

3. Síťová vrstva – Směrování

PB156: Počítačové sítě

Eva Hladká

Fakulta informatiky Masarykovy univerzity

jaro 2010

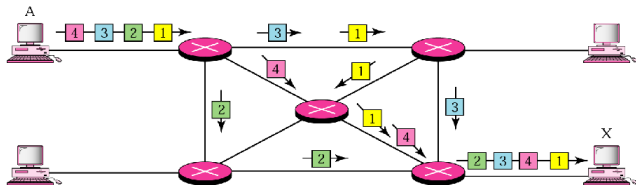
Struktura přednášky

- 1 Směrování obecně
- 2 Směrování
 - Základní přístupy
- 3 Směrovací algoritmy
- 4 Distribuované směrování
 - Distance Vector
 - Link State
 - Link State vs. Distance Vector
- 5 Hierarchie směrování
 - Původní představy
 - Autonomní systémy
 - Autonomní systémy – směrování
- 6 Multicastové směrování – IP Multicast
 - Motivace
 - IP Multicast
 - Protokoly

- 1 Směrování obecně
- 2 Směrování
 - Základní přístupy
- 3 Směrovací algoritmy
- 4 Distribuované směrování
 - Distance Vector
 - Link State
 - Link State vs. Distance Vector
- 5 Hierarchie směrování
 - Původní představy
 - Autonomní systémy
 - Autonomní systémy – směrování
- 6 Multicastové směrování – IP Multicast
 - Motivace
 - IP Multicast
 - Protokoly

Směrování obecně

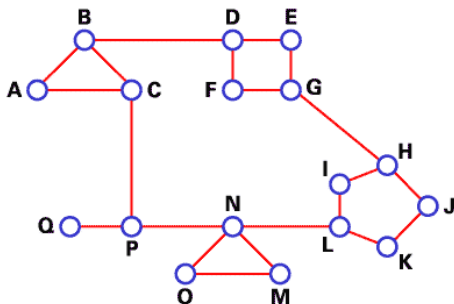
- Internet na L3 – datagramový přístup k přepínání paketů
 - data vyšších vrstev umísťována do datagramů
 - datagramy (fragmenty) putují sítě nezávisle



- **směrování (Routing)** = proces nalezení cesty mezi dvěma komunikujícími uzly
 - cesta musí splňovat určité omezující podmínky
 - ovlivňující faktory:
 - *statické*: topologie sítě
 - *dynamické*: zátěž sítě

Matematický pohled

- na směrování lze nahlížet jako na problém teorie grafů
- síť reprezentována grafem, kde:
 - uzly reprezentují směrovače (identifikovány svými IP adresami)
 - hrany reprezentují vzájemné propojení směrovačů (linku)
 - ohodnocení hran = cena komunikace
 - *cíl*: nalezení minimální cesty v grafu mezi libovolnými dvěma uzly



Cena komunikace

Určení ceny (ohodnocení) linky – *metrika*:

- všechny linky mají stejnou cenu (např. 1)
 - minimalizace ceny = minimalizace počtu skoků
 - nejjednodušší, nejčastěji využívané
- cena linky = převrácená hodnota kapacity ($1/\text{prenosova_kapacita}$)
 - 10Mb linka má 100x vyšší cenu než 1Gb linka
- cena linky = zpoždění linky
 - 250ms satelitní spojení má 10x vyšší cenu než 25ms pozemní linka
- cena linky = využití linky
 - linka s 90% využitím má 10x vyšší cenu než linka s 9% využitím
 - může způsobit oscilace (nezbytné tlumení)
- cena linky = reálná cena (platba) za využití linky
 - staticky přiřazeno administrátorem
- atd.

- 1 Směrování obecně
- 2 Směrování**
 - Základní přístupy
- 3 Směrovací algoritmy
- 4 Distribuované směrování
 - Distance Vector
 - Link State
 - Link State vs. Distance Vector
- 5 Hierarchie směrování
 - Původní představy
 - Autonomní systémy
 - Autonomní systémy – směrování
- 6 Multicastové směrování – IP Multicast
 - Motivace
 - IP Multicast
 - Protokoly

Směrování

- úkolem směrování je:
 - vyhledávat optimální směrovací trasy
 - kriteriem optimality je metrika
 - dopravit datový paket určenému adresátovi
- zpravidla se nezabývá celou cestou paketu
 - směrovač řeší jen jeden krok – komu paket předat jako dalšímu
 - někomu „blíže“ cíli
 - tzv. *hop-by-hop*
 - ten pak rozhoduje, co s paketem udělat dál

Směrování (Routing) vs. zasílání (Forwarding)

- *směrování*
 - společná činnost směrovačů (globální)
 - proces nalezení/vytváření a údržby směrovacích tabulek
- *zasílání*
 - lokální proces – každý směrovač samostatně
 - představuje proces průchodu paketů směrovačem
 - zaslání paketu na vybrané rozhraní směrovače (dle cílové adresy)
 - vyžaduje přístup ke směrovací tabulce

Směrovací tabulky

- základní datovou strukturou je *směrovací tabulka (routing table)*
 - sada ukazatelů, podle kterých se rozhoduje, co udělat s kterým paketem
 - obsahují cesty k „prefixům“
 - počáteční IP adresa a blok
 - agregace záznamů – hledá se nejdelší prefix, který vyhovuje požadavku
 - existence více vyhovujících prefixů ⇒ použije se nejdelší
 - tzv. *Longest-prefix Match Algorithm*

	Mask	Destination address	Next-hop address	Interface
	/8	14.0.0.0	118.45.23.8	m1
Host-specific →	/32	192.16.7.1	202.45.9.3	m0
	/22	193.14.4.0	84.12.6.20	m1
	/24	193.14.5.0	84.78.4.12	m2
Default →	/0	/0	145.11.10.6	m0

Problém globálního pohledu

- globální znalost topologie celé sítě je problematické
 - je složité ji získat
 - když už se to podaří, není aktuální
 - musí být lokálně relevantní
- lokální představu o topologii reprezentuje směrovací tabulka
- rozpor mezi lokální a globální znalostí může způsobit
 - cykly (černé díry)
 - oscilace (adaptace na zátěž)

Směrování – základní přístupy

Členění dle způsobu vytvoření/udržování směrovací tabulky:

- *statické (neadaptivní)*
 - administrátorem ručně editované záznamy
 - směrovač nemůže vytvářet alternativní cesty, pokud se nastavená cesta přeruší
 - jednodušší, málo flexibilní
 - vhodné pro statickou topologii
 - *Otázka:* Používá se v Internetu?
- *dynamické (adaptivní)* – reagují na změny v síti
 - složité (většinou distribuované) algoritmy
 - (většinou) nutnost pravidelné aktualizace směrovacích tabulek
 - nutnost existence protokolu pro aktualizaci směrovacích tabulek
 - možnost dočasné nekonzistence
 - nezaručuje pořadí doručení
 - např.
 - *centralizované* – vše řídí centrum
 - *izolované* – každý sám za sebe
 - *distribuované* – kooperace uzlů

Dynamické směrování – centralizované směrování

- v síti je *Routing Control Center (RCC)*
 - každý směrovač mu posílá zprávy o své situaci (stavu)
 - RCC informace sbírá, vypočte optimální cesty a rozešle směrovačům jejich tabulky
- výhody:
 - globální informace (\Rightarrow optimální řešení)
 - ulehčení práce směrovačů
- nevýhody:
 - špatně škáluje – nelze využít pro velké sítě (nemožnost získání globální informace)
 - pomalé
 - při výpadku centra se přestane aktualizovat

Dynamické směrování – izolované směrování

- neposílají se žádné informace o stavu sítě, každý se rozhoduje sám za sebe
- příklady:
 - *náhodná procházka* – paket pošle do náhodně vybrané linky
 - vysoká robustnost
 - „*horký brambor*“ (*hot potatoe*) – paket pošle do linky s nejkratší frontou
 - forma náhodné procházky (\Rightarrow vysoká robustnost)
 - *záplava (flooding)* – paket pošle do všech linek kromě té, po níž přišel
 - enormní zátěž sítě – obrovská režie, nutno řešit cykly
 - mimořádně robustní – pokud cesta existuje, vždy ji najde
 - dokonce tu nejlepší možnou (zkouší totiž všechny)
 - *zpětné učení (backward learning)* – učí se z procházejících paketů
 - do paketu se zapisuje vzdálenost, kterou urazil
 - směrovač se dozví, že příchozí linkou vede cesta k odesílateli nanejvýš dané délky

Dynamické směrování – distribuované směrování

- směrovací informace si vyměňují sousedé či malé skupiny směrovačů
- na základě periodicky šířených informací se (podle určitého algoritmu) vypočítávají mapy sítě
- mezi směrovači musí být dohoda o implementaci určitého *směrovacího algoritmu*
- dostatečně pružné a robustní, vhodné i pro rozlehlé sítě
- standardní přístup ke směrování v síti Internet

Směrování – další možná členění

distribuované	vs.	centralizované
"krok za krokem"	vs.	zdrojové
deterministické	vs.	stochastické
jedno	vs.	více cestné
dynamický	vs.	statický výběr cest
INTERNET		

Směrovací algoritmy – funkce

- zprostředkovávají funkcionalitu směrování
 - proces vytvoření a údržby směrovacích tabulek
 - zahrnuje výběr komunikační cesty
 - vlastní doručení dat
- rozdělení dle *okamžiku* rozhodování:
 - při uzavírání spojení (= vytváření okruhu)
 - spojované služby, virtuální kanály
 - při příchodu paketu
 - nespojované služby, datagramy
- rozdělení dle *místa* rozhodování:
 - jediný uzel \Rightarrow centralizované algoritmy
 - každý uzel \Rightarrow distribuované algoritmy
- definice přesných pravidel komunikace a formátu zpráv nesoucích směrovací informace (pro určitý algoritmus) \Rightarrow *směrovací protokol*

Směrovací algoritmy – požadované vlastnosti

Žádané vlastnosti směrovacího algoritmu:

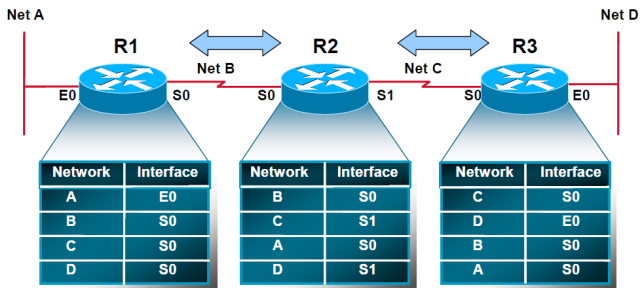
- správnost
- jednoduchost
- efektivita a škálovatelnost
 - minimalizace množství řídicích informací ($\approx 5\%$ provozu!)
 - minimalizace velikosti směrovacích tabulek
- robustnost a stabilita
 - nezbytný je distribuovaný algoritmus
- spravedlivost (fairness)
- optimálnost
 - „Co je to nejlepší cesta?“

Distribuované směrování – základní přístupy

Třídy distribuovaných směrovacích protokolů (dle charakteru směrovací informace):

- *Distance Vector (DV)* – Bellman-Fordův algoritmus
 - sousední směrovače si v pravidelných intervalech či při topologické změně (např. výpadek zařízení) vyměňují kompletní kopie svých směrovacích tabulek
 - na základe obsahu přijatých updatů si pak doplňují nové informace a inkrementují své *distance vektor číslo*
 - metrika udávající počet hopů k dané síti
 - čili „*všechny informace jen svým sousedům*“
- *Link State (LS)*
 - jednotlivé směrovače si zasílají pouze informace o stavu linek, na něž jsou bezprostředně připojeny
 - udržují si tak kompletní informace o topologii dané sítě – zařízení jsou si vědoma všech ostatních zařízení na síti
 - pak se počítá nejkratší cesta
 - čili „*informace o svých sousedech všem*“

Distance Vector I.



- směrovač si udržuje všechny známé routy v tabulce ve formě uspořádaných trojic (N, G, D) , kde:
 - N ... cílová síť
 - G ... adresa následujícího směrovače
 - D ... vzdálenost do cílové sítě (metrika)
- tabulky se upravují tak, aby se směrovalo nejkratší cestou
- problémy: pomalá konvergence, příliš mnoho režijních dat

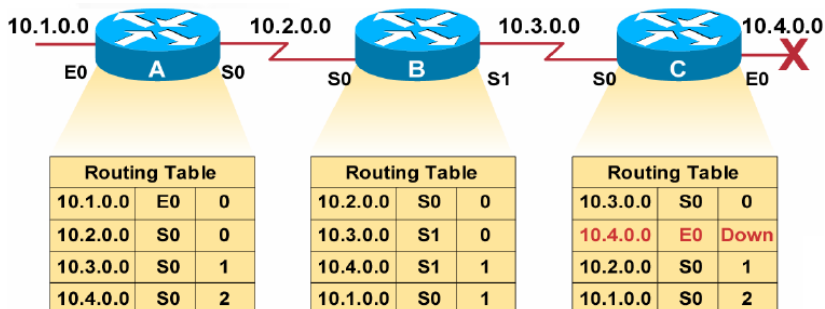
Distance Vector II.

Algoritmus

- Předpoklad:
 - každý směrovač zná pouze cestu a cenu ke svým sousedům
- Cíl:
 - v každém směrovači směrovací tabulka pro každý cíl
- Idea:
 - řekni sousedům svou představu směrovací tabulky
- Inicializace:
 - sousedé: známá cena
 - Distance Vector = $\langle cil, cena \rangle$
 - ostatní: nekonečno
 - resp. hodnota definovaná jako nekonečno (pro RIP např. 16)
- Aktualizace:
 - pokud je cesta v získaném DV zvětšená o cenu cesty k danému sousedovi lepší než stávající uložená, aktualizuj tabulku

Distance Vector III.

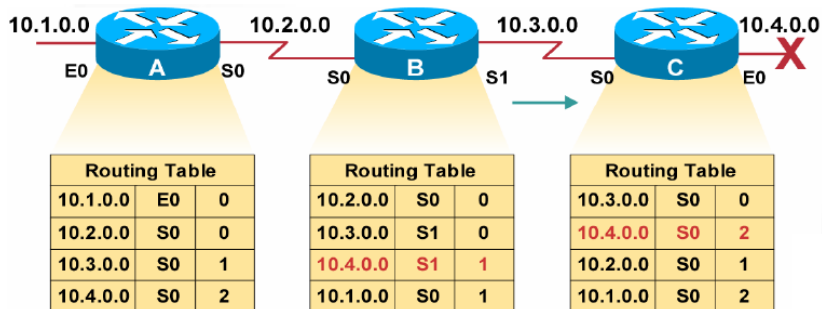
Ilustrace problému pomalé konvergence



- pomalá konvergence zapříčiní vznik nesprávných údajů ve směrovacích tabulkách

Distance Vector III.

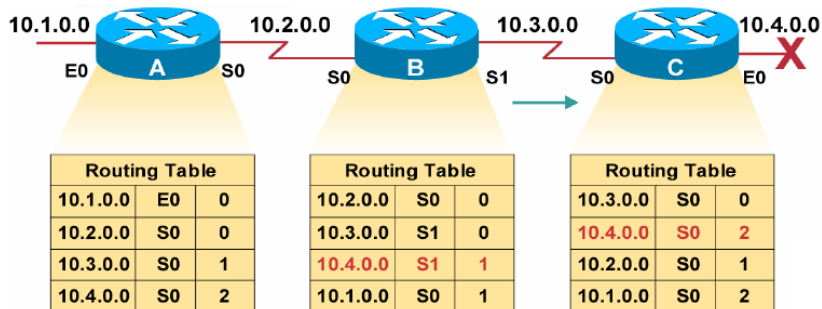
Ilustrace problému pomalé konvergence



- směrovač C usoudí, že nejlepší cesta do sítě 10.4.0.0 je přes směrovač B

Distance Vector III.

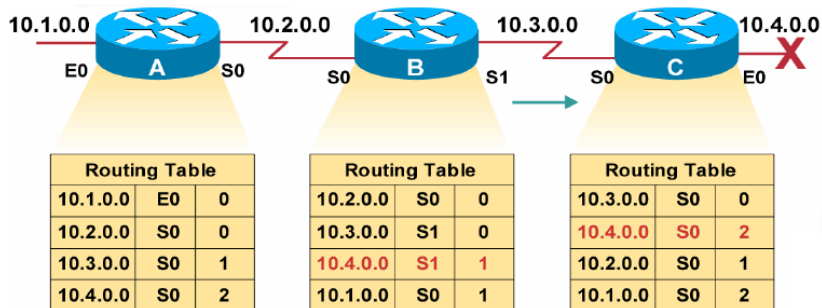
Ilustrace problému pomalé konvergence



- směrovač A opraví svojí směrovací tabulku – chybně

Distance Vector III.

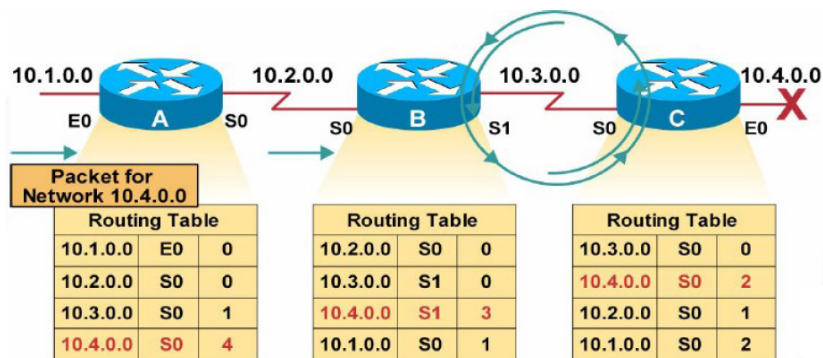
Ilustrace problému pomalé konvergence



- metrika pro síť 10.4.0.0 roste do nekonečna (v rámci RIP do 16)

Distance Vector III.

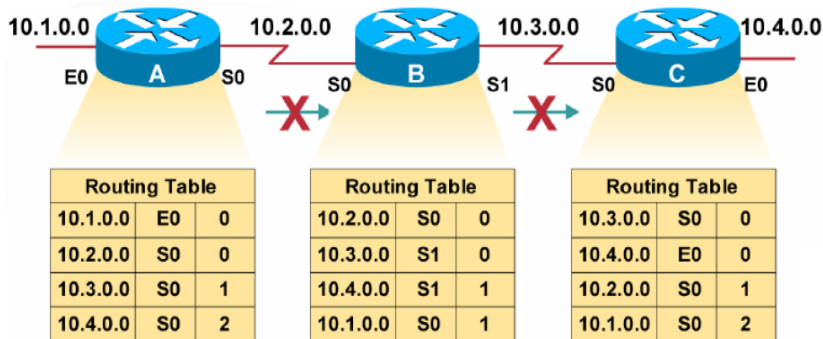
Ilustrace problému pomalé konvergence



- Důsledek: vznik směrovací smyčky
 - paket pro síť 10.4.0.0 skáče mezi routery B a C

Distance Vector III.

Ilustrace problému pomalé konvergence

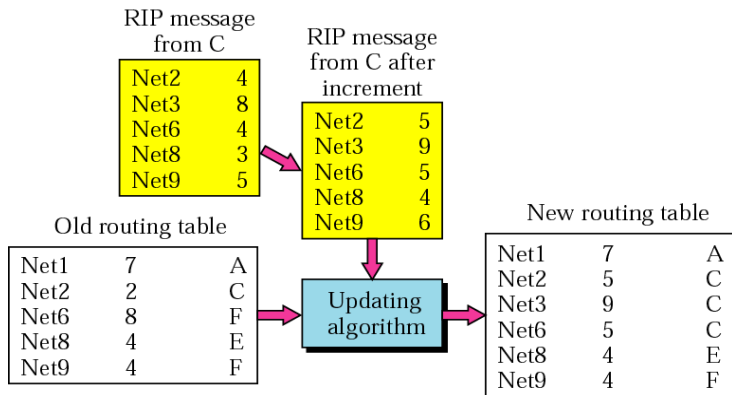


- Řešení: *dělení horizontu*
 - směrovač nesdílí cestu zpět uzlu, od kterého se o ní dozvěděl
 - problém zůstává ve složitějších topologiích (navržena řada rozšíření)

Distance Vector IV. – protokol RIP I.

- hlavní představitel DV směrování
 - RIPv1 (RFC 1058)
 - RIPv2 (RFC 1723) – přidává např. autentizaci směrovacích informací
- sítě identifikovány s využitím mechanismu CIDR
- jako metrika se využívá počet hopů
 - přenos paketu mezi 2 sousedními směrovači má délku 1
 - nekonečno = 16
 - \Rightarrow nelze použít pro sítě s minimálním počtem hopů mezi libovolnými dvěma směrovači > 15
- směrovače zasílají informaci každých 30 sekund
 - triggered update při změně stavu hrany
 - časový limit 180s (detekce chyb spojení)
- použití:
 - vhodné pro malé sítě a stabilní linky
 - není příliš vhodný pro redundantní sítě

Distance Vector IV. – protokol RIP II.



Net1: No news, do not change

Net2: Same next hop, replace

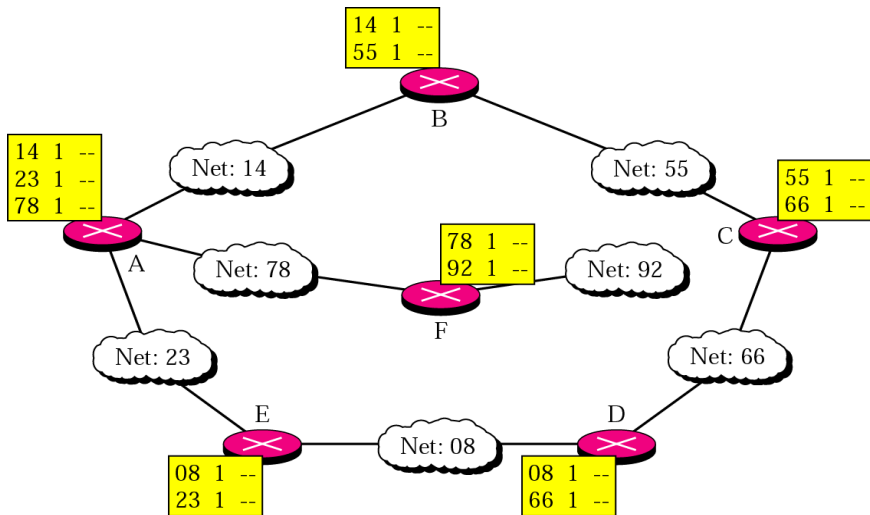
Net3: A new router, add

Net6: Different next hop, new hop count smaller, replace

Net8: Different next hop, new hop count the same, do not change

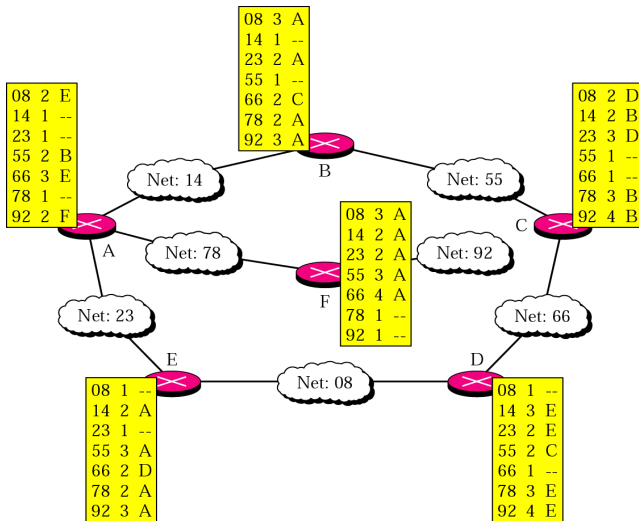
Net9: Different next hop, new hop count larger, do not change

Distance Vector IV. – protokol RIP III.



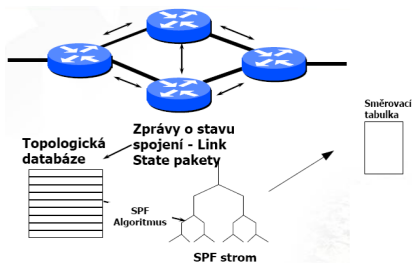
Obrázek: RIP – příklad: iniciální stav tabulek.

Distance Vector IV. – protokol RIP IV.



Obrázek: RIP – příklad: finální stav tabulek.

Link State I.



- směrovače si zasílají pouze informaci o stavu linek, na něž jsou bezprostředně připojeny
- získají tím kompletní mapu sítě
 - pak si počítají nejkratší cesty (např. s využitím Dijkstrova algoritmu)
 - při každé změně stavu linek
- směrovače testují pouze dosažitelnost svých bezprostředních sousedů
- výhoda: zaručená a rychlá konvergence, vhodné i pro rozsáhlé sítě
- nevýhoda: složitější algoritmus \Rightarrow větší nároky na CPU a paměť směrovače

Link State II.

Algoritmus

- Předpoklad:
 - každý směrovač zná pouze cestu a cenu ke svým sousedům
- Cíl:
 - v každém směrovači směrovací tabulka pro každý cíl
- Idea:
 - šíří se topologie, cesty si počítají směrovače samy
 - fáze 1: šíření topologie (broadcast)
 - fáze 2: výpočet nejkratší cesty – (Dijkstra)
 - směrovače si udržují databázi stavů linek a periodicky posílají LS pakety svým sousedům
 - obsah LS paketu: identifikátor uzlu, cena spojů k sousedům, pořadové číslo, doba platnosti
 - každý směrovač přeposílá LS pakety dále (kromě toho, od něžž informaci dostal)

Link State III. – Výpočet nejkratších cest

Dijkstrův algoritmus

„klasický“ algoritmus hledání nejkratší cesty

- hledá nejkratší cesty z jednoho vrcholu do všech ostatních

Nechť

- N je množina všech uzlů v grafu (síti)
- $l(i, j)$ označuje nezápornou cenu hrany (spoje (i, j))
- s je aktuální (zdrojový) uzel
- M množina uzlů, které již byly navštíveny
- $C(n)$ cena cesty z s do n ; ∞ pokud cesta neexistuje

Link State III. – Výpočet nejkratších cest

Dijkstrův algoritmus – pseudokód

```
M = {s}
for each n in N\M
  C(n) = l(s,n)
  while (N != M)
    přidej w do M
      tak, že C(w) je minimální pro všechna w z (N\M)
    for each n in N\M
      C(n) = min(C(n), C(w)+l(w,n))
```

- ilustrace výpočtu: <http://www.unf.edu/~wkloster/foundations/DijkstraApplet/DijkstraApplet.htm>
- animace:
<http://www.cse.yorku.ca/~aaw/HFHuang/DijkstraStart.html>
- více viz *PA165: Grafy a sítě*

Link State IV. – protokol OSPF

- *Open Shortest Path First*
- nejpoužívanější LS protokol současnosti
- metrika: *cena (cost)*
 - číslo (v rozsahu 1 až 65535) přiřazené ke každému rozhraní směrovače
 - čím menší číslo, tím má cesta lepší metriku (bude tedy preferována)
 - standardně je ke každému rozhraní přiřazena cena automaticky odvozená z šířky pásma daného rozhraní
 - $cost = 100000000 / bandwidth$ (bw v bps)
 - možno ručně měnit
- rozšíření:
 - autentizace zpráv
 - směrovací oblasti – další úroveň hierarchie
 - load-balancing – více cest se stejnou cenou

Link State vs. Distance Vector

Link State

- *Složitost:*
 - každý uzel musí znát cenu každé linky v síti $\Rightarrow O(nE)$ zpráv
 - změnu ceny některé z linek potřeba vypropagovat na *všechny* uzly
- *Rychlost konvergence:*
 - $O(n^2)$ alg., zasílá $O(nE)$ zpráv
 - trpí na oscilace
- *Robustnost:*
 - špatně fungující/kompromitovaný směrovač může šířit nesprávné informace jen o k němu přímo připojených linkách
 - každý směrovač si přepočítává směrovací tabulky sám za sebe \Rightarrow odděleno od vlastního šíření informací \Rightarrow forma robustnosti
- *Použití:*
 - vhodné i pro rozsáhlé sítě

Distance Vector

- *Složitost:*
 - po změně ceny některé z linek je toto zapotřebí vypropagovat jen *nejbližšímu sousedovi*; dále se propaguje jen tehdy, pokud daná změna znamená změnu stromu nejkratších cest
- *Rychlost konvergence:*
 - může konvergovat pomaleji než LS
 - problémy se směrovacími cykly, *count-to-infinity* problém
- *Robustnost:*
 - nesprávný výpočet je postupně šířen sítí \Rightarrow může znamenat „zmatení“ ostatních směrovačů a nesprávné vypočtené směrovací tabulky
- *Použití:*
 - vhodné jen pro menší sítě

- 1 Směrování obecně
- 2 Směrování
 - Základní přístupy
- 3 Směrovací algoritmy
- 4 Distribuované směrování
 - Distance Vector
 - Link State
 - Link State vs. Distance Vector
- 5 Hierarchie směrování**
 - Původní představy
 - Autonomní systémy
 - Autonomní systémy – směrování
- 6 Multicastové směrování – IP Multicast
 - Motivace
 - IP Multicast
 - Protokoly

Kde se nyní nacházíme?

- máme vybudovaný Internet
 - složený z mnoha internetů
- umíme identifikovat jednotlivé sítě/uzly
 - s využitím notace CIDR
 - cesty ve směrovacích tabulkách agregovány
- umíme směrovat data mezi sítěmi
 - libovolné dva uzly mohou komunikovat
 - pro směrování využít LS nebo DV algoritmus
- *Kde je problém?*
 - obrovský rozsah Internetu \Rightarrow nutnost správy obrovských směrovacích tabulek
 - problém se správou – „Kdo je zodpovědný za který kus sítě?“

Původní představy aneb Jak běžel „směrovací“ čas I.

- v počátcích velmi malé sítě
 - každý počítač na síti zná cestu ke všem ostatním
 - aneb „každý uzel zná celý Internet“
 - pro rozsáhlejší sítě neúnosné (rozsah tabulek, udržování vzájemné konzistence)
- přesun směrovací znalosti na hraniční uzly sítí (brány/směrovače)
 - aneb „každá brána zná celý Internet“
 - pro rozsáhlejší sítě stále neúnosné (rozsah tabulek, udržování vzájemné konzistence)
- ⇒ *hierarchické členění Internetu*
 - každá brána zná cesty jen do k ní přidružených podsítí (bezprostřední okolí); pro ostatní využita implicitní (default) brána
 - lokální působnost směrovacích informací (menší rozsah tabulek, jen lokální udržování vzájemné konzistence)
 - na nejvyšší úrovni (jediná) páteřní síť
 - páteřní brány musí mít „úplnou“ znalost celého Internetu

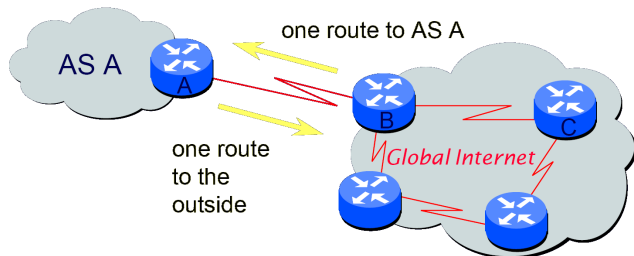
Původní představy aneb Jak běžel „směrovací“ čas II.

- nedostatky prvotního návrhu hierarchického členění:
 - organizace předávání směrovacích informací páteřním branám
 - „Jak je propagovat ze „zanořených“ sítí obecně náležejících různým organizacím?“
 - nutnost využívání jednotných mechanismů směrování v rámci celé sítě
 - včetně stejné metriky
- ⇒ rozšíření hierarchického členění na koncepci tzv. **autonomních systémů (AS)**
 - *základní myšlenka*: vzájemně propojené sítě, které spadají pod společnou správu, budou tvořit jediný autonomní systém, za který plně odpovídá jeho provozovatel
 - zůstává nutnost jednotného způsobu vzájemného předávání směrovacích informací mezi jednotlivými autonomními systémy
 - ⇒ v rámci svého AS má každý možnost zajistit si přenos a aktualizaci směrovacích údajů podle svého, ale navenek musí všichni postupovat jednotně

Autonomní systémy

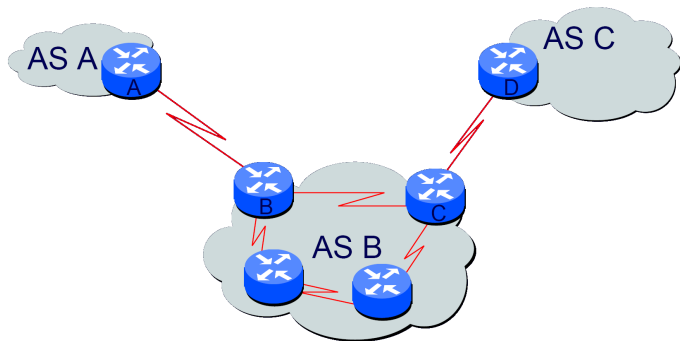
- cílem rozdělení Internetu na *autonomní systémy* je
 - snížení směrovací reže
 - jednodušší směrovací tabulky, snížení množství vyměňovaných směrovacích informací, atp.
 - zjednodušení správy celé sítě
 - správa jednotlivých internetů různými organizacemi
- autonomní systémy = domény
 - každému AS/doméně přiřazen 16bitový identifikátor
 - *Autonomous System Number (ASN)* – RFC 1930
 - přiřazuje organizace *ICANN (Internet Corporation For Assigned Names and Numbers)*
 - odpovídají administrativním doménám
 - sítě a směrovače uvnitř jednoho AS spravovány jednou organizací
 - např. CESNET, PASNET, ...
 - dělení v závislosti na způsobu připojení AS do sítě:
 - *Stub AS*
 - *Multihomed AS*
 - *Transit AS*

Autonomní systémy – *Stub AS*



- autonomní systém A je tzv. *stub AS*
 - je připojen pouze k jednomu dalšímu AS
- směrovač A (tzv. *hraniční směrovač*) je v rámci AS A výchozí směrovač pro všechny sítě ležící mimo AS A

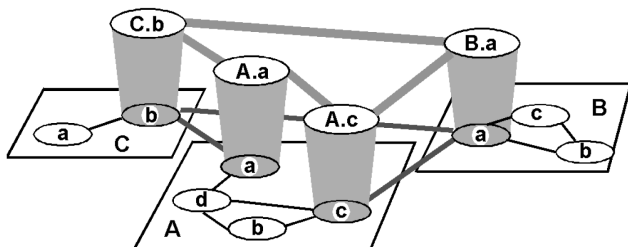
Autonomní systémy – *Multihomed* a *Transit AS*



- autonomní systém B je
 - *multihomed AS*, pokud je připojen k nejméně dvěma dalším AS, mezi kterými však neumožňuje přenášení provozu
 - *transit AS*, pokud je připojen k nejméně dvěma dalším AS, mezi kterými umožňuje přenášení provozu (skrze své LANs)

Autonomní systémy – směrování I.

- oddělené směrování z důvodů škálovatelnosti
 - intradoménové – *interior routing*
 - směrování uvnitř AS
 - plně pod kontrolou správce AS
 - tzv. *Interior Gateway Protocols (IGP)* (např. RIP, OSPF)
 - interdoménové/mezidoménové – *exterior routing*
 - směrování mezi AS
 - tzv. *Exterior Gateway Protocols (EGP)* (např. EGP, BGP-4)
 - nutná spolupráce interior a exterior směrovacích protokolů

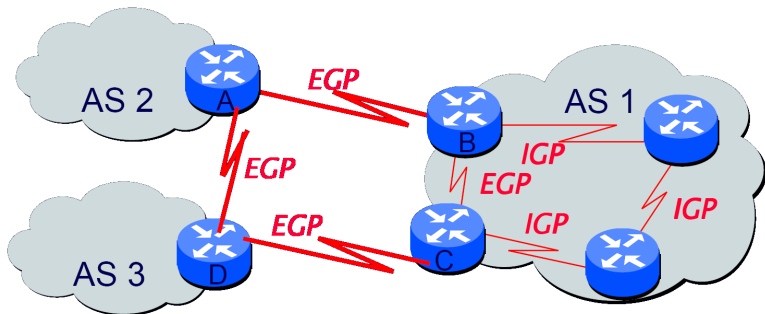


Autonomní systémy – směrování II.

- interní směrovače (směrovače uvnitř AS)
 - znají cestu do všech podsítí uvnitř AS
 - mohou využít implicitní (default) cesty
 - skrze hraniční směrovače
- *hraniční směrovače (Border Routers)*
 - sumarizují a zveřejňují interní cesty
 - aplikují směrovací „pravidla“ (*policy*)
- jádro sítě nepoužívá implicitní cesty
 - ⇒ směrovače musí znát cesty ke **všem** sítím

- Proč rozlišovat mezi směrováním uvnitř AS a mezi AS?
 - uvnitř AS hraje hlavní roli výkon
 - mezi AS hrají hlavní roli politiky (typicky jde o peníze) a škálovatelnost (velikosti tabulek)

Autonomní systémy – mezidoménové směrování



- AS1 propojen s AS2 a AS3
- směrovací „pravidla“ (*policies*) AS1 mohou zakazovat, aby se v případě výpadku linky mezi AS2 a AS3 směrovalo mezi AS2 a AS3 skrze AS1

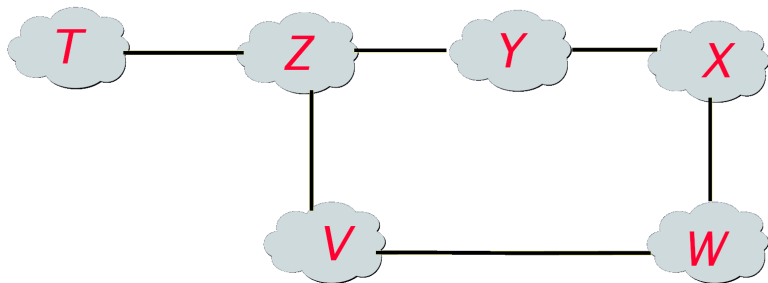
Autonomní systémy – mezidoménové směrování

Směrovací pravidla

- volba cesty není nezávislá na lokálních požadavcích
 - obchodní rozhodnutí
- lokální rozhodnutí definují
 - výběr cesty
 - zveřejnění interních podsítí
- *důsledky*:
 - kombinace nejlepších lokálních pravidel nemusí představovat globální optimum
 - asymetrie cest

Autonomní systémy – mezidoménové směrování

Směrovací pravidla – ilustrace I.

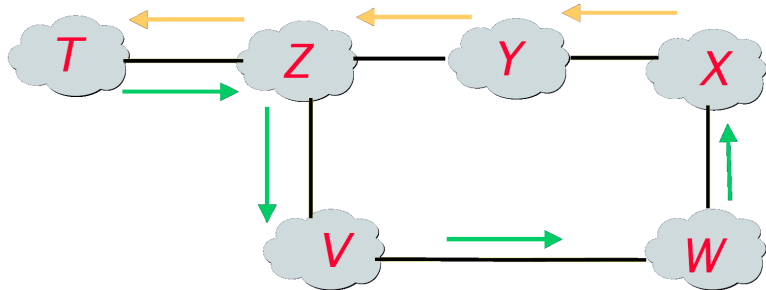


Předpokládejme, že AS Z chce oznámit AS T cestu $Z \rightarrow Y \rightarrow X$

- tato cesta může být AS T akceptována jen tehdy, pokud AS Y umožňuje přenos jeho provozu

Autonomní systémy – mezidoménové směrování

Směrovací pravidla – ilustrace II.

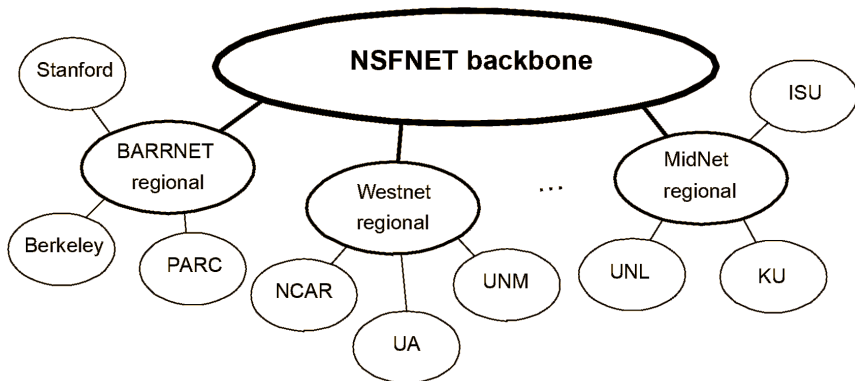


- jestliže AS Y neumožní přenos provozu AS T, ale umožní přenos provozu AS X, budou data mezi AS T a AS X přenášena asymetricky

Mezidoménové směrování – protokol EGP I.

- *Exterior Routing Protocol*
- první protokol mezidoménového směrování (navržen v roce 1983)
- využívá DV přístup
 - distance vektory kombinují cesty a pravidla
- cílem dosažitelnost, nikoliv efektivita
- navržen pro stromovou strukturu Internetu
 - přílišné zjednodušení
 - nepodporuje redundanci, neumí se vypořádat s cykly
 - ⇒ již se nepoužívá

Mezidoménové směrování – protokol EGP II.

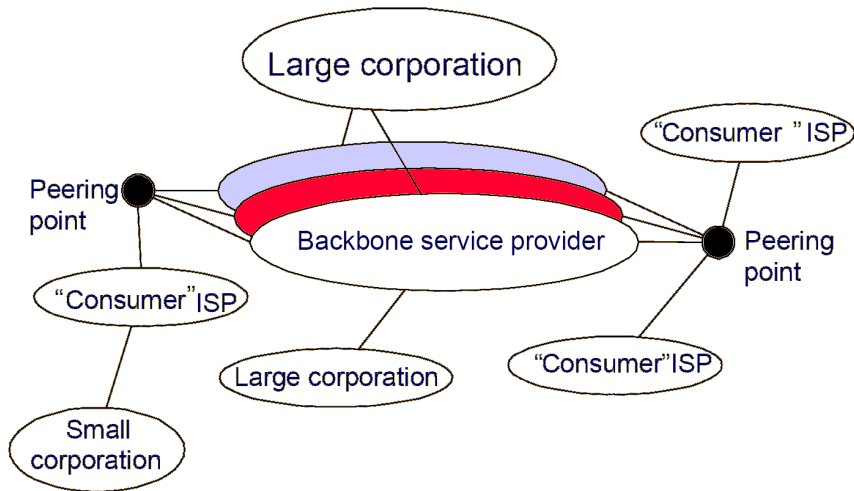


Obrázek: Představa Internetu podle EGP.

Mezidoménové směrování – BGP I.

- *Border Gateway Protocol*
 - aktuálně verze 4 (BGP-4)
- navržen v důsledku růstu Internetu a požadavků na podporu komplexnějších topologií
 - podporuje redundantní topologie, vypořádá se s cykly
- využívá *Path Vector* směrování
 - nevyměňují se ceny cest, ale popis celých cest zahrnující všechny skoky
- umožňuje definici pravidel směrování
- pracuje nad spolehlivým protokolem (TCP)
- používá CIDR pro agregaci cest

Mezidoménové směrování – BGP II.



Obrázek: Představa Internetu podle BGP.

Mezidoménové směrování – BGP III.

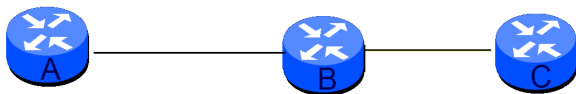
Path Vector I.

- *Path Vector (PV)*
 - obdoba DV
 - posílají se celé cesty (ne jen koncové uzly)
 - snadná detekce cyklů
 - umožňuje definici pravidel (přátelské vs. nepřátelské AS)
 - kratší cesty preferovány (pokud „policy“ nerozhodne jinak)
 - nepoužívá žádnou metriku, řeší se pouze dostupnost
 - důsledek: není nutné, aby všechny AS využívaly stejnou metriku

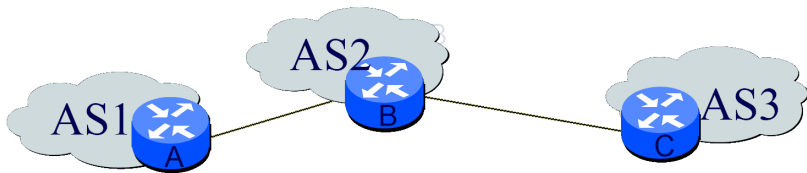
Mezidoménové směrování – BGP III.

Path Vector II.

Distance Vector přístup: C je vzdáleno 2 hopy od A



Path Vector přístup: cesta z AS1 do AS3 vede skrze AS2



- 1 Směrování obecně
- 2 Směrování
 - Základní přístupy
- 3 Směrovací algoritmy
- 4 Distribuované směrování
 - Distance Vector
 - Link State
 - Link State vs. Distance Vector
- 5 Hierarchie směrování
 - Původní představy
 - Autonomní systémy
 - Autonomní systémy – směrování
- 6 Multicastové směrování – IP Multicast**
 - **Motivace**
 - **IP Multicast**
 - **Protokoly**

Skupinová komunikace

Výzva: *způsob zasílání stejných zpráv skupině koncových stanic*

Příklady reálného světa:

- Televize či rozhlas, informace od zdroje k dynamické skupině
- Přednášející x auditorium, informace od zdroje ke skupině příjemců
 - s ojedinelou zpětnou vazbou
- Pracovní porada, informace od více zdrojů k více příjemcům
- Moderovaná diskuze, existence rolí ve skupině
- ...

Skupina se liší počtem členů, dynamikou, vzdáleností, aktivitou členů ...

Skupinová komunikace v síti

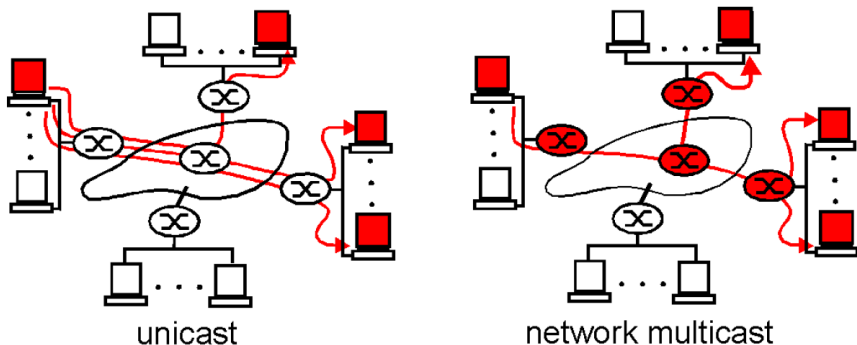
Obdoba předchozího.

- Data jsou od zdroje přenášena ke skupině příjemců
- Původní dvoubodová komunikace → vícebodová komunikace
- Nutno zajistit replikaci dat a jejich doručení

Pokud by replikace byla součástí aplikace, musela by každá aplikace obsahovat replikační modul. Proto je lepší řešit replikaci a směrování dat ve skupině odděleně od aplikace.

- IP Multicast
- Virtuální síť

Skupinová komunikace v síti – Unicast vs. Multicast



Obrázek: Doručení dat skupině příjemců – Unicast vs. Multicast

Příklady skupinové komunikace v síti

- Streamované video vysílané ve smyčce
 - Nedefinovaně mnoho příjemců
 - Šířka pásma: jednotky Kb/s až Mb/s
 - Kvalita přenosu
- Data produkovaná přístrojem (např. LHC)
 - Definovaně mnoho příjemců
 - Velké objemy dat po dlouhou dobu
 - Spolehlivé doručení
- Videokonference (např. nekomprimovaná HD videokonference)
 - Omezeny počet příjemců
 - Komunikace každý s každým
 - Šířka pásma: stovky Kb/s až Gb/s
 - Nízká latence (reálný čas)

IP Multicast – úvod

Klasické řešení skupinové komunikace v síti.

- Každým spojem nejvýše jedna kopie dat
- Vlastnost sítě (hop by hop, nikoliv end-to-end služba)
- Doručení nezaručené (best effort, UDP, skupinová adresa)
- Rozsah šíření omezen TTL (Time To Live) paketů

Jak identifikovat skupinu?

- \Rightarrow multicastová IP adresa
 - IPv4: třída D (224.0.0.0 – 239.255.255.255)
 - IPv6: prefix ff00::/8

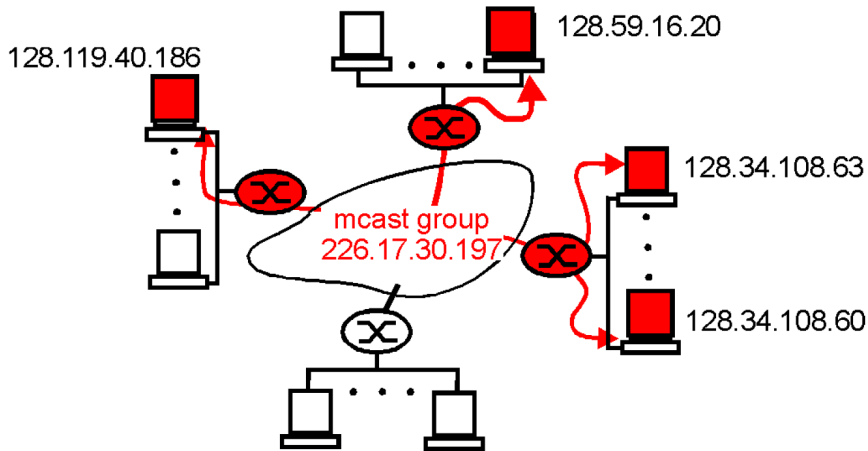
Dva základní přístupy k multicastovému směrování:

- *Source Based Tree*
- *Shared Tree (Core Based Tree)*

IP Multicast – komunikující strany

- *Vysílající:*
 - každý může vysílat (pokud zná multicastovou/skupinovou adresu)
 - stačí zasílat pakety na skupinovou adresu
 - vysílajících je proměnný počet
 - může, ale nemusí být členem skupiny
- *Přijímající:*
 - žádný, jeden, více
 - kdokoliv se může přidat či může opustit skupinu
 - může patřit do více skupin současně

IP Multicast – identifikace skupiny příjemců



Obrázek: Identifikace příjemců – datagram zaslaný do multicastové skupiny je doručen všem členům skupiny.

Source Based Tree vs. Core Based Tree

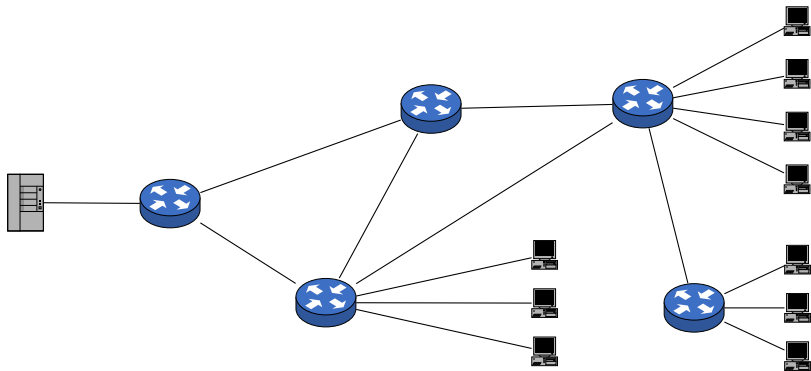
Source Based Tree

- Aktivita shora od zakládajícího
- Periodický broadcast
- Ořezávání větví bez členů
- Omezení šířky – TTL
- Pro úzce lokalizované skupiny
- Nevýhoda: režie, záplava broadcasty
- Protokoly: DVMRP (RIP), MOSPF (OSPF), PIM-DM

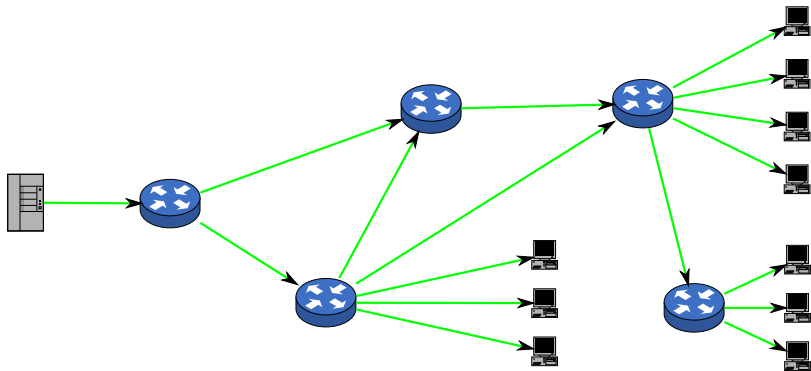
Core Based Tree

- Ustaveno jádro – body setkání (MP)
- Zájemce o skupinu kontaktuje MP
- Aktivita zdola od příjemce
- Redukce broadcastu → lépe škáluje
- Nevýhoda: závislost na dostupnosti jádra
- Protokoly: CBT, PIM-SM (protokolově nezávislé)

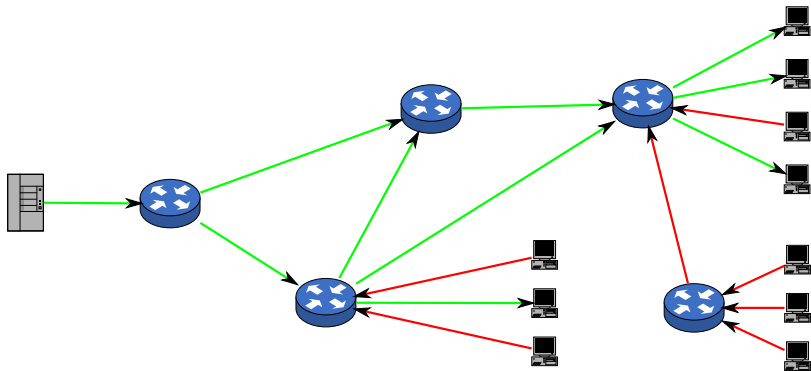
Source Based Tree



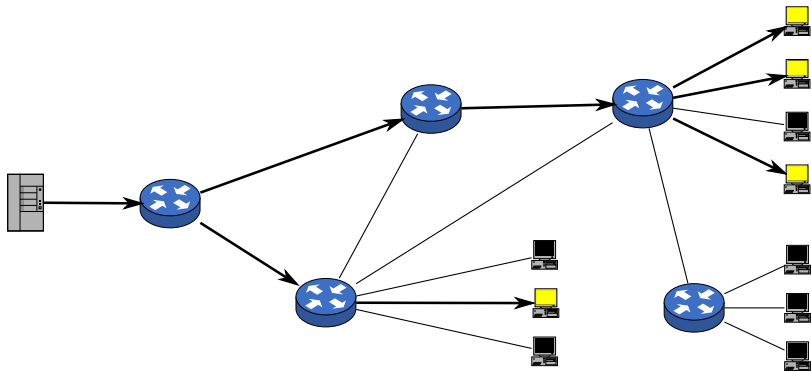
Source Based Tree



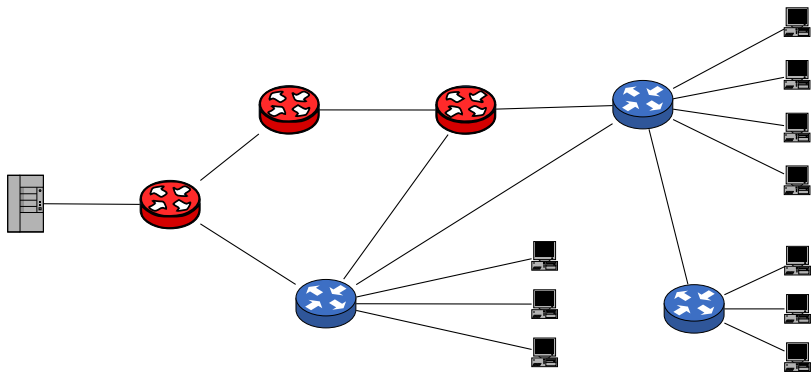
Source Based Tree



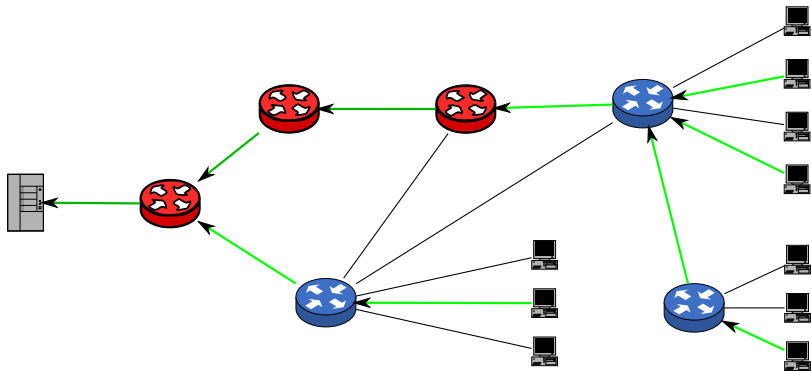
Source Based Tree



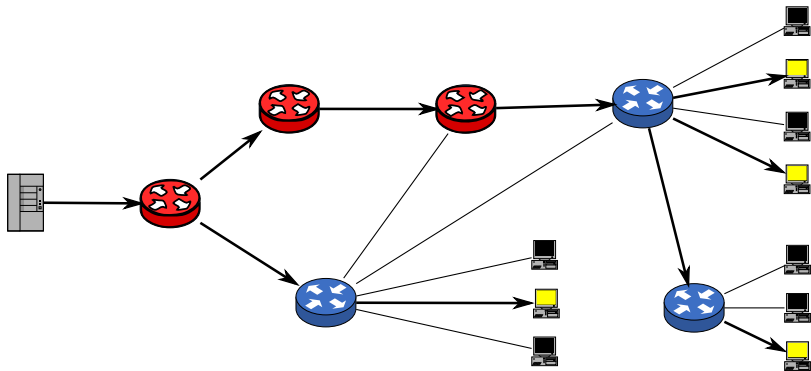
Core Based Tree



Core Based Tree



Core Based Tree

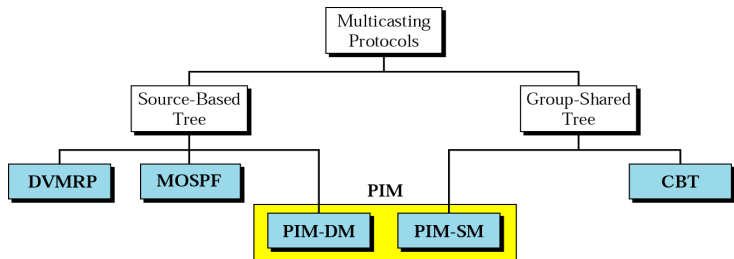


IP Multicast – vlastnosti

- Pozitivní:
 - „Nekonečná“ škálovatelnost
 - Nezatěžuje síť násobnými kopiemi
- Negativní:
 - Problematické účtování
 - Problém se zajištěným doručením
 - Snadný terč útoku (DoS, DDoS)
 - Absence kontroly členství (nelze zjistit přijímající)

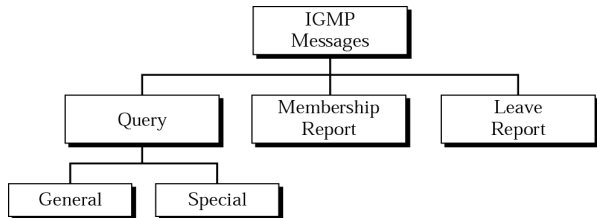
IP Multicast – protokoly

- Správa skupiny:
 - pouze v rámci LAN
 - *Internet Group Management Protocol (IGMP)*
- Směrování:
 - mezi multicastovými směrovači
 - Source Based Tree – DVMRP (RIP), MOSPF (OSPF), PIM-DM
 - Core Based Tree – CBT, PIM-SM

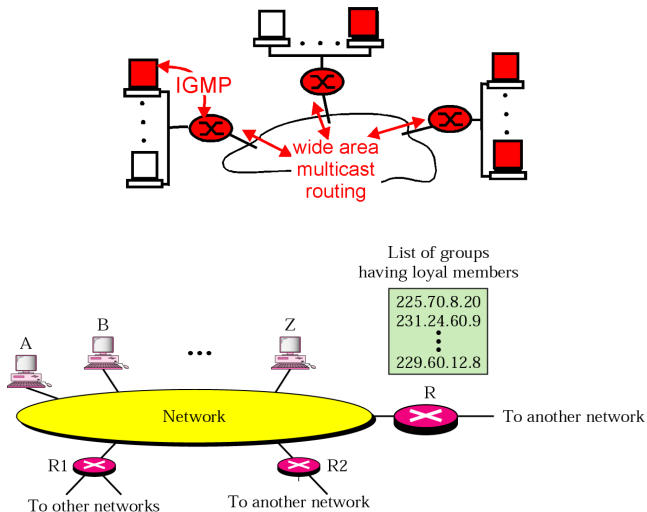


IP Multicast – správa skupiny – IGMP

- IGMP (RFC 1112), IGMPv2 (RFC 2236)
- správa členství ve skupině
 - spravuje informace o členech skupiny (pouze v rámci LAN)
 - pouze lokální působnost
 - síť a k ní přidružený multicastový směrovač
- typy zpráv:
 - přihlášení ke skupině (*Membership Report*)
 - odhlášení ze skupiny (*Leave Report*)
 - monitoring skupiny (*Query*)
 - např. dotazy směrovače na zájem uzlů setrvat ve skupině (řeší odstranění náhle vypadlých uzlů)



IP Multicast – správa skupiny – IGMP II.



Obrázek: Ilustrace lokální působnosti IGMP protokolu.

IP Multicast – Source Based Tree – protokoly

Distance Vector Multicast Routing Protocol (DVMRP)

- rozšíření unicastového DV směrování, využívá informací získaných RIP protokolem
- 3 přístupy pro budování stromu:
 - *Reverse Path Forwarding (RPF)*
 - *Reverse Path Broadcasting (RPB)*
 - *Reverse Path Multicasting (RPM)*

Multicast Open Shortest Path First (MOSPF)

- rozšíření unicastového OSPF protokolu
- využívá vytvořené znalosti topologie OSPF protokolem
 - všechny uzly počítají strom cest z kořene, kterým je zdroj multicastového vysílání

Protocol Independent Multicast – Dense Mode (PIM-DM)

- využit v prostředí, kdy je pravděpodobné, že většina směrovačů bude participovat na multicastování
- podobný DVMRP protokolu
 - využívá RPF přístup
 - rozdíl: ke své činnosti nevyžaduje unicastový protokol (tj. RIP)

IP Multicast – Core Based Tree – protokoly

Core-Based Tree (CBT)

- zdroj jako kořen budovaného stromu
- AS rozdělen na regiony, pro každý region zvolen „bod setkání“ (tzv. *Rendezvous Router*)
 - ⇒ vytvoření jádra
- uzly (v případě zájmu) kontaktují „body setkání“
 - budování stromu od listů

Protocol Independent Multicast – Sparse Mode (PIM-SM)

- využít v prostředí, kdy je malá pravděpodobnost, že většina směrovačů bude participovat na multicastování
- podobný CBT protokolu
 - také využívá *Rendezvous Points (RPs)*
 - oproti CBT si buduje záložní RPs pro účely jejich výpadků
 - v případě potřeby (= mnoho příjemců vzdálených od RP) je schopen přepnout do strategie *Source-based Tree*

