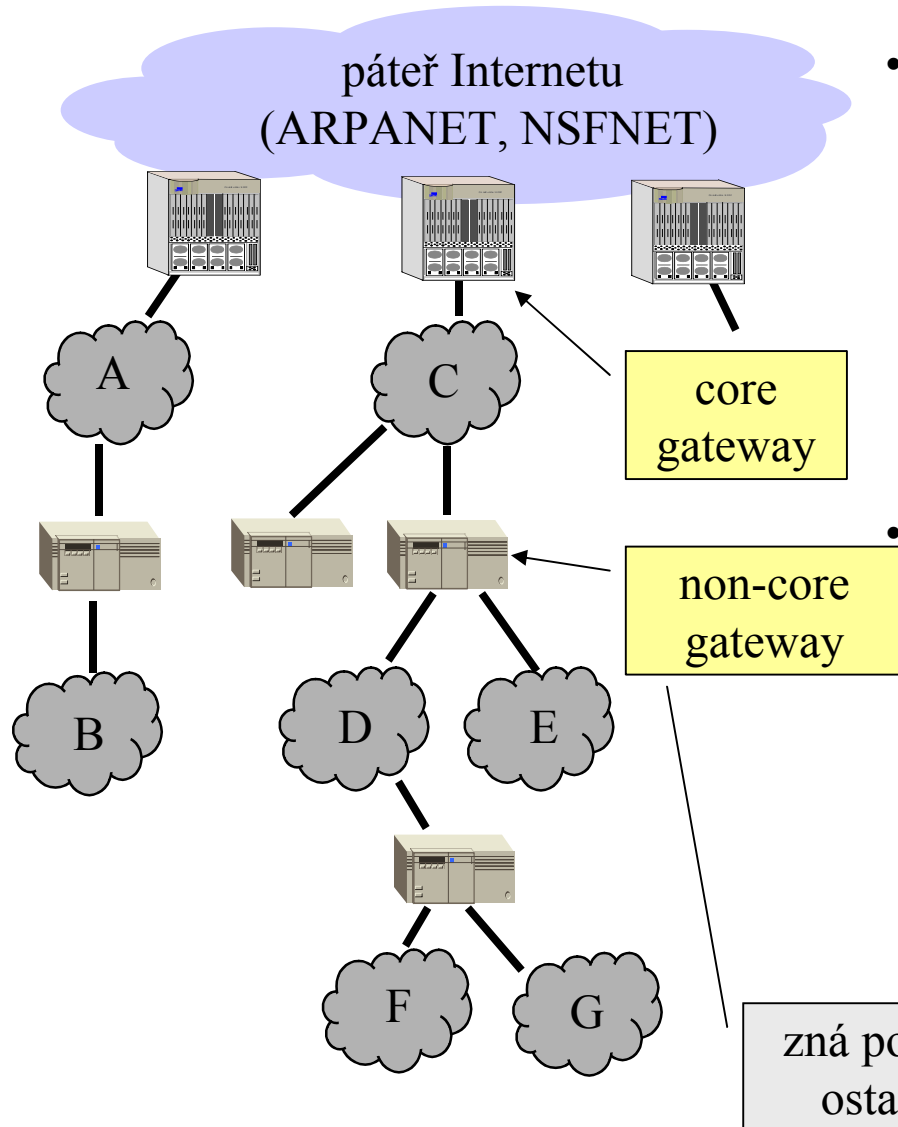


- jde o odpověď na Router Solicitation, nebo o samostatně generovanou "reklamu" (advertisement)
- preference umožňují příjemci stanovit, přes který směrovač vede implicitní cesta (default route)
 - životnost říká, jak dlouho má být záznam o směrovači ponechán ve směrovací tabulce příjemce

aktualizace směrovacích informací

- základní problém:
 - jak zajistit rychlou a správnou reakci na změny, tak aby s tím nebyla spojena příliš velká režie
 - navíc: jak to udělat v rozsahu dnešního Internetu?
- je třeba průběžně šířit aktualizací informace
 - ze kterých se průběžně vypočítávají údaje (next hop) ve směrovacích tabulkách
 - lze to řešit na principu "vector distance" nebo "link state"
- řešení tohoto problému se měnilo s vývojem Internetu
 - hlavně v důsledku jeho zvětšování
- zpočátku:
 - Internet byl malý, existovaly centrální směrovače s úplnou informací
- později:
 - vznikla 2-úrovňová struktura
 - core směrovače s úplnou informací
 - non-core směrovače s neúplnou informací
- ještě později
 - došlo k "dekompozici" Internetu
 - vzniku tzv. autonomních systémů (AS), které v sobě lokalizují detailní směrovací informace a nešíří je mimo sebe

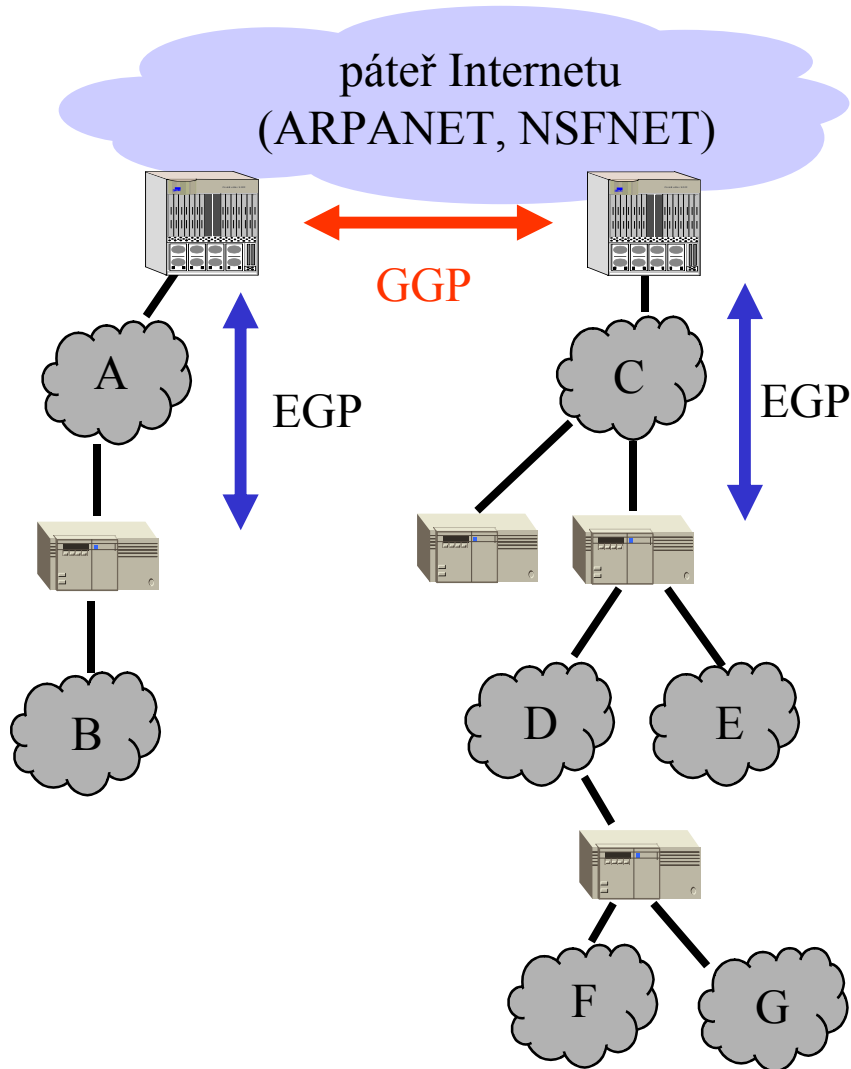
směrování v ranném Internetu



- existovala soustava tzv. core gateways (centrálních směrovačů), nacházejících se v páteřní části Internetu
 - tyto core gateways měly úplnou informaci o celé topologii Internetu
 - byly centrálně spravovány (pověřenou organizací)
- ostatní směrovače byly "non-core gateways"
 - pracovali s neúplnou informací o topologii Internetu
 - "znaly" jen sítě "pod sebou", provoz do ostatních sítí směrovaly přes implicitní cesty do core gateways
 - inzerovaly existenci "svých" sítí (sítí "pod sebou") směrem ke core

zná pouze cestu k sítím D,E, F a G,
ostatní posílá přes default route

směrování v ranném Internetu

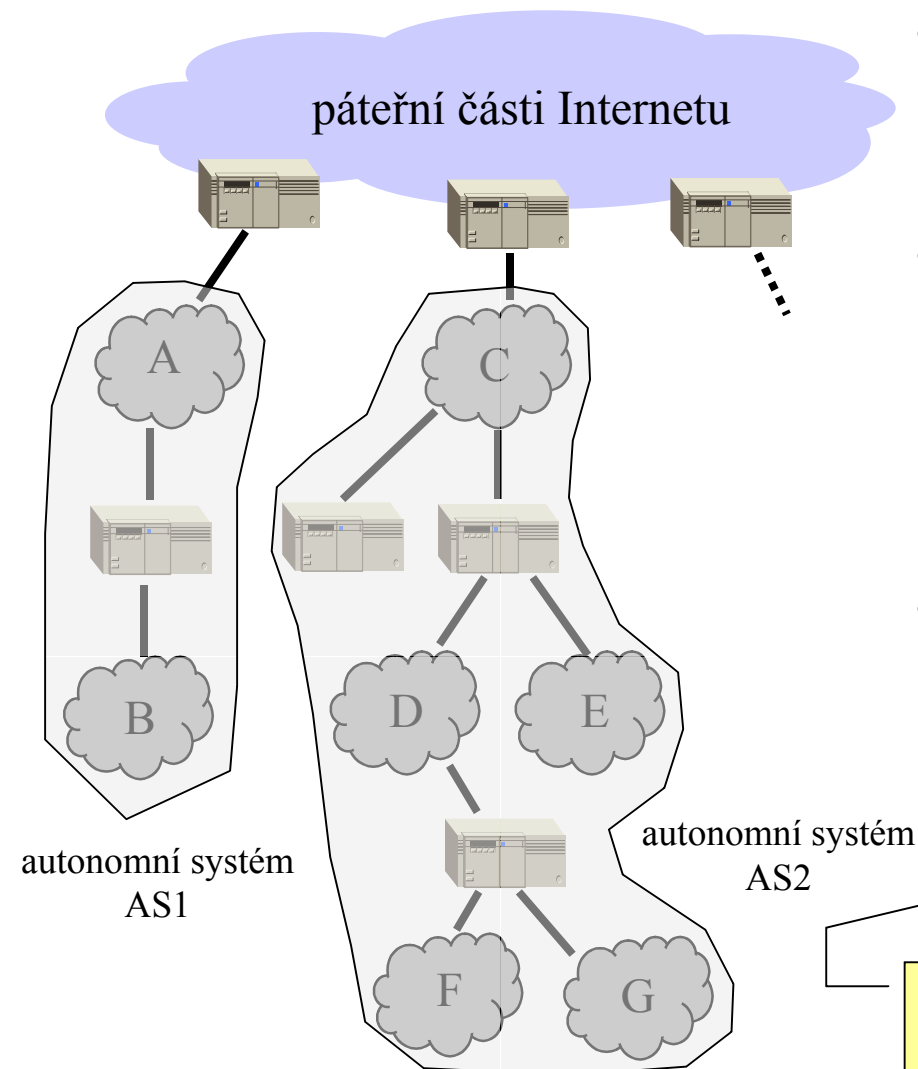


- pro vzájemnou komunikaci centrálních směrovačů ("core gateways") byl vytvořen protokol GGP
 - Gateway-to-Gateway Protocol
- pro komunikaci mezi centrálními a "ostatními" (vnějšími) směrovači byl vytvořen protokol EGP
 - Exterior Gateway Protocol
- terminologie:
 - původně se IP směrovačům říkalo "IP Gateways"
 - proto GGP a EGP
- problém tohoto řešení:
 - nebylo dostatečně škálovatelné

další vývoj – autonomní systémy

- s růstem Internetu se řešení s “core gateways” stalo neúnosné
 - úplná informace o celé topologii Internetu je příliš velká, režie na distribuci této informace mezi všemi “core gateways” neúnosná
- “core gateways” nešlo donekonečna “nafukovat”
 - muselo se najít jiné řešení
- souvislost:
 - Internet přešel do komerční sféry, “směrovací politika” jednotlivých částí Internetu již nemusela být stejná
 - bylo třeba vyhovět individuálním požadavkům jednotlivých providerům, požadavkům na peering,
- princip “jiného řešení”
 - “dekompozice” Internetu z hlediska směrování
 - detailní (“úplná”) směrovací informace nebude šířena po celém Internetu
 - resp. po páteřní části
 - ale zůstane lokalizována v určitých oblastech
 - bude šířena pouze uvnitř těchto oblastí, ne mimo ně
 - tyto oblasti budou šířit kolem sebe pouze mnohem “menší” informace o dostupnosti
- jde o tzv. **autonomní systémy**

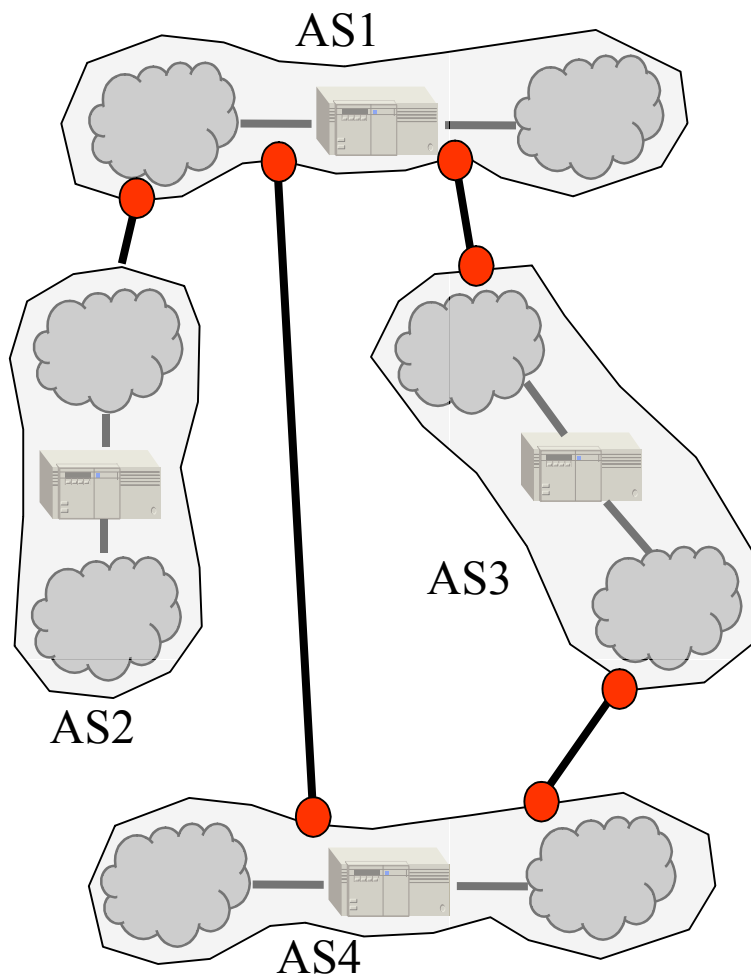
představa autonomního systému



- autonomní systém "navenek" neinformuje o své vnitřní struktuře
 - ani o detailních směrovacích informacích
- je "autonomní" v tom smyslu, že si může sám stanovit svou vlastní směrovací politiku
 - včetně toho, jakým způsobem je uvnitř AS řešena aktualizace směrovacích informací
- navenek autonomní systém zveřejňuje pouze informace o dostupnosti
 - ve smyslu:
 - AS1: "uvnitř mne se nachází sítě A až B"
 - AS2: "uvnitř mne se nachází sítě C až G"

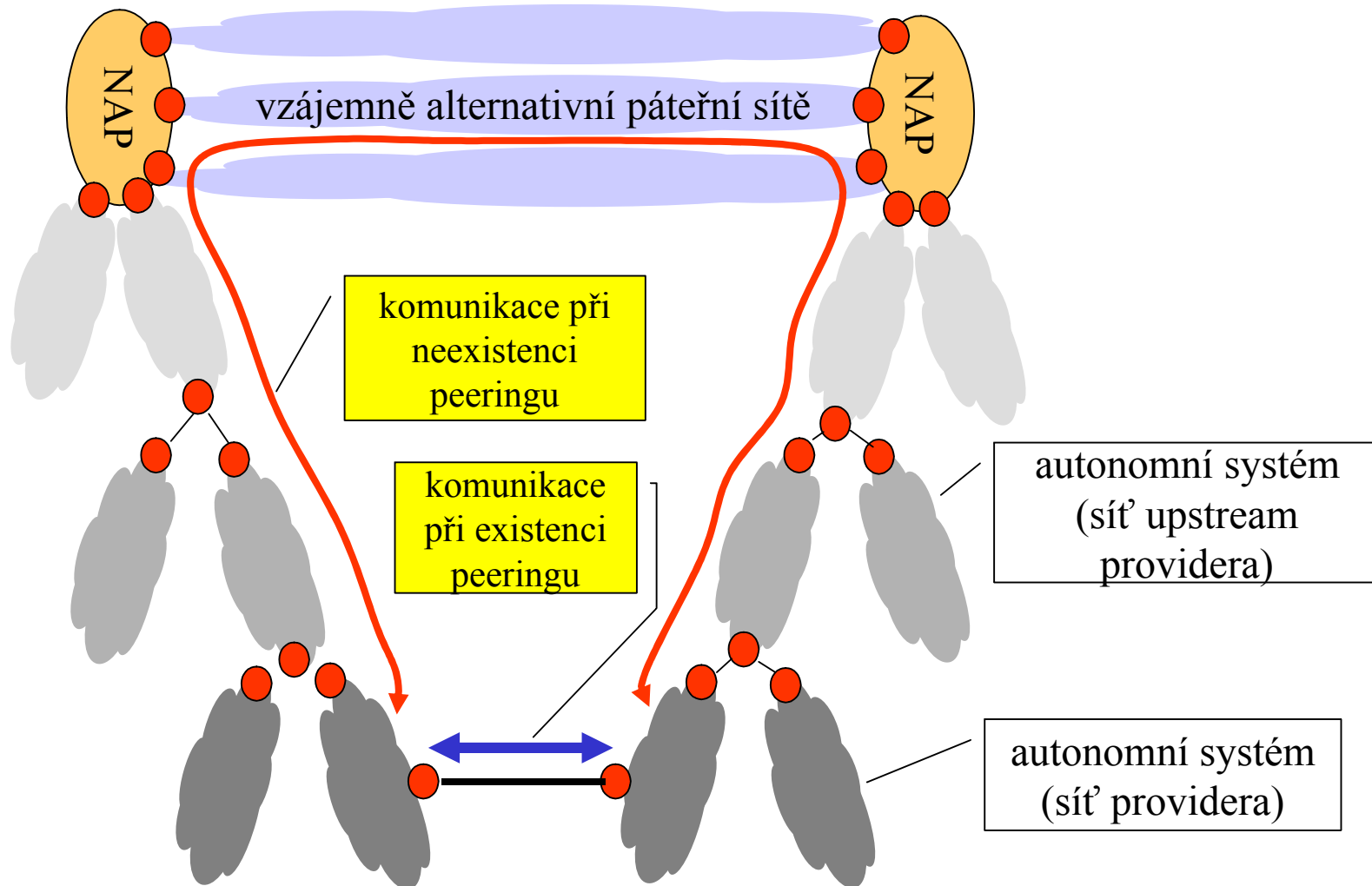
je to typicky "intervalová informace" (od-do), tvořená rozsahem IP adres (resp. CIDR bloků náležejících do AS)

představa autonomního systému



- každý autonomní systém má určitý (malý) počet vstupních/výstupních bodů
 - skrz tyto body se propojují s ostatními autonomními systémy
 - skrz tyto body si vyměňují informace o dostupnosti (o svém obsahu)
 - a také testují svou vzájemnou existenci
- původně musela být struktura autonomních systémů striktně stromovitá
 - dnes již nemusí
 - každý AS si může sám zvolit, jak ("kudy") chce komunikovat s jinými autonomními systémy
 - díky tomu je možný peering – přímé propojení autonomních systémů, obcházející implicitní propojení přes páteřní části

dnešní struktura Internetu



Exterior Gateway Protocols

- mezi autonomními systémy musí probíhat výměna informací
 - o dostupnosti, existenci, "navazování vzájemných vztahů", ...)
- k tomu jsou zapotřebí vhodné protokoly
- dříve se používal protokol EGP (Exterior Gateway Protocol)
 - byl šit na míru "centralizovanému Internetu", s jediným páteřním autonomním systémem
 - nepřipouštěl nic jiného než stromovou strukturu
 - nedokázal využít více alternativních "páteřních AS"
- dnes je "Exterior Gateway Protocols" generické označení pro všechny protokoly, které zajišťují komunikaci mezi AS
- dnes se používá modernější protokol **BGP** (Border Gateway Protocol)
 - napravuje nedostatky EGP
 - připouští obecné propojení autonomních systémů
 - ne pouze "do stromu"
 - umožňuje stanovit různá kritéria při volbě mezi alternativními směry
 - správce AS může stanovit priority, například v závislosti na rychlosti, kapacitě linek, spolehlivosti atd.
 - podporuje CIDR
 - dnes verze BGP-4

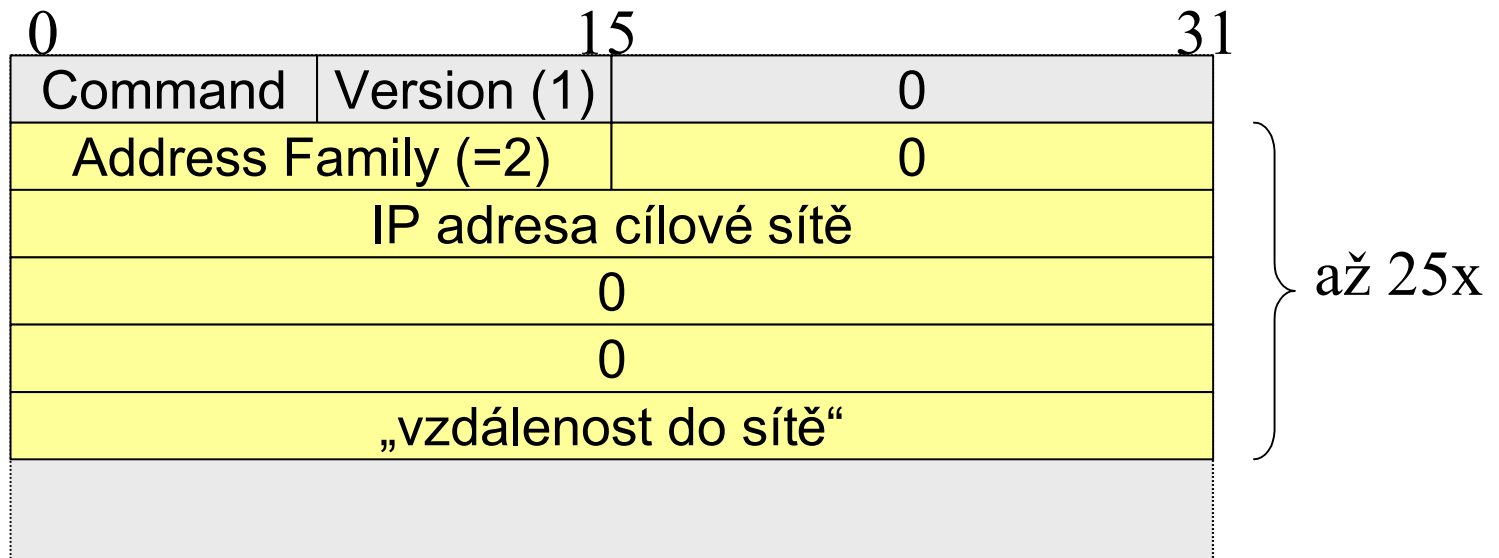
IGP – Interior Gateway Protocols

- připomenutí: uvnitř sebe sama si každý autonomní systém může řešit směrování tak jak uzná za vhodné
 - může aplikovat vlastní směrovací politiku
 - týká se to hlavně aktualizace směrovacích informací
- existuje více alternativních protokolů, které lze použít pro aktualizaci směrovacích informací uvnitř AS
 - obecně jsou označovány jako IGP (Interior Gateway Protocols)
- příklady protokolů IGP:
 - RIP (Routing Information Protocol)
 - pracuje na principu "vector distance"
 - vyvinut firmou Xerox ve středisku PARC, použit mnoha firmami, používal se již v původním ARPANETu
 - vhodný pro malé až střední sítě, ne pro velké
 - OSPF (Open Shortest Path First)
 - pracuje na principu "link state"
 - vhodný i pro větší sítě (větší autonomní systémy)

RIP – Routing Information Protocol

- je typu vector-distance
 - uzly si vyměňují aktualizace tvořené směrovým vektorem a jeho ohodnocením (vzdáleností k cíli)
 - metrika je pevná, a to počet přeskoků!!
- aktualizace (updaty):
 - se vysílají každých 30 sekund
 - když do 180 sekund nepřijde update od nějakého konkrétního (sousedního) routeru, jsou všechny cesty vedoucí přes tento router označeny jako nekonečně dlouhé.
 - po dalších 120 sekundách jsou odstraněny z tabulky (nastoupí jakýsi garbage collector).
- výpočet cest je distribuovaný
 - každý počítá kousek
 - algoritmus probíhá trvale, nikdy nekončí!!!
 - každý je závislý na ostatních, chyba jednoho ovlivňuje druhé
- aktualizace (updaty):
 - posílají se jen k přímým sousedům (routerům)
 - každý uzel se od svých sousedů dozvídá jen o dostupnosti cílových sítí (a metrice), ne o dalším routování za svými sousedy (nevidí dál než ke svým sousedům)
 - nemá informace o celé topologii!!!
 - obsahují údaje o dostupnosti ostatních uzlů z daného uzlu (s jakou cenou)
 - v zásadě jde o obsah celé směrovací tabulky
 - alternativní cesty nejsou uvažovány, (cesty se stejným ohodnocením jsou ignorovány)
 - aby se zabránilo oscilacím, RIP nahradí již existující cestu pouze cestou, která má nižší metriku!!! (nestačí stejná)!!!

protocol RIP

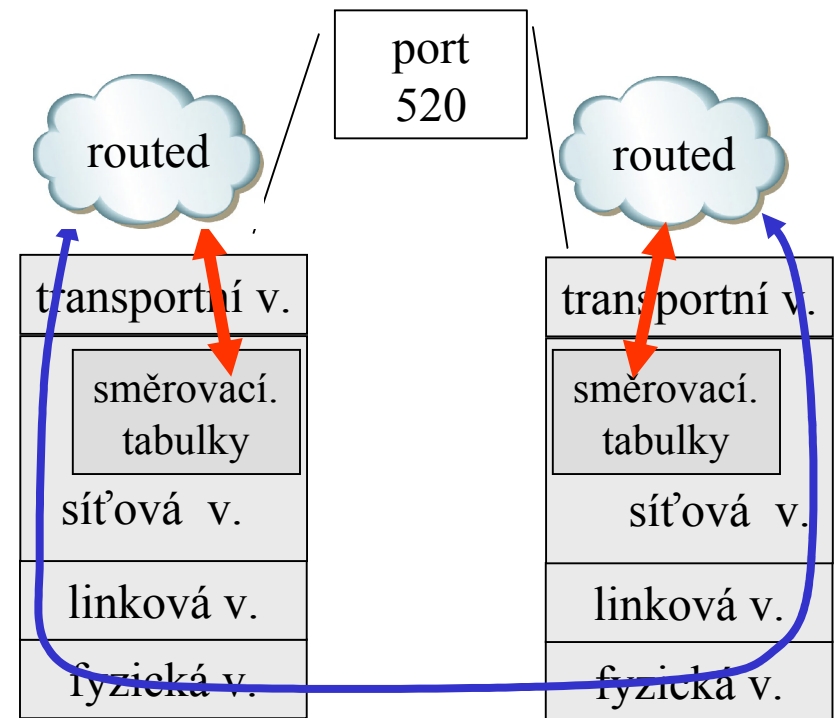


Zpráva RIP-u obsahuje:

- pole **Command**, obsahuje buď výzvu k zaslání routovacích informací, nebo odpověď na tuto výzvu.
- pole **Address Family** (= 2 pro RIP)
- pole "vzdálenost" musí obsahovat číslo od 1 do 15, zatímco 16 je považována za nekonečno.

protocol RIP – představa fungování

- RIP je v TCP/IP je implementován jako aplikace (na aplikační úrovni)
 - jako démon "routed"
 - běží nad UDP
 - sedí na portu 520 (well-known portu)
 - velikost každého RIP paketu je max. 512 bytů
- výhoda:
 - je to jednoduché, nemusí se konfigurovat
 - stačí spustit démona routed, ten již nastaví směrovací tabulky



RIP2, RIPng

- v roce 1993 byl původní RIP updatován
 - na verzi RIP-2
- nové vlastnosti/schopnosti:
 - podpora IP adres s maskami a CIDR
 - "Next Hop Specification"
 - v RIP záznamu je explicitně uvedena IP adresa směrovače, přes který vede spojení do cílové sítě
 - zvyšuje efektivnost
 - umožňuje směrovat provoz i přes směrovače, které nepodporují RIP
 - autentizace
 - ochrana proti útokům skrze falešné RIP zprávy
 - "Route Tag"
 - další informace o inzerované cestě
 - použití multicastu pro rozesílání zpráv (RIP Response)
 - zasílají se na adresu 224.0.0.9, která je vyhrazena pro RIP
 - všechny uzly "v dosahu" musí podporovat multicast
- RIP-2 může koexistovat s RIP-1
 - RIP-2 vkládá svá nová data do nevyužitých částí zpráv RIP-1
- v roce 1997 byl RIP upraven i pro IPv6
 - RIPng, RIPv6
 - umožňuje používat IP adresy verze 6
 - má jiný formát zpráv

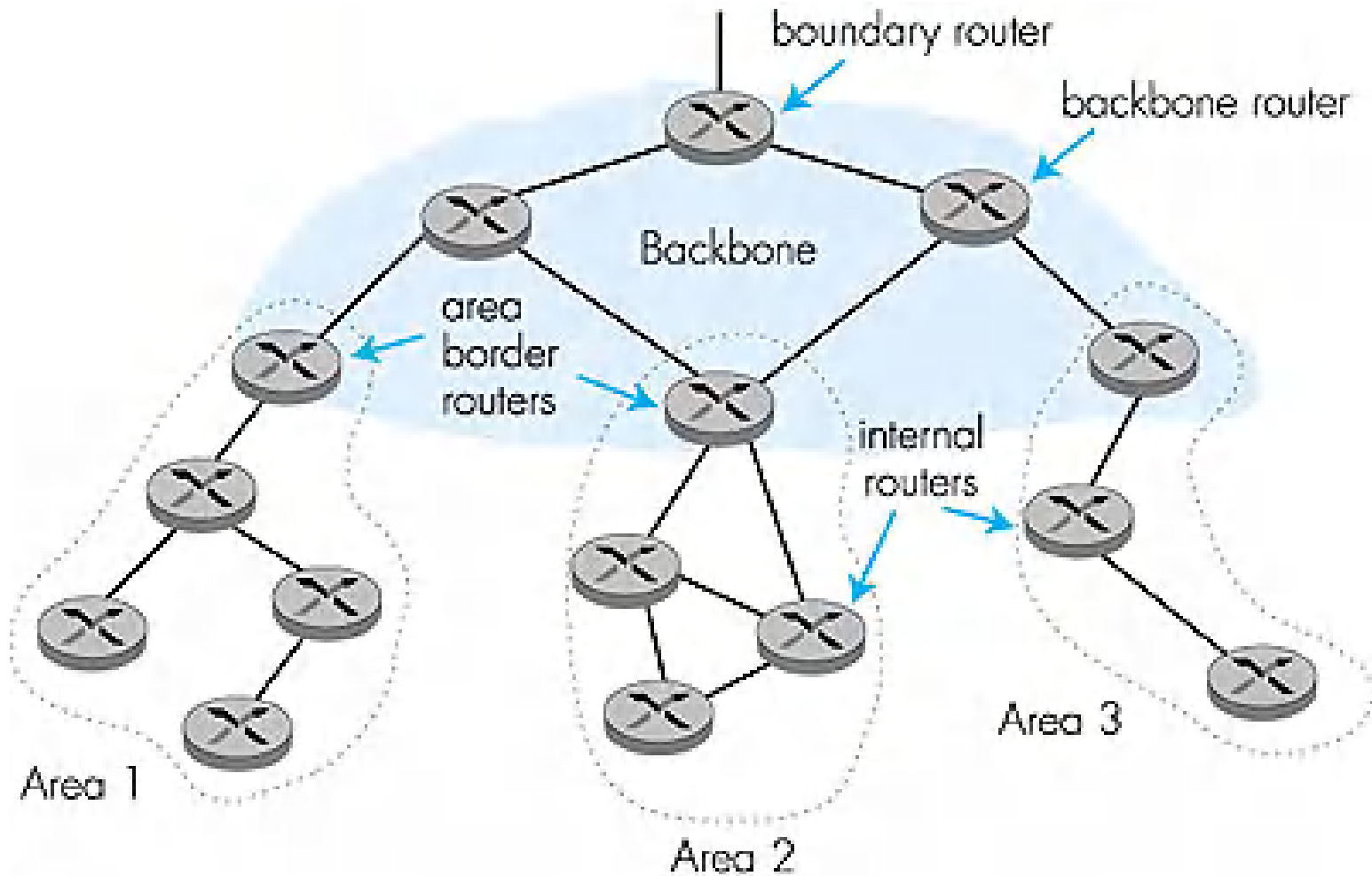
protokol OSPF (Open SPF)

- je "otevřenou verzí staršího protokolu SPF (Shortest Path First)
 - jeho specifikace jsou veřejně přístupné, pochází od IETF
- je typu link-state
 - každý uzel testuje dostupnost svých sousedů
 - stav linky
 - každý uzel sestavuje "link state paket", ve kterém uvede údaje o dostupnosti svých sousedů
 - stav linky a její ohodnocení
 - tyto pakety jsou rozesílány všem uzlům v síti/soustavě sítí
 - stačí ale jen při změně nějakého údaje !!!!
 - jinak pro osvěžení každých 30 minut
- všechny uzly v síti mají úplnou informaci o jednotlivých spojích a mohou si vypočítat optimální cesty
 - každý počítá "za sebe", chybou ovlivní jen sebe sama
- OSPF podporuje alternativní cesty
 - umožňuje definovat různé cesty pro různé druhy provozu
 - podporuje load balancing
- OSPF podporuje další "dekompozici"
 - umožňuje rozdělení sítě na menší "areas", které jsou analogické autonomním systémům v tom, že jejich topologie není šířena mimo danou "area"
 - minimalizuje to objemy aktualizací

protokol OSPF – oblasti

- jde o vlastnost, která zvětšuje "dosah" OSPF
 - umožňuje větší škálovatelnost, tj. realizovat AS
- celý autonomní systém se rozdělí na (disjunktní) oblasti
 - jedna se prohlásí za páteřní (backbone)
- směrovače v oblastech se rozdělí na
 - interní
 - zajišťují směrování v rámci oblasti, mají stejné informace (navzájem)
 - páteřní (Backbone)
 - zajišťují směrování v rámci páteřní oblasti
 - na rozhraní (Area Border)
 - patří současně do oblasti i do páteře, vyměňují informace mezi nimi
 - hraniční (Boundary)
 - v páteřní oblasti, vyměňují směrovací informace s jinými AS

příklad: hierarchické OSPF



fungování OSPF

- každý OSPF směrovač si udržuje:
 - databázi přímých sousedů
 - každý ji má jinou
 - udržuje aktuální pomocí HELLO paketů, které pravidelně posílá svým sousedům
 - každých 10 sekund
 - každý směrovač si udržuje "topologickou databázi"
 - databázi s údaji o topologii celé sítě
 - všichni (v oblasti) by ji měli mít stejnou
 - pomocí této databáze počítá "nejkratší" cesty
 - směrovací tabulky
 - používají se pro samotné směrování IP paketů
- komunikace mezi OSPF směrovači probíhá pomocí OSPF paketů
 - vkládají se přímo do IP paketů
 - Protocol No. 89
 - existuje 5 druhů zpráv:
 - Hello
 - Database Description
 - Link State Request
 - Link State Update
 - Link State Acknowledgement

fungování OSPF

- "nový" směrovač:
 - nejprve zjistí, jaké má sousedy
 - řeší se jinak v prostředí s broadcastem, bez broadcastu a na dvoubodových spojích
 - s každým sousedem si synchronizuje svou topologickou databázi
 - pomocí příkazů
 - Database Description
 - Link State Request
 - Link State Acknowledgement
 - po úspěšné synchronizaci oba směrovače ve dvojici "ohlásí světu své sousedství"
 - pomocí broadcastu oba rozešlou všem ostatním směrovačům v síti tzv. LSA (Link State Advertisement)
 - informaci o existenci spojení (vazby, hrany) mezi nimi
 - pomocí příkazů Link State Update
- "již fungující" směrovač
 - trvale monitoruje dostupnost svých přímých sousedů
 - pomocí HELLO paketů, každých 10 sekund
 - pokud není změna:
 - každých 30 minut opakuje všem své "sousedství"
 - rozesílá LSA s údaji o dostupnosti souseda, pomocí broadcastu
 - pokud je změna
 - okamžitě informuje o změně pomocí LSA (Link State Advertisement)
 - OSPF příkaz "Link State Update"
- LSA se šíří jako inteligentní broadcast
 - fakticky jako záplava (záplavové směrování), s eliminací duplicitních paketů