

3. Síťová vrstva – Směrování

PB156: Počítačové sítě

Eva Hladká

Fakulta informatiky Masarykovy univerzity

jaro 2010

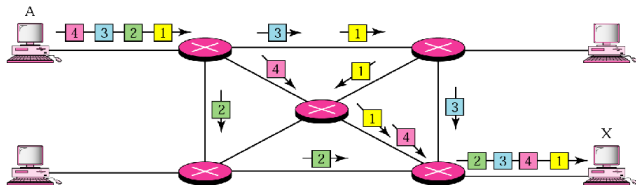
Struktura přednášky

- 1 Směrování obecně
- 2 Směrování
 - Základní přístupy
- 3 Směrovací algoritmy
- 4 Distribuované směrování
 - Distance Vector
 - Link State

- 1 Směrování obecně
- 2 Směrování
 - Základní přístupy
- 3 Směrovací algoritmy
- 4 Distribuované směrování
 - Distance Vector
 - Link State

Směrování obecně

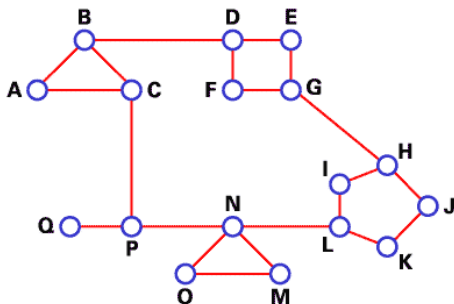
- Internet na L3 – datagramový přístup k přepínání paketů
 - data vyšších vrstev umísťována do datagramů
 - datagramy (fragmenty) putují sítí nezávisle



- **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

Směrování

- úkolem směrování je:
 - vyhledávat optimální směrovací trasy
 - kritériem 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:

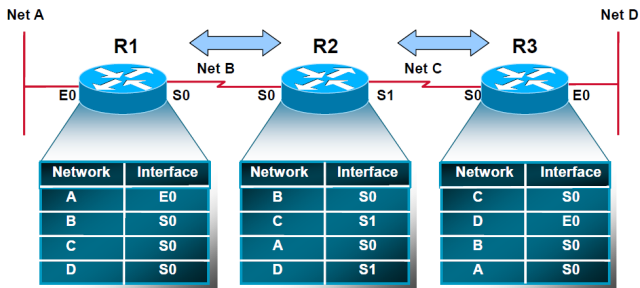
- správnost
- jednoduchost
- efektivita a škálovatelnost
 - minimalizace množství řídicích informací ($\approx 5\%$ provozu!)
 - minimalizace velikosti směrovacích tabulek
- robustnost a stabilita
 - nezbytný je distribuovaný algoritmus
- spravedlivost (fairness)
- optimálnost
 - „Co je to nejlepší cesta?“

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

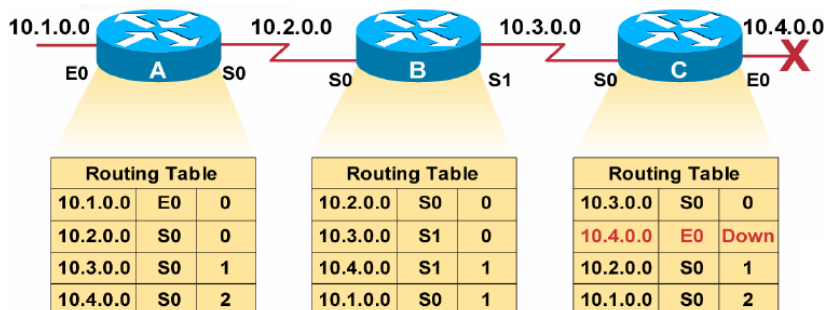
Distance Vector II.

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.

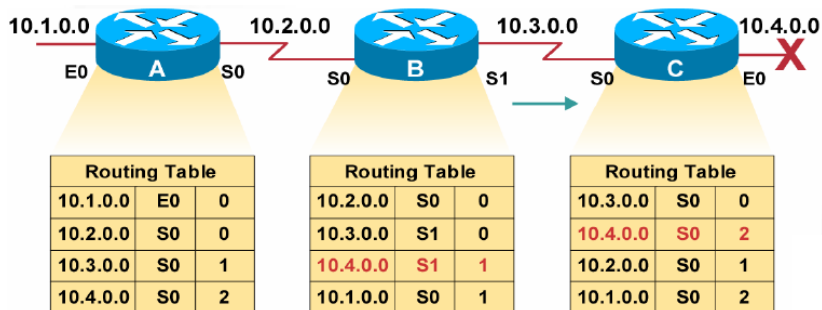
Ilustrace problému pomalé konvergence



- pomalá konvergence zapříčiní vznik nesprávných údajů ve směrovacích tabulkách

Distance Vector III.

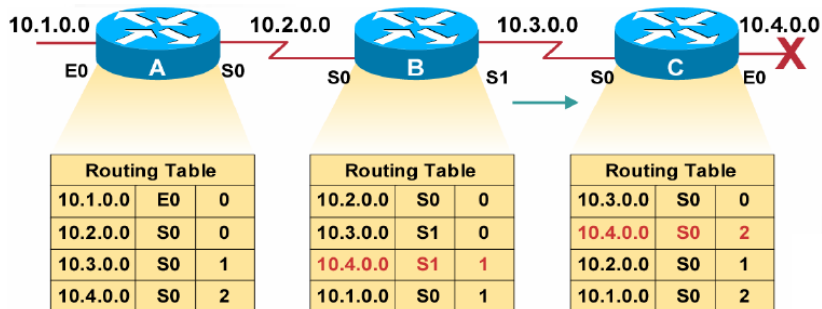
Ilustrace problému pomalé konvergence



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

Distance Vector III.

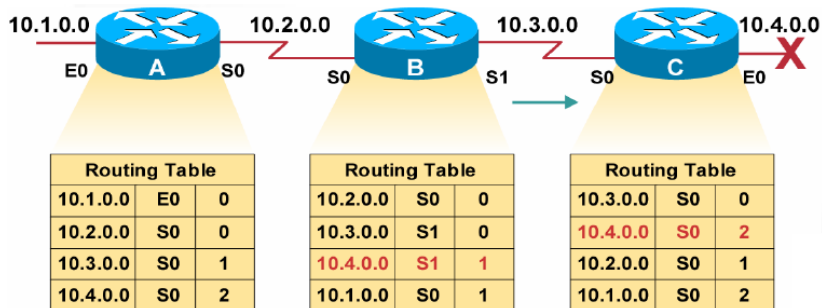
Ilustrace problému pomalé konvergence



- směrovač A opraví svojí směrovací tabulku – chybně

Distance Vector III.

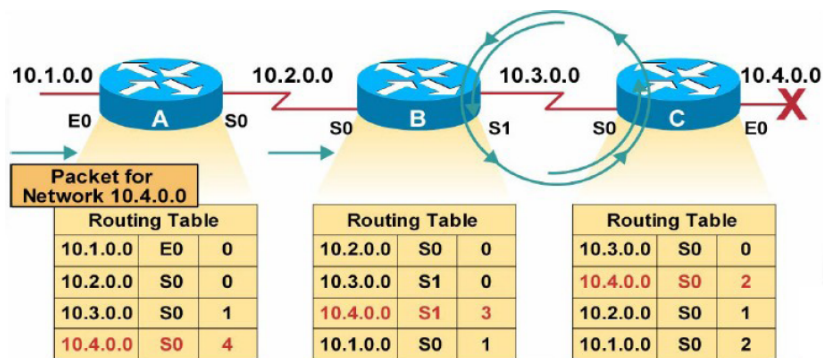
Ilustrace problému pomalé konvergence



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

Distance Vector III.

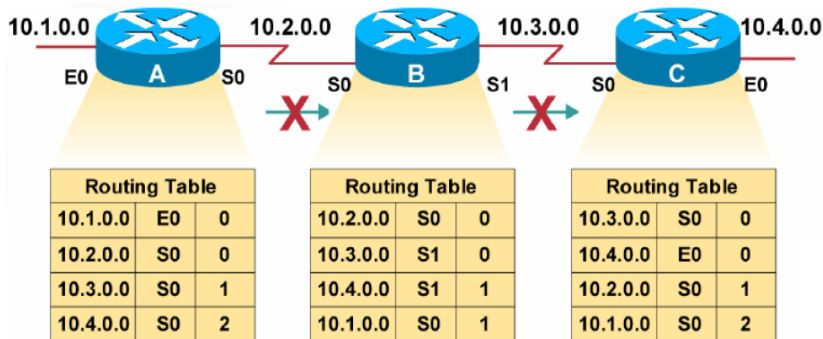
Ilustrace problému pomalé konvergence



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

Distance Vector III.

Ilustrace problému pomalé konvergence

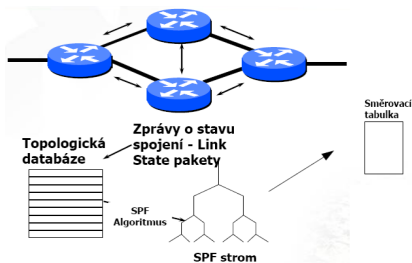


- Řešení: *dělení horizontu*
 - směrovač nesdílí cestu zpět uzlu, od kterého se o ní dozvěděl
 - problém zůstává ve složitějších topologiích (navržena řada rozšíření)

Distance Vector IV. – protokol RIP

- 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

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. – 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