

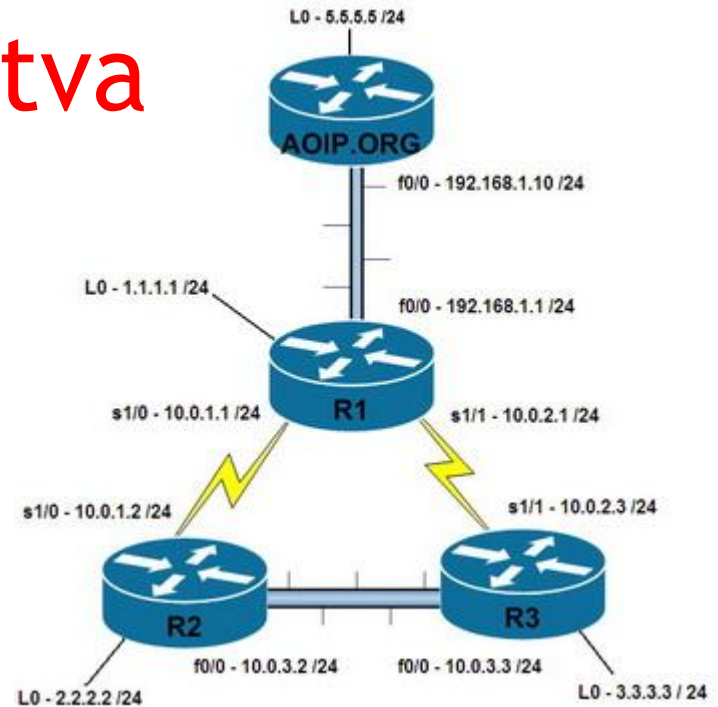
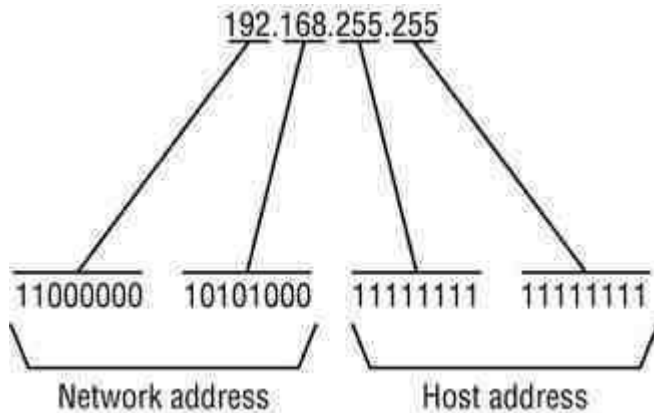
8. prednáška

4-bit	8-bit	16-bit	32-bit	
Ver.	Header Length	Type of Service	Total Length	
Identification			Flags	Offset
Time To Live	Protocol	Checksum		
Source Address				
Destination Address				
Options and Padding				

158.197.31.4/24

fe80::231:5cff:fe64:db91/64

Siet'ová vrstva 3.časť



Prehľad prednášky

❑ Smerovacie algoritmy

- ❖ LSA

- ❖ DVA

- ❑ Protokoly smerovania vnútri autonómnych systémov

- ❖ RIP

- ❖ OSPF

- ❑ Hierarchické smerovanie

- ❑ Protokol smerovania medzi autonómnyimi systémami : BGP

- ❑ Princípy smerovania broadcastu

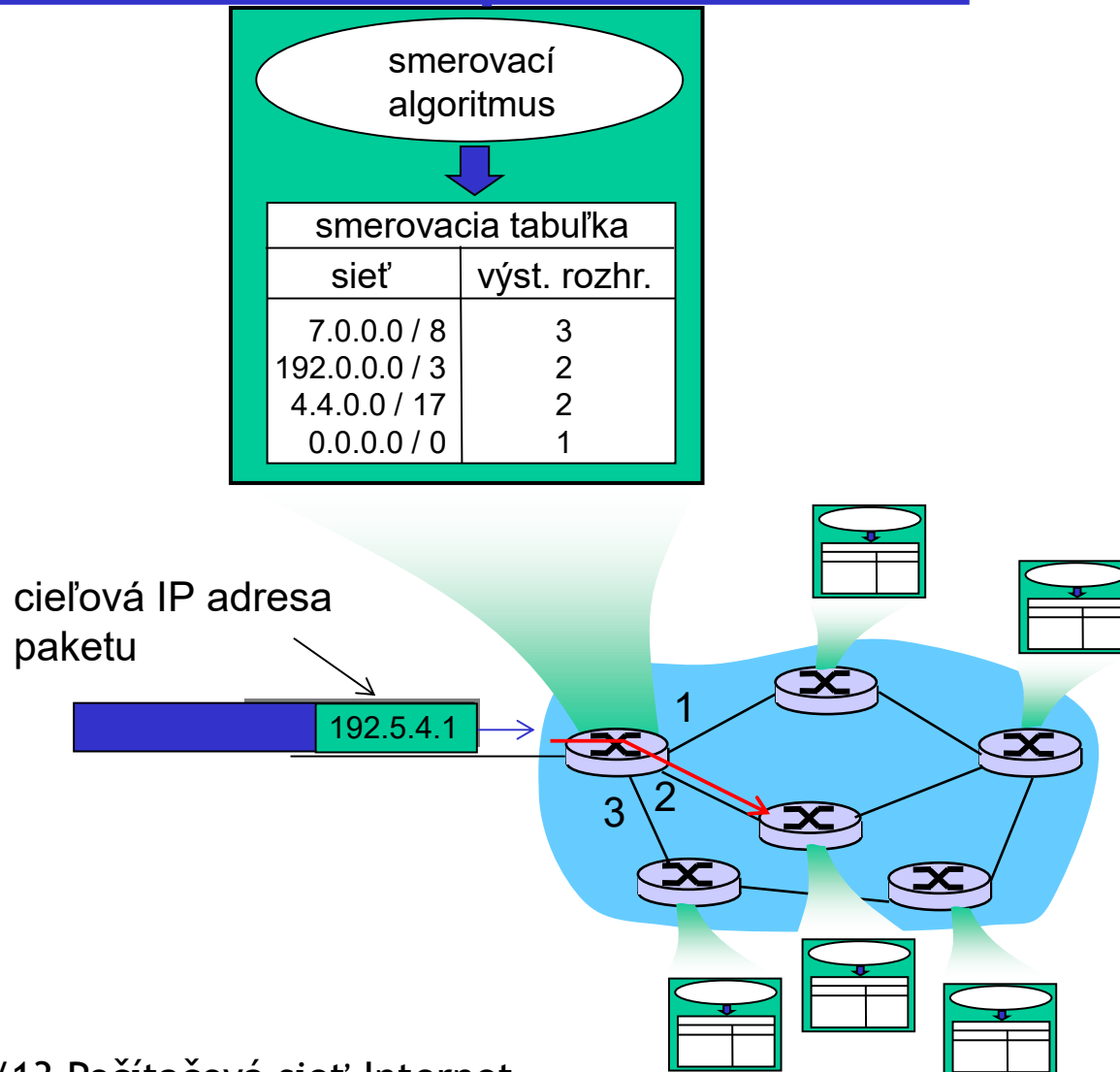
- ❑ Princípy smerovania multicastu

- ❑ Protokoly multicastového smerovania

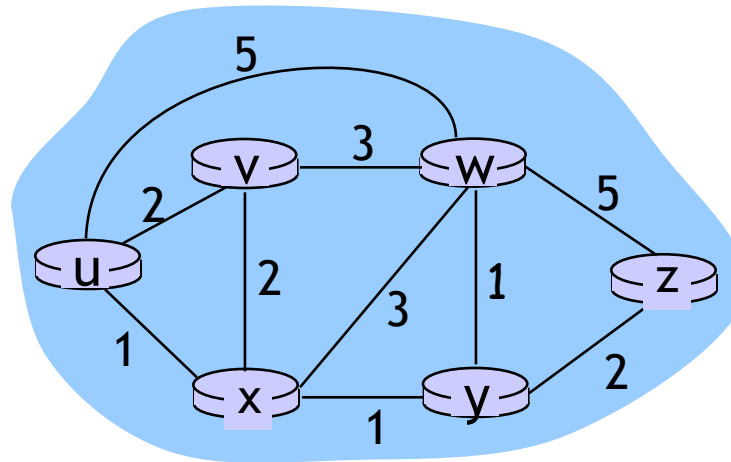
- ❖ DVMRP

- ❖ PIM

Cieľ smerovacieho algoritmu a smerovacích protokolov



Sieť routrov ako graf

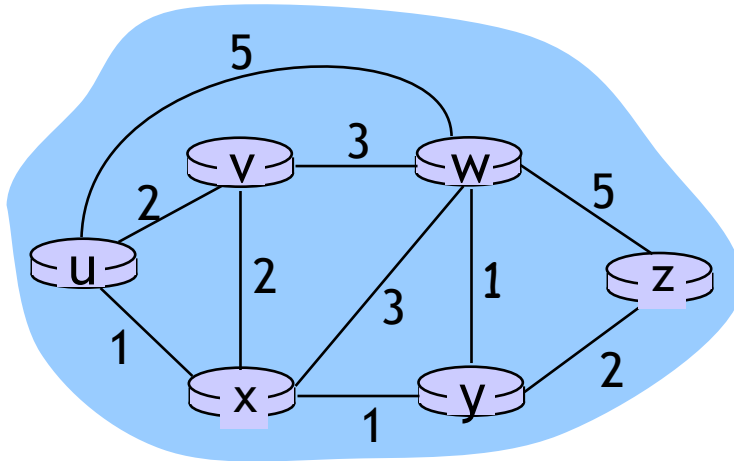


Graf: $G = (V,E)$

$V =$ množina vrcholov/uzlov/routrov = $\{ u, v, w, x, y, z \}$

$E =$ množina hrán/spojení = $\{ (u,v), (u,x), (u,w), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

Ceny (váhy) hrán grafu



$c(v_1, v_2)$ = cena hrany (v_1, v_2)

- napr. $c(w, z) = 5$

- cena každej hrany môže byť 1, ale môže byť nastavovaná napr. podľa šírky pásma alebo zahľtenia spoja

Cena cesty $(v_1, v_2, v_3, \dots, v_p) = c(v_1, v_2) + c(v_2, v_3) + \dots + c(v_{p-1}, v_p)$

Otázka: Aká je najkratšia cesta medzi routrami u a w ?

Smerovací algoritmus: nastaví smerovaciu tabuľku tak, aby cesty paketov k cieľom boli čo najlacnejšie

Klasifikácia smerovacích algoritmov

S globálnym prehľadom:

- všetky routre majú kompletnú informáciu o topológii siete a cenách hrán

- **“link state” algoritmus**

Decentralizovaný alg.:

- router pozná iba fyzicky naňho napojených susedov a ceny týchto hrán

- iteratívny proces výpočtu

- výmena (medzi-)výsledkov medzi susedmi

- **“distance vector” algoritmus**

Statický:

- riadky smerovacej tabuľky sa menia zriedka

Dynamický:

- riadky smerovacej tabuľky sa aktualizujú často

- ❖ periodické obnovovanie tabuľky

- ❖ reaguje na zmeny cien hrán

Prehľad prednášky

☐ Smerovacie algoritmy

- ❖ LSA

- ❖ DVA

☐ Protokoly smerovania vnútri autonómnych systémov

- ❖ RIP

- ❖ OSPF

☐ Hierarchické smerovanie

☐ Protokol smerovania medzi autonómnyimi systémami : BGP

☐ Princípy smerovania broadcastu

☐ Princípy smerovania multicastu

☐ Protokoly multicastového smerovania

- ❖ DVMRP

- ❖ PIM

“Link-State” algoritmus (LSA)

Dijkstrov algoritmus

- všetky uzly vedia celú topológiu siete aj ceny hrán
- ❖ získané broadcastovými alebo multicastovými správami
- ❖ všetky uzly vedia to isté
- počíta najkratšie (najlacnejšie) cesty z daného uzla ku všetkým ostatným
- ❖ získame smerovaciu tabuľku daného uzla
- v každej iterácii cyklu algoritmu zistíme najkratšiu cestu k ďalšiemu uzlu

Označenia:

- $c(x,y)$: cena cesty z uzla x do uzla y ; $= \infty$, ak nie sú spojené žiadnou cestou, alebo cestu z x do y ešte nepoznáme
- $D(v)$: aktuálne najmenšia zistená cena cesty zo štartovacieho uzla do uzla v
- $p(v)$: predchodca uzla v na ceste zo štartovacieho uzla do uzla v
- N : množina uzlov, ku ktorým už poznáme najlacnejšiu cestu

Dijkstrov algoritmus

Inicializácia:

$N = \{u\}$ // u je štartovací uzol

pre všetky uzly v

ak v je susedom u tak

$$D(v) = c(u,v)$$

inak $D(v) = \infty$

Výpočet:

opakuj

vezmi $w \notin N$ také, že $\forall x \notin N: D(w) \leq D(x)$

pridaj w do N

zmeň $D(v)$ pre $v \notin N$, ktoré sú susedom w :

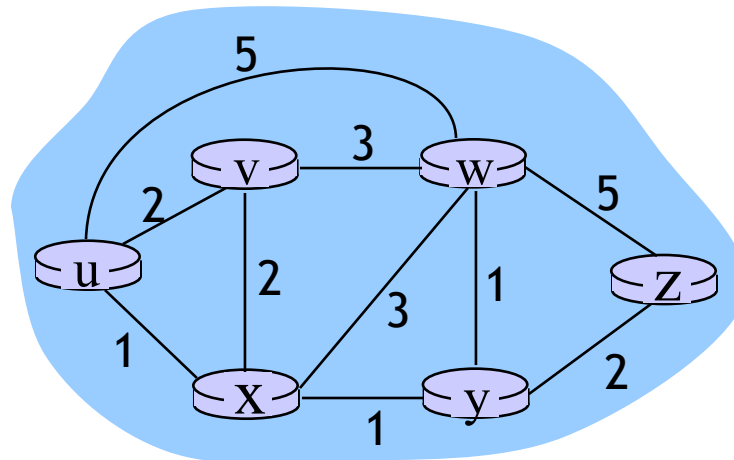
$$D(v) = \min(D(v), D(w) + c(w,v))$$

ak sa $D(v)$ zmenilo nastav $p(v)=w$

pokiaľ N neobsahuje všetky uzly

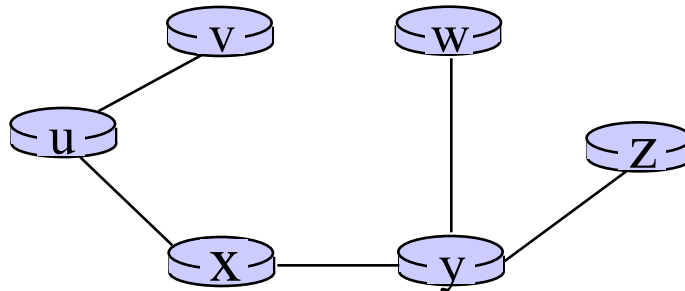
Dijkstrov algoritmus: príklad

iterácia	N	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Dijkstrov algoritmus: príklad

Dostaneme strom najkratších ciest z uzla u:



Nastavíme smerovaciu tabuľku routra u:

cieľ	spoj (rozhranie)
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Dijkstrov algoritmus: diskusia

Zložitosť algoritmu: máme $|V|$ uzlov (routrov) - 1 iteráciu pre každý

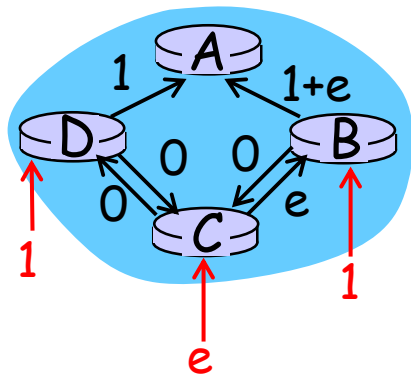
□ v každej iterácii: potrebujeme nájsť minimálny uzol $w \notin N$ - treba čas $O(|V||V|)$, alebo $O(\log(|V|))$, ak použijeme haldu

□ Dokopy prerátame cesty pre $|E|$ hrán pre novopridané uzly

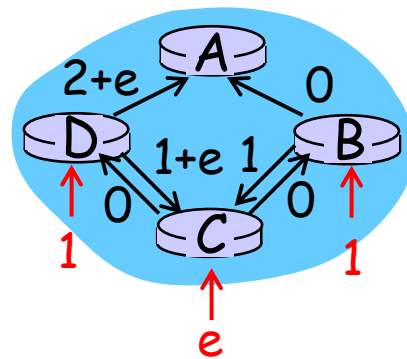
□ Celkovo, ak použijeme haldu: $O(|E| \log(|V|))$

Nedostatok tohto prístupu: možné oscilácie

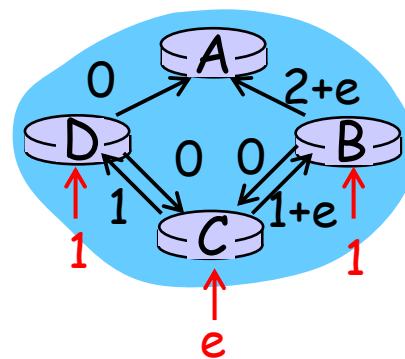
□ Napríklad, ak cena spoja = množstvo prenášaných dát za nejaký čas



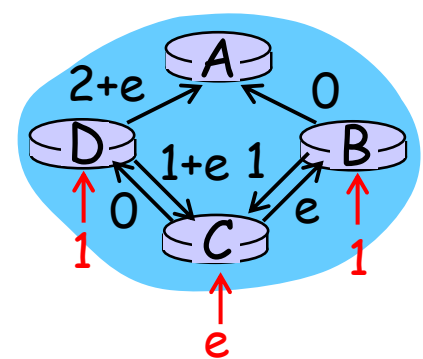
začiatok vysielania
k uzlu A



... prepočítame
vzdialenosti



... prepočítame



... prepočítame

Prehľad prednášky

❑ Smerovacie algoritmy

- ❖ LSA

- ❖ DVA

❑ Protokoly smerovania vnútri autonómnych systémov

- ❖ RIP

- ❖ OSPF

❑ Hierarchické smerovanie

❑ Protokol smerovania medzi autonómnyimi systémami : BGP

❑ Princípy smerovania broadcastu

❑ Princípy smerovania multicastu

❑ Protokoly multicastového smerovania

- ❖ DVMRP

- ❖ PIM

Distance Vector Algorithmus (DVA)

Bellman-Fordova rovnica (z dynamického programovania)

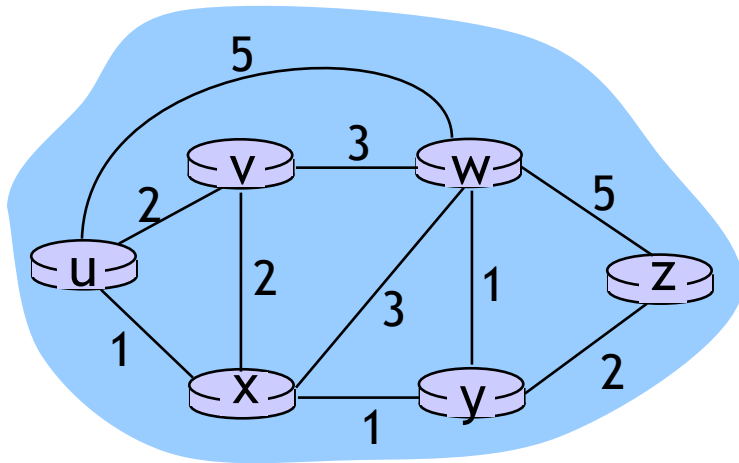
$n(x,y)$:= cena najlacnejšej cesty z uzla x do uzla y

Potom

$$n(x,y) = \min_v \{c(x,v) + n(v,y)\}$$

kde minimum je cez všetkých susedov v uzla x

Bellman-Ford: príklad



Dozvieme sa od susedov, že
cena cesty od nich do uzla z je:
 $n(v, z) = 5$, $n(x, z) = 3$, $n(w, z) = 3$

z B-F rovnice vieme:

$$\begin{aligned} n(u,z) &= \min \{ c(u, v) + n(v, z), \\ &\quad c(u, x) + n(x, z), \\ &\quad c(u, w) + n(w, z) \} \\ &= \min \{ 2 + 5, \\ &\quad 1 + 3, \\ &\quad 5 + 3 \} = 4 \end{aligned}$$

**Uzol, cez ktorý získame nové minimum,
bude určovať smer pre smerovaciu tabuľku**

Distance Vector Algorithmus (DVA)

□ $n(x, y)$ = očakávaná najmenšia cena cesty z x do y

□ Uzol x vie cenu spojenia k svojim susedom v :
 $c(x, v)$

□ Uzol x si uchováva vektor vzdialeností k
všetkým uzlom $N_x = [n(x, y): y \in N]$

□ Uzol x si uchováva aj vektory vzdialeností
svojich susedov

❖ Pre každého suseda v má:

$N_v = [n(v, y): y \in N]$

Distance Vector Algorithmus (DVA)

Základná idea:

- Každý uzol periodicky posiela svoj vektor vzdialeností svojim susedom
- Keď uzol x dostane nejaký vektor vzdialeností, obnoví si svoj vektor vzdialeností pomocou B-F rovnice:

$$n(x, y) \leftarrow \min_v \{c(x, v) + n(v, y)\} \text{ pre každý uzol } y \in N$$

- Vzdialenosti $n(x, _)$ vo vektore vzdialeností N_x uzla x postupne konvergujú k skutočne najmenším vzdialenostiam

Distance Vector Algoritmus (DVA)

Iteratívny, asynchrónny:

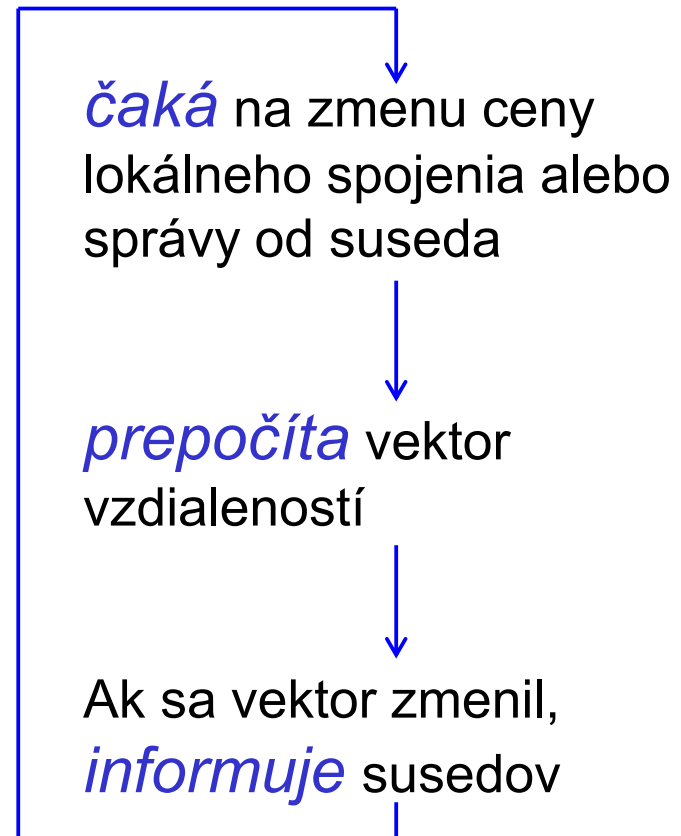
každá iterácia je spôsobená:

- zmenou ceny lokálneho spojenia so susedom
- zmenou vektora vzdialeností a jeho zaslaním susednému uzlu

Distribuoovaný:

- každý uzol informuje svojich susedov, *iba* keď sa zmení jeho vektor vzdialeností

Každý uzol:



$$n(x, y) = \min\{c(x, y) + n(y, y), c(x, z) + n(z, y)\}$$

$$= \min\{2 + 0, 7 + 1\} = 2$$

$$n(x, z) = \min\{c(x, y) + n(y, z), c(x, z) + n(z, z)\}$$

$$= \min\{2 + 1, 7 + 0\} = 3$$

uzol x

		cena k		
		x	y	z
od	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

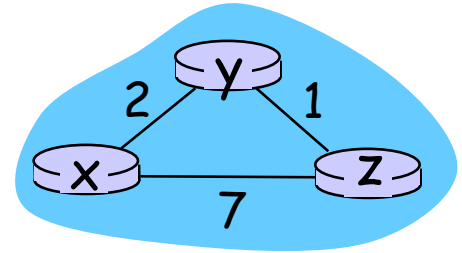
		cena k		
		x	y	z
od	x	0	2	3
	y	2	0	1
	z	7	1	0

uzol y

		cena k		
		x	y	z
od	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

uzol z

		cena k		
		x	y	z
od	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0



čas →

$$n(x, y) = \min\{c(x, y) + n(y, y), c(x, z) + n(z, y)\} = \min\{2 + 0, 7 + 1\} = 2$$

$$n(x, z) = \min\{c(x, y) + n(y, z), c(x, z) + n(z, z)\} = \min\{2 + 1, 7 + 0\} = 3$$

uzol x

		cena k		
		x	y	z
od	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

		cena k		
		x	y	z
od	x	0	2	3
	y	2	0	1
	z	7	1	0

		cena k		
		x	y	z
od	x	0	2	3
	y	2	0	1
	z	3	1	0

uzol y

		cena k		
		x	y	z
od	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

		cena k		
		x	y	z
od	x	0	2	7
	y	2	0	1
	z	7	1	0

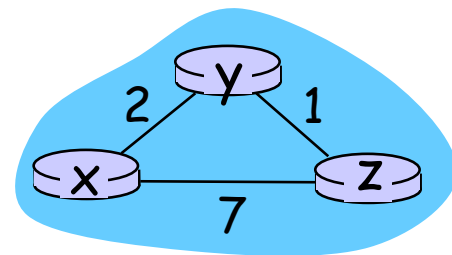
		cena k		
		x	y	z
od	x	0	2	3
	y	2	0	1
	z	3	1	0

uzol z

		cena k		
		x	y	z
od	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

		cena k		
		x	y	z
od	x	0	2	7
	y	2	0	1
	z	3	1	0

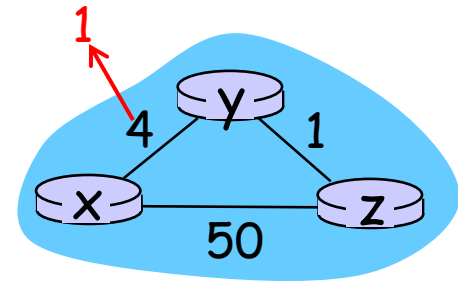
		cena k		
		x	y	z
od	x	0	2	3
	y	2	0	1
	z	3	1	0



čas

DVA: zmena cien spojení

- uzol zistí zmenu ceny lokálneho spojenia
- obnoví vektor vzdialeností a smerovaciu tabuľku
- ak sa smerovacia tabuľka zmenila, informuje susedov



V čase t_0 , y zistí zmenu ceny spojenia , zmení svoj vektor vzdialeností, a informuje svojich susedov

V čase t_1 , z dostane vektor od y a zmení svoju tabuľku.
z vypočíta novú najmenšiu cenu k x a pošle susedom svoj vektor vzdialeností

V čase t_2 , y dostane vektor od z a obnoví svoju tabuľku.
Vektor vzdialeností uzla y sa nezmenil a teda y už neposiela nič

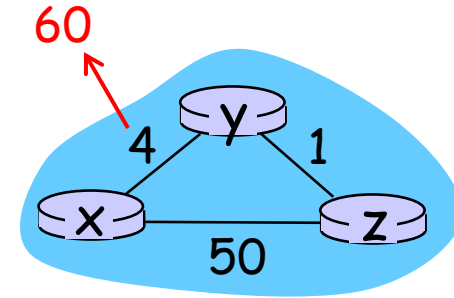
DVA: zmena cien spojení

Zmena ceny spojenia:

- ❑ zníženie ceny sa prejaví rýchlo
- ❑ zvýšenie ceny sa prejaví pomaly-
“count to infinity” problém!
- ❑ v našom príklade 47 iterácií pokiaľ sa dosiahne stabilný stav

Poisoned reverse:

- ❑ Ak z smeruje pakety pre x cez y :
 - ❖ uzol z povie uzlu y, že jeho vzdialenosť (uzla z) k uzlu x je nekonečno (takže y nebude smerovať pre x cez z)
- ❑ Vyrieši to úplne “count to infinity” problém ?



Na začiatku:

y má (4, 0, 1); z má (5, 1, 0)

y -> z ($\min\{60 + 0, 1 + 5\} = 6, 0, 1$)
y posielala správy pre x cez z

z -> y ($\min\{50 + 0, 1 + 6\} = 7, 1, 0$)
z posielala správy pre x cez y

y -> z ($\min\{60 + 0, 1 + 7\} = 8, 0, 1$)
y posielala správy pre x cez z

...

Porovnanie LSA (Dijkstra) a DVA

Počet správ

- LSA: pre $|V|$ uzlov a $|E|$ hrán posielame $O(|V||E|)$ správ
- DVA: správy iba medzi susedmi
- ❖ čas konvergovania môže byť rôzny

Čas výpočtu do ustáleného stavu

- LSA: $O(|E| \log(|V|))$
- ❖ môžeme skončiť pri oscilácii
- DVA: záleží od stavu siete
- ❖ môžeme mať dočasne smerovacie cykly
- ❖ “count to infinity” problém

Prispôsobiteľnosť: čo sa stane, ak sa nejaké spoje pokazia?

LSA:

- ❖ uzol, ktorý to spozoruje, môže rozoslať žiadosť o prepočítanie
- ❖ každý uzol si vypočíta novú tabuľku

DVA:

- ❖ uzol vie rozosielať iba nesprávnu cenu cesty (nie spoja)
- ❖ táto zmena sa môže postupne prejaviť u susedov a následne ďalej

Prehľad prednášky

- ❑ Smerovacie algoritmy
 - ❖ LSA
 - ❖ DVA
- ❑ Protokoly smerovania vnútri autonómnych systémov
 - ❖ RIP
 - ❖ OSPF
- ❑ Hierarchické smerovanie

- ❑ Protokol smerovania medzi autonómnyimi systémami : BGP
- ❑ Princípy smerovania broadcastu
- ❑ Princípy smerovania multicastu
- ❑ Protokoly multicastového smerovania
 - ❖ DVMRP
 - ❖ PIM

Smerovanie vnútri-AS

❑ alebo **Interior Gateway Protocols (IGP)**

❑ najznámejšie IGP protokoly:

❖ RIP: Routing Information Protocol

❖ OSPF: Open Shortest Path First

❖ IGRP: Interior Gateway Routing Protocol
(proprietárny Cisco protokol)

Prehľad prednášky

□ Smerovacie algoritmy

- ❖ LSA

- ❖ DVA

□ Protokoly smerovania vnútri autonómnych systémov

- ❖ RIP

- ❖ OSPF

□ Hierarchické smerovanie

□ Protokol smerovania medzi autonómnymi systémami : BGP

□ Princípy smerovania broadcastu

□ Princípy smerovania multicastu

□ Protokoly multicastového smerovania

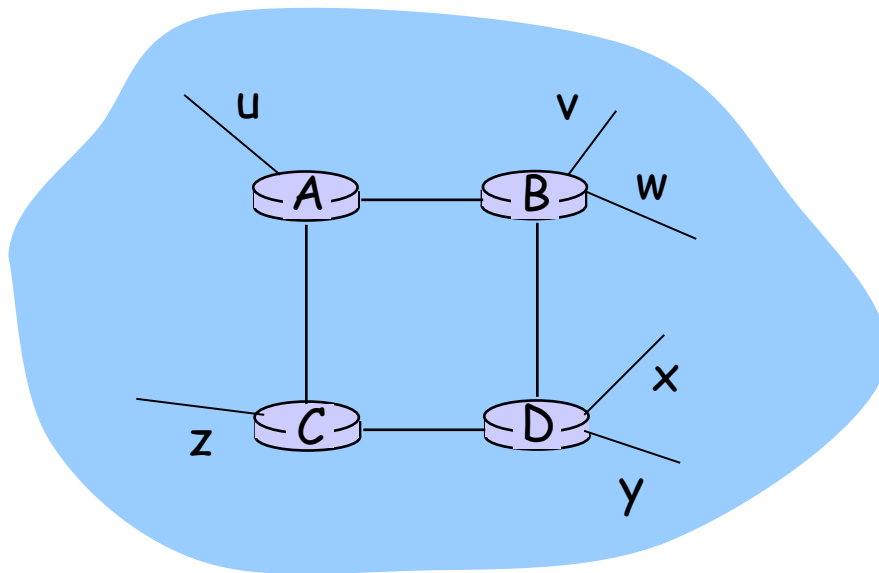
- ❖ DVMRP

- ❖ PIM

RIP = Routing Information Protocol

- ❑ distance vector algoritmus
- ❑ súčasť BSD-UNIXu už v roku 1982
- ❑ maximálna vzdialenosť 15 hopov (15 routrov na ceste)
- ❖ vzdialenejšie routre majú vzdialenosť 16, ktorá sa považuje za nekonečno

Vzdialenosť od routra A do sietí:

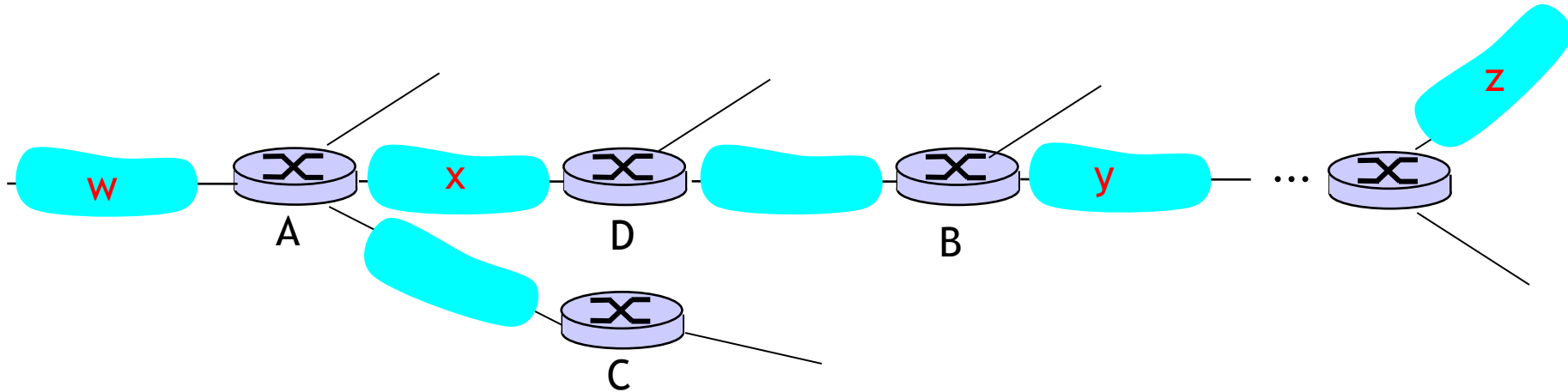


<u>sieť</u>	<u>hopov</u>
u	1
v	2
w	2
x	3
y	3
z	2

Výmena vektorov v RIP

- vektory vzdialeností sa vymieňajú medzi susedmi každých 30 sekúnd cez “Response Message”
- v každej výmene je zoznam maximálne 25 záznamov z vektora vzdialeností vo vnútri AS

RIP: Príklad



cieľová sieť	d'alší router	počet hopov k cieľovej sieti
w	A	2
y	B	2
z	B	7
x	-	1
...

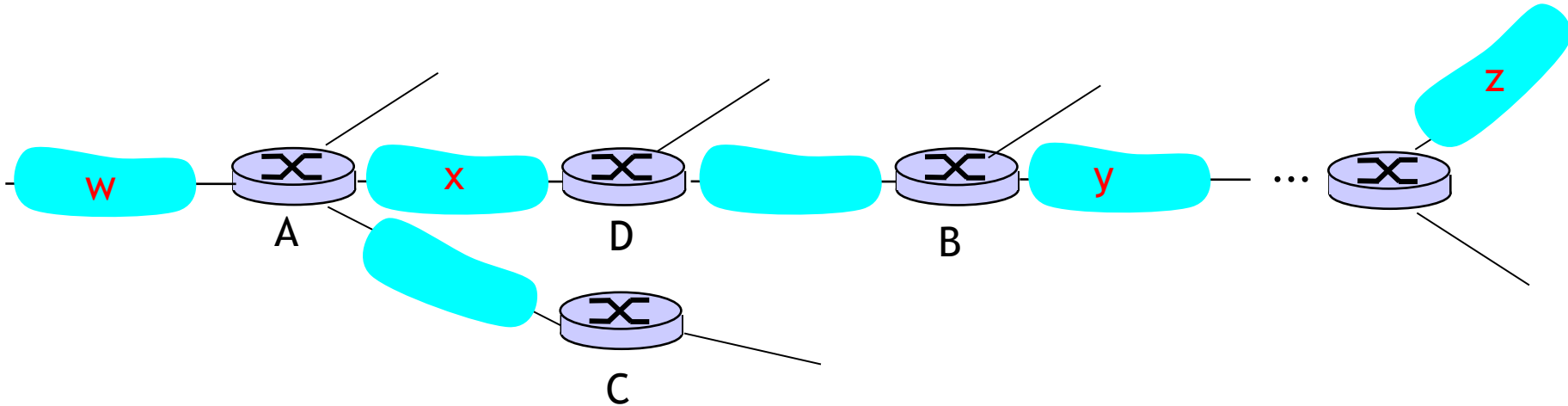
smerovacia tabuľka routra D

RIP: Príklad

cieľ ďalší hopov

w	-	1
x	-	1
z	C	4
...

dôjde vektor z A ku D



cieľová sieť ďalší router počet hopov k cieľovej sieti

w	A	2
y	B	2
z	B A	7 5
x	-	1
...

smerovacia tabuľka routra D

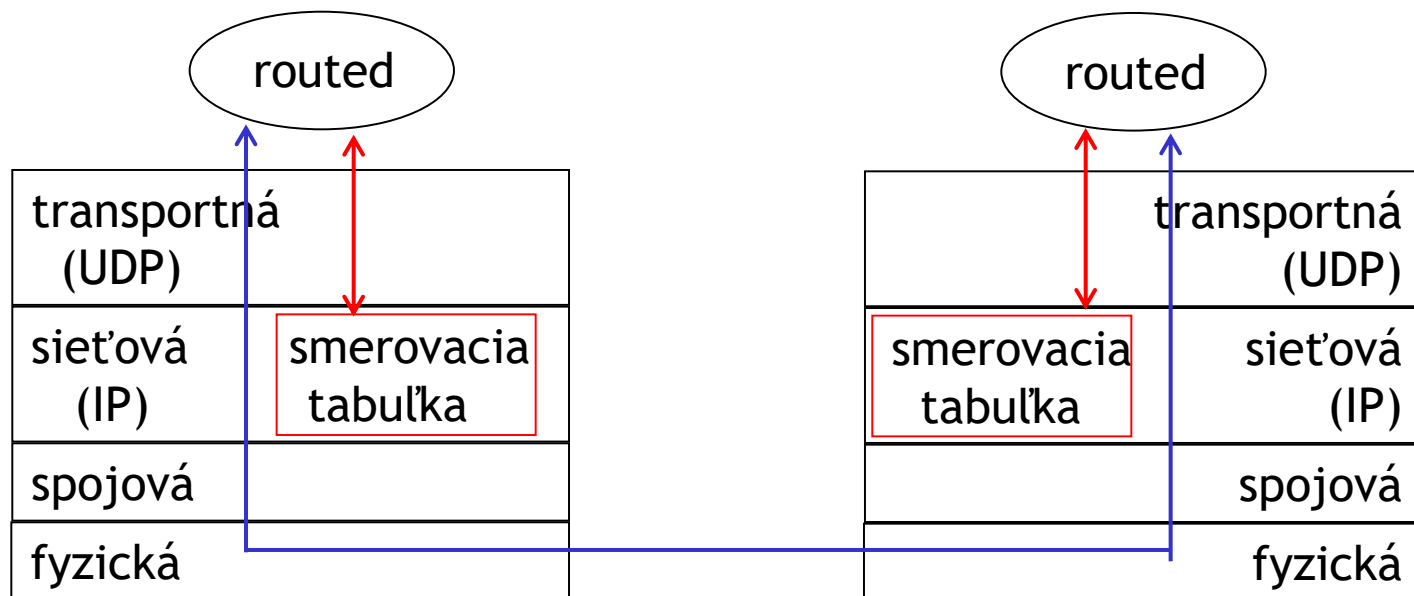
RIP: zotavenie z vypadnutia spoja

Ak neprídu od suseda žiadne vektory po dobu 180 s -> spojenie sa považuje za “mŕtve”

- ❖ cesty cez tohto suseda sa stanú neplatnými
- ❖ nový vektor je zaslaný susedom
- ❖ susedia zase pošlú svoje vektory, ak sa im zmenili tabuľky
- ❖ informácia o chybe spoja sa postupne šíri po “celej” sieti

RIP je aplikačný protokol

- ❑ RIP smerovanie je riadené aplikáciou `routed`
- ❑ vektory sú transportované v UDP segmentoch



Prehľad prednášky

❑ Smerovacie algoritmy

- ❖ LSA

- ❖ DVA

❑ Protokoly smerovania vnútri autonómnych systémov

- ❖ RIP

- ❖ **OSPF**

❑ Hierarchické smerovanie

❑ Protokol smerovania medzi autonómnyimi systémami : BGP

❑ Princípy smerovania broadcastu

❑ Princípy smerovania multicastu

❑ Protokoly multicastového smerovania

- ❖ DVMRP

- ❖ PIM

OSPF (Open Shortest Path First)

- ❑ “open”: otvorený softvér
- ❑ používa LSA
 - ❖ rozposielanie susediacich stavov všetkým uzlom
 - ❖ topológia je známa všetkým uzlom, aktualizovaná aspoň raz za 30 minút
 - ❖ počítanie ciest cez Dijkstrov algoritmus
- ❑ rozposielané dáta idú priamo v dátach IP paketu

“prídavné” funkcie OSPF oproti RIP

- ❑ **bezpečnosť**: všetky OSPF správy sú autentifikované, aby sa zabránilo neoprávneným odosielateľom
- ❑ umožňuje uchovať viac ciest s rovnakou cenou
- ❑ pre každý spoj viac metrík pre rôzne typy služieb
- ❑ integrovaná podpora pre unicast aj **multicast**
- ❑ **hierarchická verzia** OSPF pre veľa routrov.

Prehľad prednášky

❑ Smerovacie algoritmy

- ❖ LSA

- ❖ DVA

❑ Protokoly smerovania vnútri autonómnych systémov

- ❖ RIP

- ❖ OSPF

❑ Hierarchické smerovanie

❑ Protokol smerovania medzi autonómnyimi systémami : BGP

❑ Princípy smerovania broadcastu

❑ Princípy smerovania multicastu

❑ Protokoly multicastového smerovania

- ❖ DVMRP

- ❖ PIM

Hierarchické smerovanie

- ❑ RIP a OSPF sa nedajú použiť globálne pre všetky routre sveta
- ❑ nie všetky routre sú rovnaké a rovnako výkonné

na svete sú milióny routrov:

- ❑ nemôžeme si uchovávať všetky v tabuľke
- ❑ správy na výpočet cien ciest by zahltili internet!

nezávislosť administrácie:

- ❑ internet = sieť sietí
- ❑ každý sieťový administrátor môže potrebovať vlastné riešenie routovania jeho sietí

Hierarchické smerovanie

- spájanie routrov do regiónov: “autonómnych systémov” (AS)

- routre v tom istom AS komunikujú rovnakým smerovacím protokolom

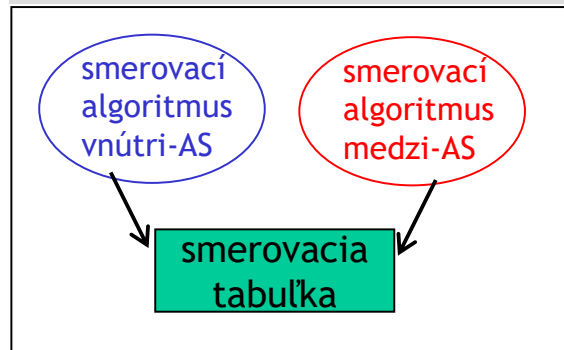
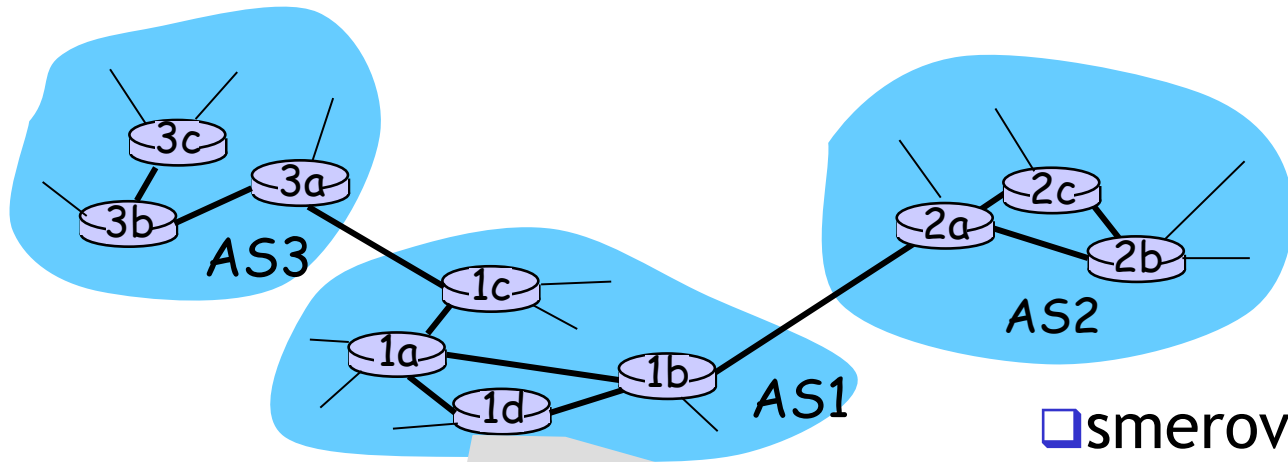
- ❖ smerovací protokol “vnútri-AS”

- ❖ routre v rôznych AS môžu používať rôzny smerovací protokol vnútri-AS

Gateway router

- má priame spojenie s routrom z iného AS

Prepojené autonómne systémy



□ smerovacia tabuľka je menená smerovacím algoritmom aj vnútri-AS aj medzi-AS

❖ alg. vnútri-AS nastavuje riadky pre vnútorné cieľové uzly

❖ alg. medzi-AS aj vnútri-AS nastavujú riadky pre externé cieľové uzly

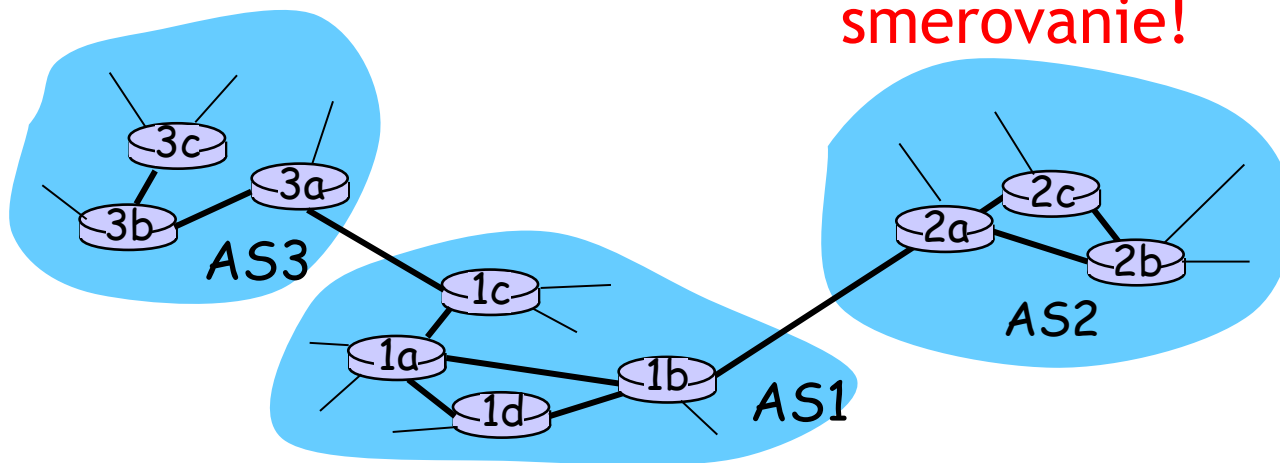
Úlohy algoritmov medzi-AS

- predpokladajme, že router v AS1 dostane paket pre sieť mimo AS1
- ❖ router by mal nasmerovať paket ku gateway routru jeho AS1, ale ku ktorému?

AS1 musí:

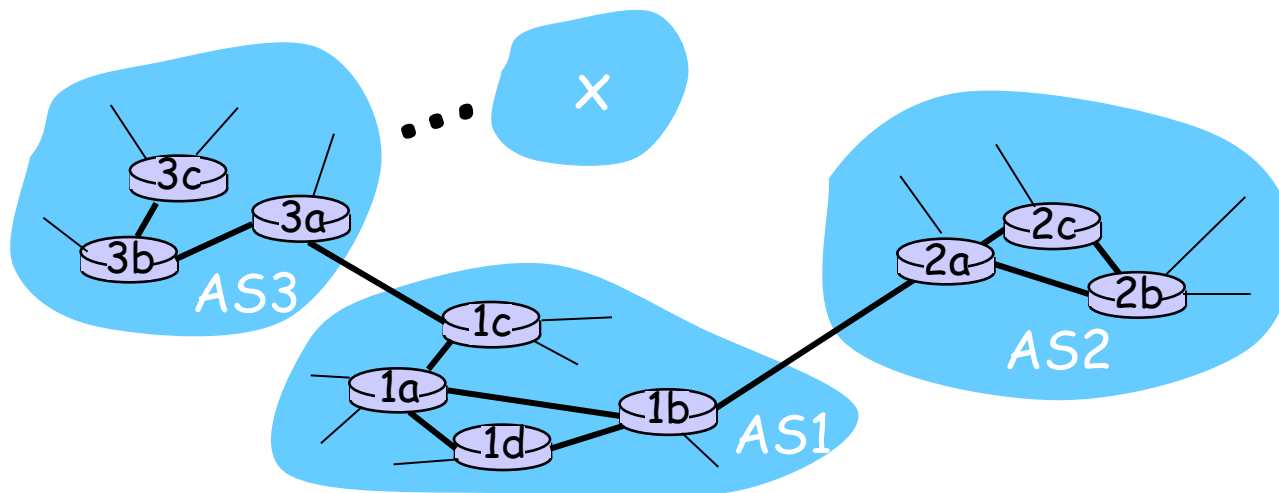
1. naučiť sa, ktoré adresy sú dostupné cez AS2 a ktoré cez AS3
2. zaznamenať túto dostupnosť na všetkých routroch v AS1

Práca pre medzi-AS smerovanie!



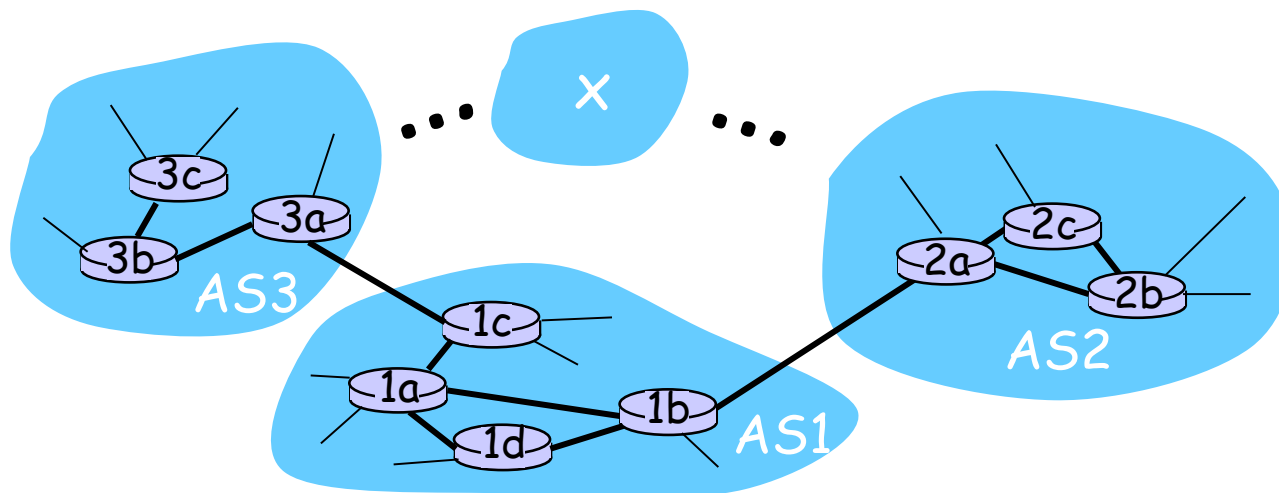
Príklad nastavenia smerovacej tabuľky v routri 1d

- predpokladajme, že AS1 sa naučí (cez medzi-AS protokol), že sieť x je dostupná cez AS3 (gateway 1c), ale nie cez AS2 (gateway 1b)
- medzi-AS protokol informuje všetky vnútorné routre o tejto dostupnosti.
- router 1d zistí cez protokol vnútri-AS, že jeho rozhranie r smeruje na najkratšiu cestu k 1c.
- ❖ pridá do smerovacej tabuľky záznam (x,r)



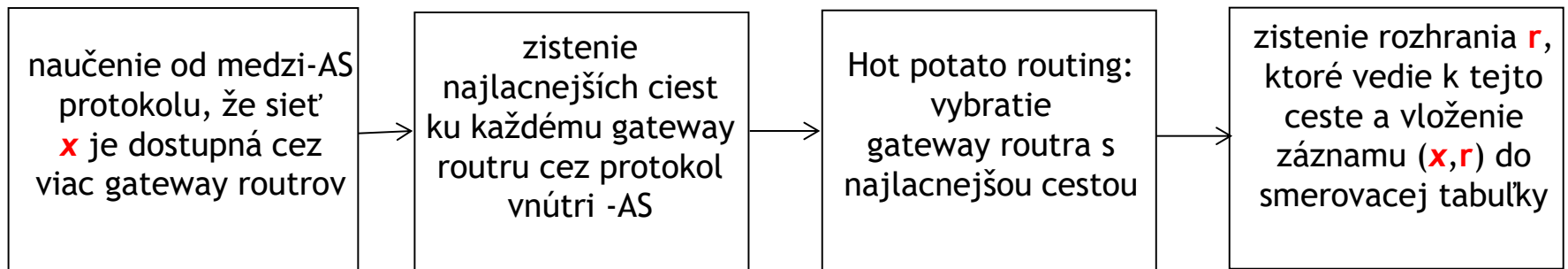
Príklad: výber z viacerých AS

- teraz predpokladajme, že AS1 sa naučilo od medzi-AS protokolu, že sieť **x** je dostupná cez AS2 aj AS3
- router 1d musí zistiť, cez ktorý gateway router má posielat' pakety do siete **x**
- ❖ aj toto je úloha pre protokol vnútri-AS



Príklad: výber z viacerých AS

- ❑ teraz predpokladajme, že AS1 sa naučilo od medzi-AS protokolu, že sieť **x** je dostupná cez A2 aj A3
- ❑ router 1d musí zistiť, cez ktorý gateway router má posielat' pakety do siete **x**
- ❖ aj toto je úloha pre protokol vnútri-AS
- ❑ **Hot potato routing** (smerovanie horúceho zemiaku)
- ❖ poslanie paketov cez bližší gateway router



Prehľad prednášky

- ❑ Smerovacie algoritmy
 - ❖ LSA
 - ❖ DVA
- ❑ Protokoly smerovania vnútri autonómnych systémov
 - ❖ RIP
 - ❖ OSPF
- ❑ Hierarchické smerovanie

- ❑ Protokol smerovania medzi autonómnymi systémami : BGP
- ❑ Princípy smerovania broadcastu
- ❑ Princípy smerovania multicastu
- ❑ Protokoly multicastového smerovania
 - ❖ DVMRP
 - ❖ PIM

Internetové medzi-AS smerovanie: BGP

□ BGP (Border Gateway Protocol)

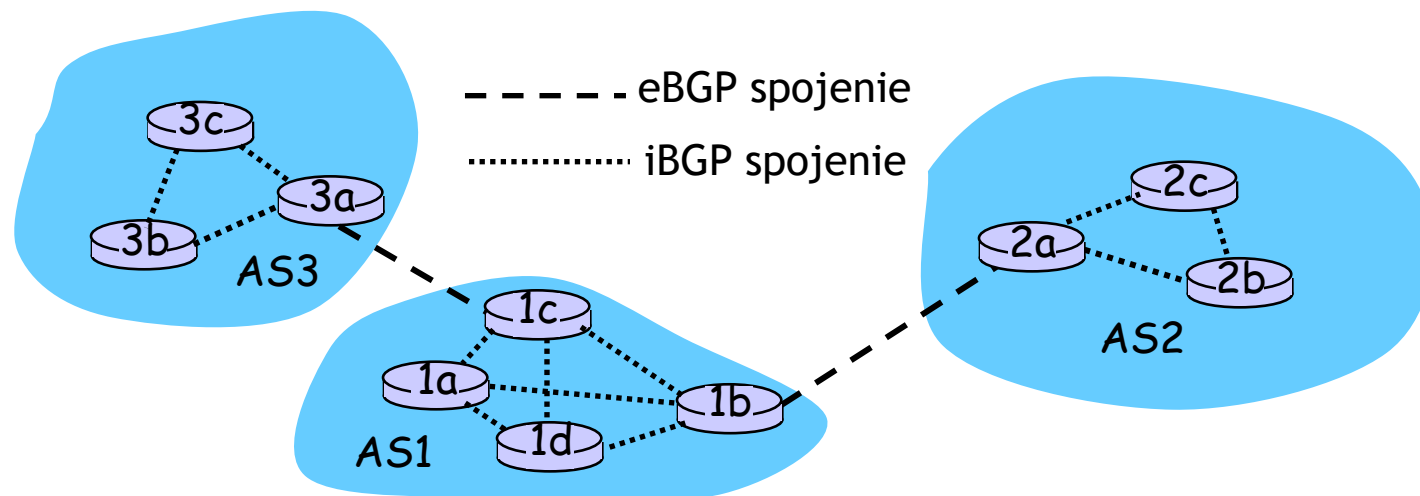
□ BGP poskytuje každému AS:

1. možnosť získať dostupnosti sietí od okolitých AS
2. rozoslanie dostupností k všetkým routrom vnútri AS
3. určenie “vhodných” ciest k cudzím sieťam na základe dostupnosti a pravidiel

□ umožňuje sieťi rozširovať informáciu o svojej existencii zvyšku internetu (oznámenie - advertisement)

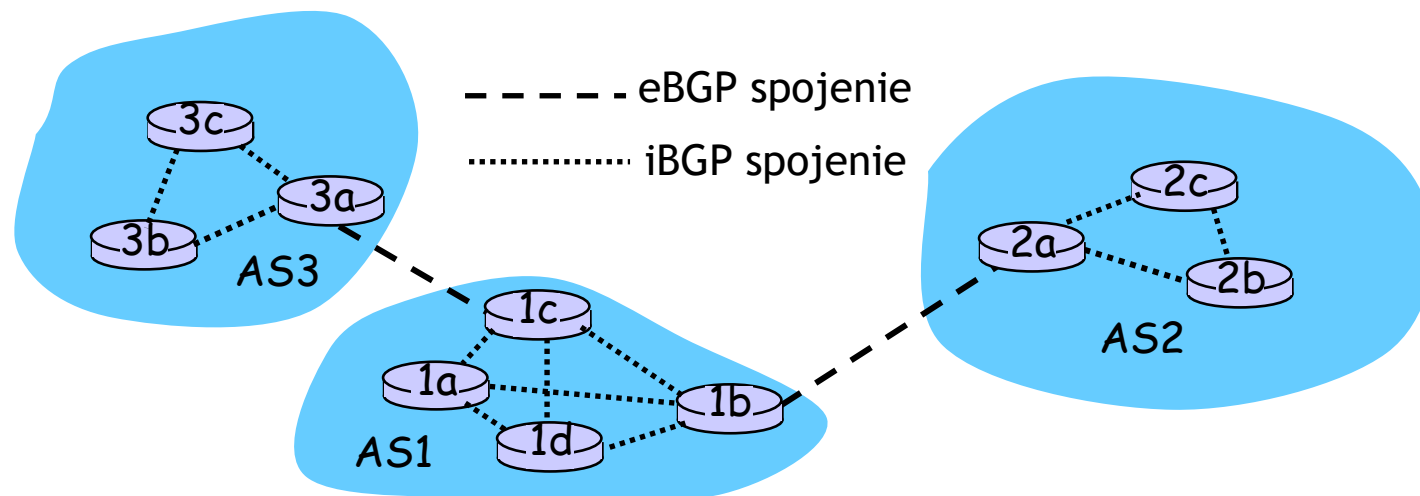
Základy BGP

- dvojice routrov (BGP peerov) si vymieňajú smerovacie informácie cez semi-permanentné TCP spojenia: **BGP spojenia**
- ❖ BGP spojenia nemusia zodpovedať fyzickým spojom
- keď AS2 odošle oznámenie, že má dostupnosť na nejakú sieťovú adresu pre AS1:
- ❖ AS2 sľúbi, že bude smerovať datagramy pre túto sieť
- ❖ AS1 môže zahrnúť túto sieť v oznámeniach o dostupnosti iným AS cez seba



Distribúcia oznámenia o dostupnosti

- ❑ AS3 pošle oznámenie o dostupnosti k AS1 pomocou eBGP spojenia medzi 3a a 1c
- ❖ 1c môže použiť iBGP spojenia na distribúciu nových informácií k všetkým routrom v AS1
- ❖ 1b môže postúpiť informácie o novej dostupnosti pre AS2 cez eBGP spojenie medzi 1b a 2a
- ❑ v routroch AS1 sa pre novú sieť vytvorí príslušný záznam v smerovacích tabuľkách



BGP: atribúty k informáciám o dostupnosti sietí

□ dva dôležité atribúty:

❖ **AS-PATH**: obsahuje postupnosť AS, cez ktoré informácia o dostupnosti danej siete prechádzala: napr. AS 67, AS 17

❖ **NEXT-HOP**: označuje konkrétny gateway router v danom AS, cez ktorý sa dá dostať von z AS smerom k danej sieti (reálne môže byť viac možností dostupnosti danej siete - cez viac gateway routrov)

□ keď gateway router dostane informáciu o dostupnosti siete, použije **importnú politiku** na jej akceptovanie alebo zamietnutie.

BGP: výber smeru

- router sa môže naučiť viac smerov k rovnakému cieľu a musí sa pre niektorý rozhodnúť
- pravidlá akceptovania/vylúčenia smeru:
 1. podľa politiky smerovania (administrátorské alebo dokonca manažérske rozhodnutia)
 2. najkratšia dĺžka postupnosti v AS-PATH
 3. najbližší NEXT-HOP router: hot potato routing
 4. ďalšie kritériá

BGP správy

❑ BGP správy sa prenášajú cez TCP

❑ BGP správy:

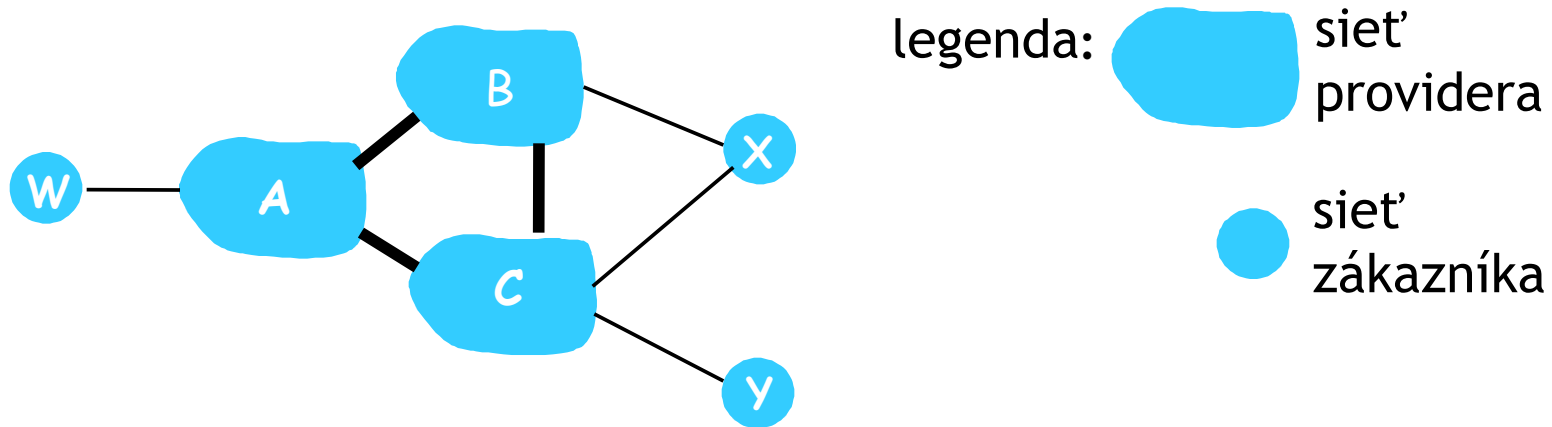
❖ **OPEN:** otvorí TCP spojenie k peerovi a autentifikuje sa

❖ **UPDATE:** pošle nové oznámenie dostupnosti alebo zruší staré

❖ **KEEPALIVE:** udržiava spojenie v prípade, že nie sú žiadne UPDATE správy; tiež potvrdzovacia správa na OPEN správu

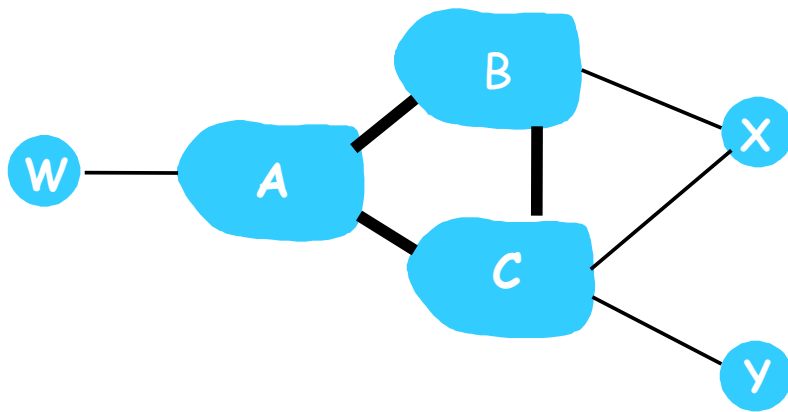
❖ **NOTIFICATION:** oznamuje chyby; zatvára spojenie



BGP: politika smerovania



- ❑ A, B, C sú **siete providerov (poskytovateľov pripojenia)**
- ❑ X, W, Y sú **zákazníci** daného providera
- ❑ X je pripojená do dvoch sietí providerov
- ❖ X nechce smerovať pakety z B do C cez svoju sieť
- ❖ .. takže X nebude posielat' do siete B oznámenie dostupnosti siete C

BGP: politika smerovania



legenda:  sieť
providera
 sieť
zákazníka

- ❑ A posielala cestu AW do siete B
- ❑ B posielala cestu BAW do siete X
- ❑ Má B posielat' cestu BAW aj do siete C?
- ❖ Nie! B nemá žiaden záujem na tom, aby fungovala cesta CBAW, keďže ani W ani C nie sú jeho zákazníci
- ❖ B chce prinútiť C, aby používalo cestu k W cez A
- ❖ B chce smerovať pakety iba pre svojich zákazníkov !
- ❖ Na druhej strane, ak padne spojenie A-C, tak C nemá alternatívu

Prečo rôzne smerovacie algoritmy vnútri-AS a medzi-AS?

Politika:

- ❑ **vnútri-AS:** administrátor chce kontrolu nad smerovaním vo vlastnom AS a kto smeruje cez jeho AS
- ❑ **medzi-AS:** mnoho administrátorov, nie je potrebné rozhodovanie o politike

Množstvo routrov:

- ❑ hierarchické smerovanie znižuje veľkosť tabuliek a množstvo potrebných smerovacích správ

Výkon:

- ❑ **medzi-AS:** zameriava sa na výkon
- ❑ **vnútri-AS:** politika môže dominovať nad výkonom

Prehľad prednášky

□ Smerovacie algoritmy

- ❖ LSA

- ❖ DVA

□ Protokoly smerovania vnútri autonómnych systémov

- ❖ RIP

- ❖ OSPF

□ Hierarchické smerovanie

□ Protokol smerovania medzi autonómnyimi systémami : BGP

□ Princípy smerovania broadcastu

□ Princípy smerovania multicastu

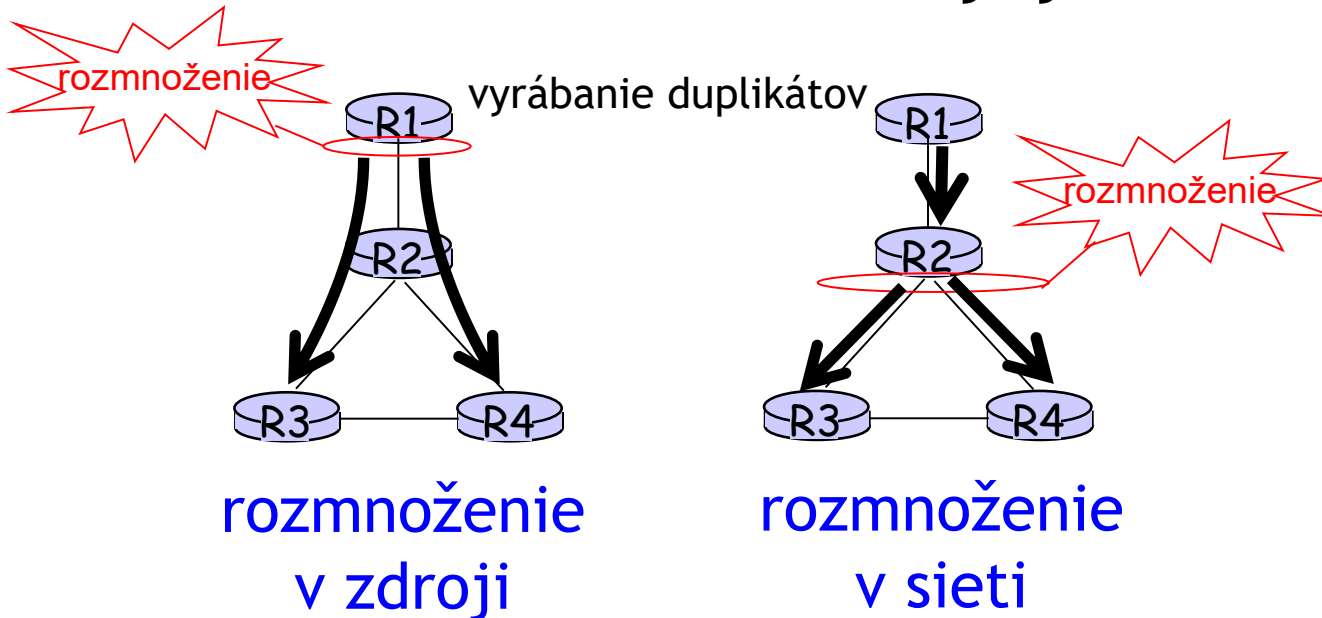
□ Protokoly multicastového smerovania

- ❖ DVMRP

- ❖ PIM

Smerovanie broadcastu

- ❑ doručenie paketu ku všetkým uzlom
- ❑ viacnásobné rozosielenie zo zdroja je neefektívne:



- ❑ rozmnoženie správy: ako zistíme, komu každému treba poslať správu?

Rozmnožovanie správy v sieti

❑ **zaplavenie:** keď router dostane broadcastový paket, pošle kópiu všetkým susedom, okrem toho, odkiaľ paket prišiel

❖ **problémy:** cykly a broadcastová búrka

❑ **kontrolované zaplavenie:** router rozosiela správu, iba pokiaľ už túto správu neposielal pred tým

❖ router si musí pamätať pakety, ktoré už rozosiela

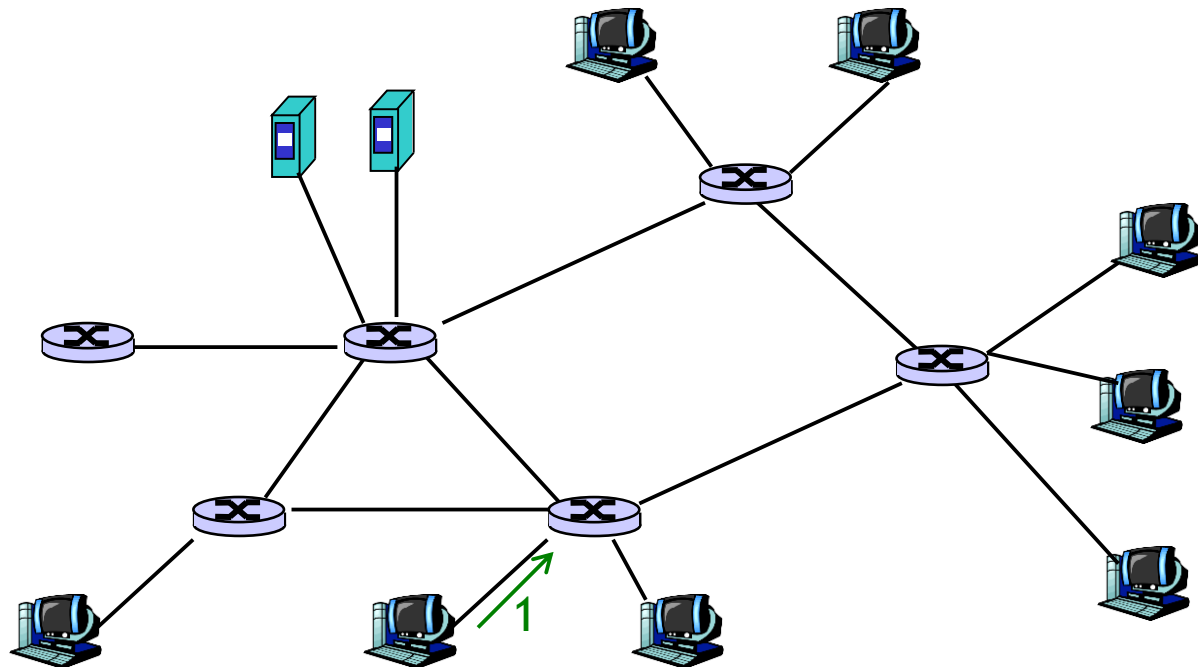
❖ alebo **reverse path forwarding (RPF):** rozosielame paket, iba ak prišiel z najkratšej cesty zo zdroja

❑ **spanning tree (minimálna kostra grafu)**

❖ žiadne redundantné pakety nedôjdu do žiadneho uzla

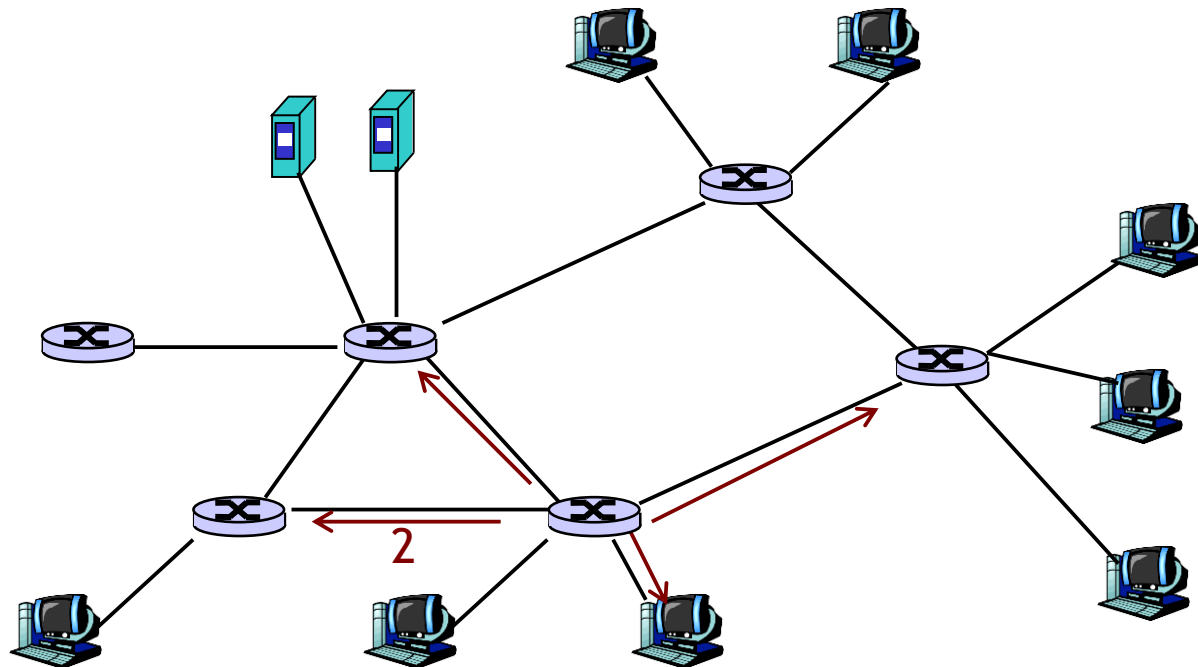
Zaplavenie - broadcastové búrky

- cykly zabezpečujú redundanciu v prípade poškodenia spoja
- broadcast však môže spôsobiť vážne problémy !



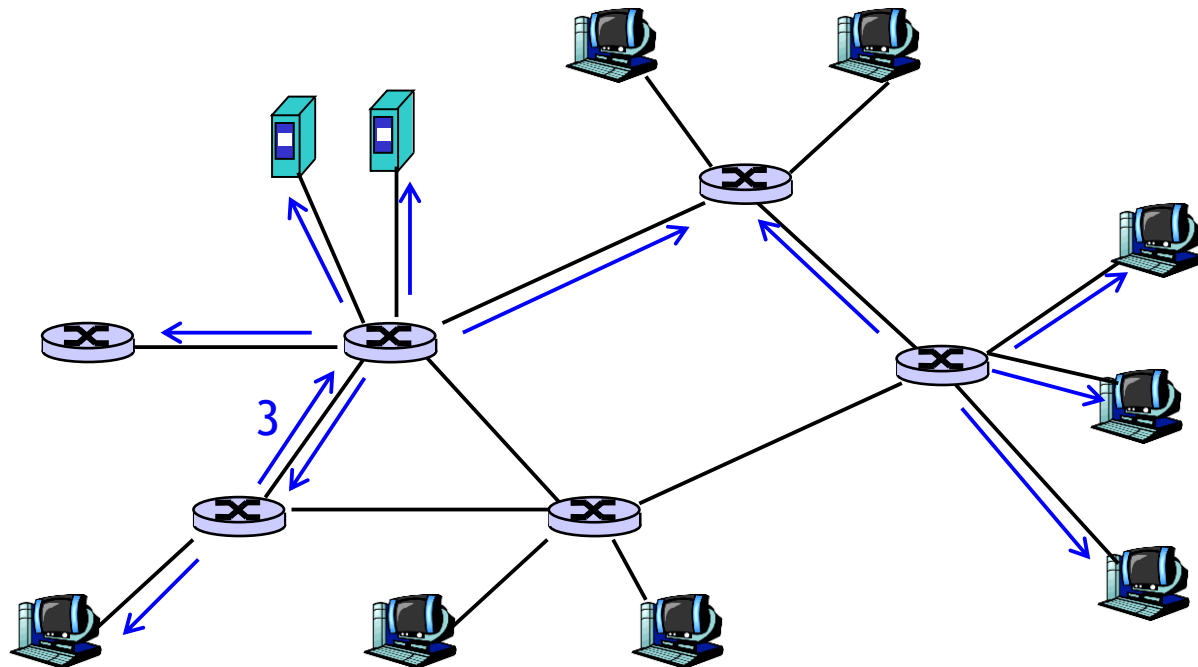
Zaplavenie - broadcastové búrky

- ❑ cykly zabezpečujú redundanciu v prípade poškodenia spoja
- ❑ broadcast však môže spôsobiť vážne problémy !



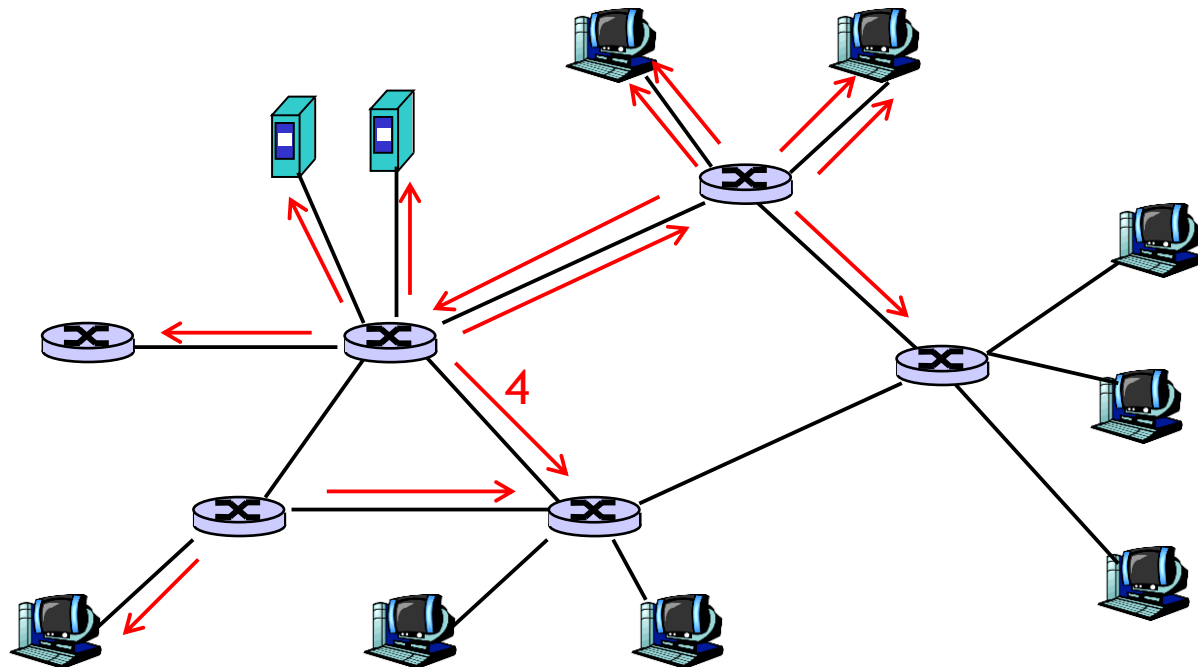
Zaplavenie - broadcastové búrky

- ❑ cykly zabezpečujú redundanciu v prípade poškodenia spoja
- ❑ broadcast však môže spôsobiť vážne problémy !



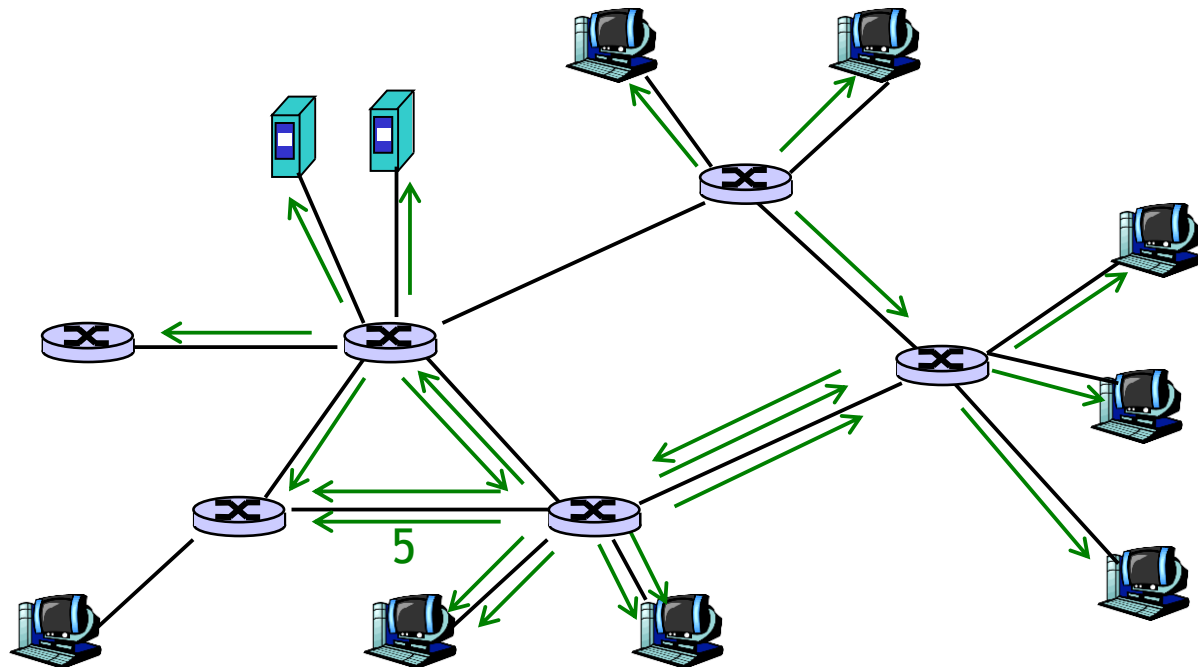
Zaplavenie - broadcastové búrky

- ❑ cykly zabezpečujú redundanciu v prípade poškodenia spoja
- ❑ broadcast však môže spôsobiť vážne problémy !



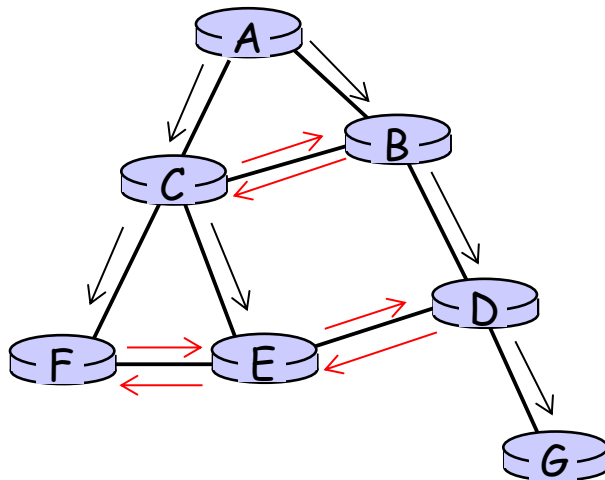
Zaplavenie - broadcastové búrky

- ❑ cykly zabezpečujú redundanciu v prípade poškodenia spoja
- ❑ broadcast však môže spôsobiť vážne problémy !

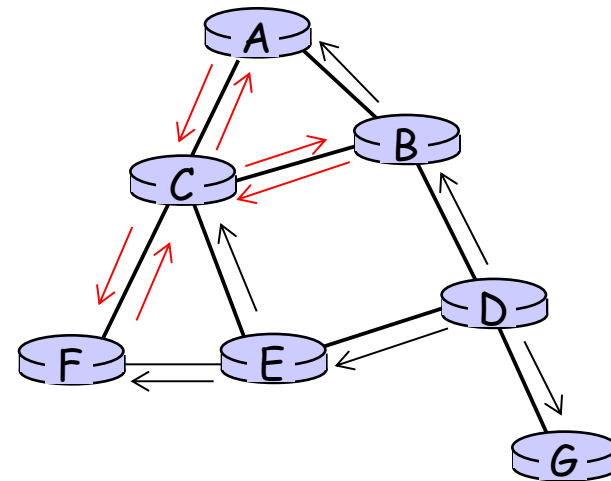


Reverse path forwarding

□ rozosielajú sa iba správy, ktoré prišli z najkratšej cesty od zdroja



(a) Broadcast z uzla A



(b) Broadcast z uzla D

červené správy nebudú preposielané ďalej

Reverse Path Forwarding

- spoliehanie sa na znalosť najkratšej cesty k zdroju zo smerovacích tabuliek routrov
- každý router spúšťa nasledovný algoritmus:

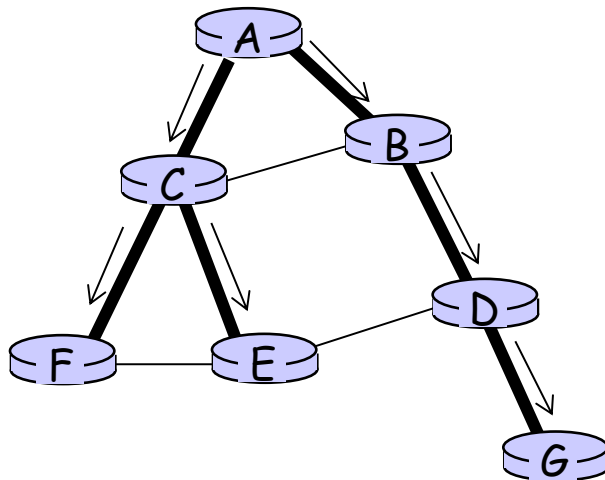
ak (broadcastový paket prišiel po spoji na najkratšej ceste k odosielateľovi)

potom rozpošli paket všetkým ostatným susedom

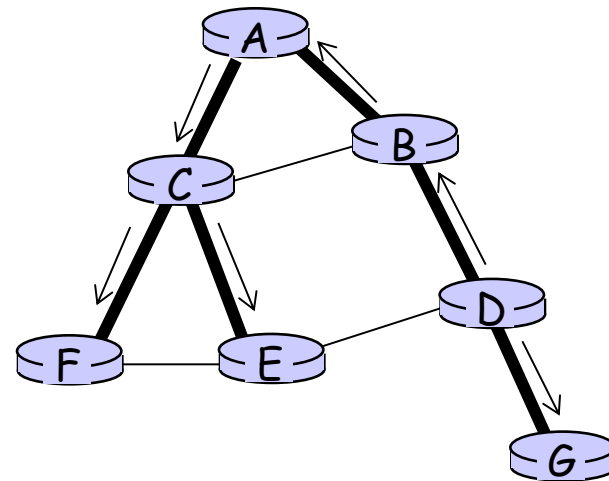
inak ignoruj paket

Spanning Tree (minimálna kostra grafu)

- Najprv sa vyrobí strom
- Uzly potom posielajú kópie správy iba po hranách vytvoreného stromu



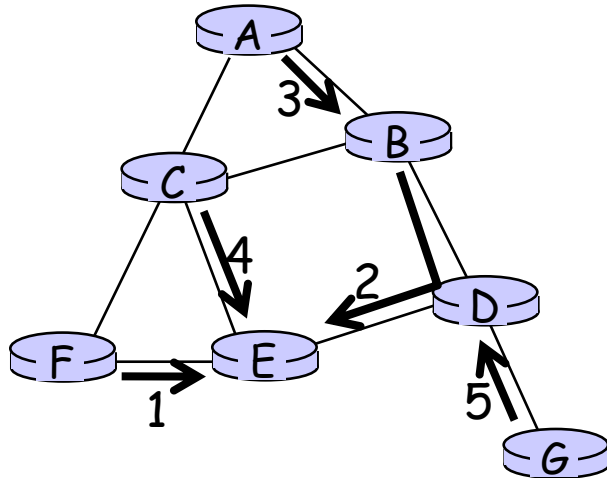
(a) Broadcast z uzla A



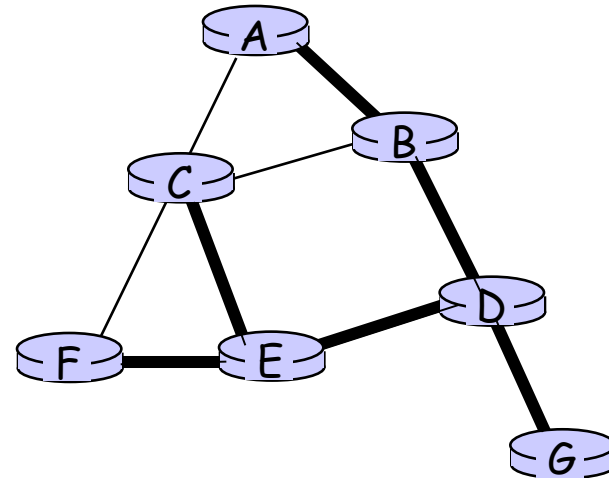
(b) Broadcast z uzla D

Spanning Tree: vytvorenie

- Centrálny uzol (rendezvous point)
- Každý z uzlov pošle správu k centrálnemu uzlu o pripojení do broadcastového stromu
- ❖ Správa je posielaná ďalej, iba pokiaľ nedôjde do uzla, ktorý už je v strome



(a) postup vytvorenia spanning tree



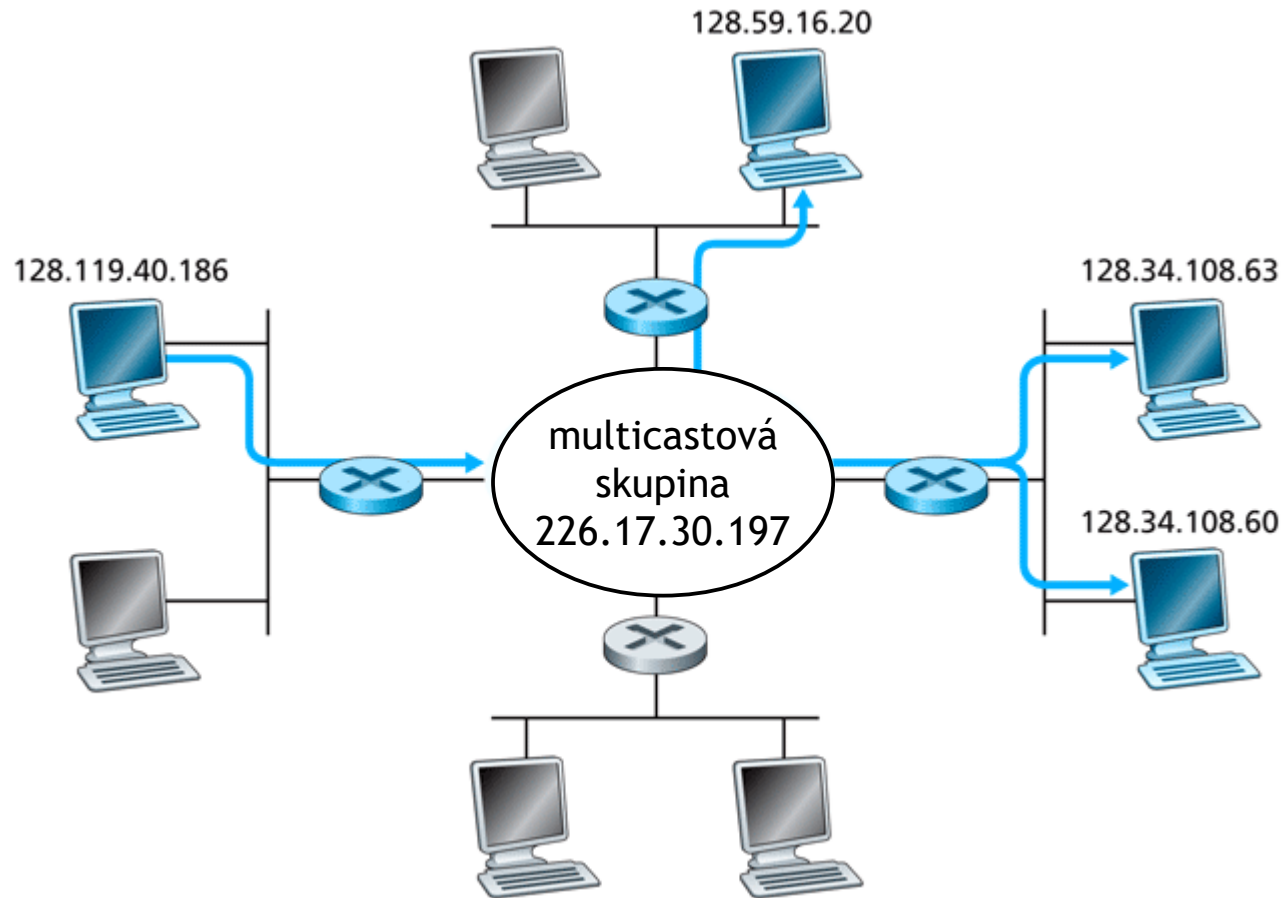
(b) Vytvorený spanning tree

Prehľad prednášky

- ❑ Smerovacie algoritmy
 - ❖ LSA
 - ❖ DVA
- ❑ Protokoly smerovania vnútri autonómnych systémov
 - ❖ RIP
 - ❖ OSPF
- ❑ Hierarchické smerovanie

- ❑ Protokol smerovania medzi autonómnymi systémami : BGP
- ❑ Princípy smerovania broadcastu
- ❑ Princípy smerovania multicastu
- ❑ Protokoly multicastového smerovania
 - ❖ DVMRP
 - ❖ PIM

Multicastové skupiny

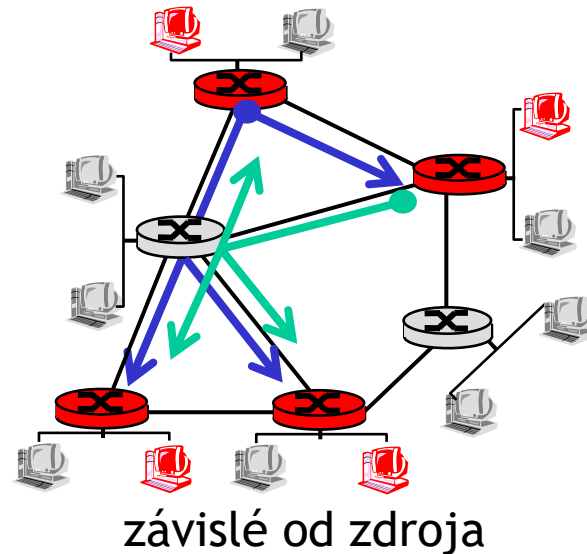
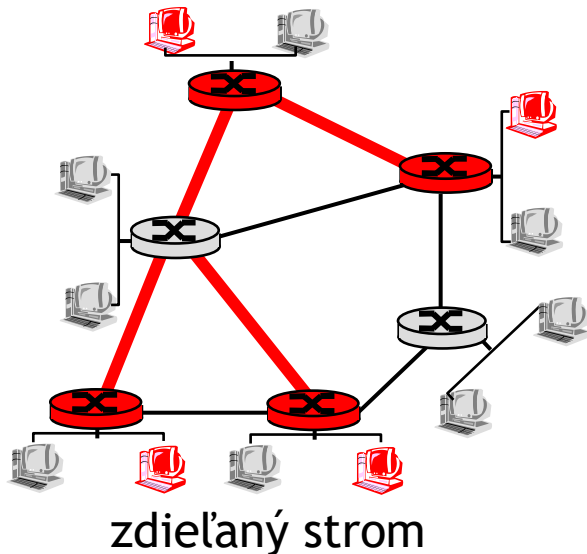


IGMP: Internet group management protocol

- ❑ podobne ako ICMP je prenášaná v tele IPv4 datagramu
- ❑ slúži na komunikáciu stanice s najbližším routrom, aby ho informovala o požiadavke na pripojenie alebo odpojenie od multicastovej skupiny
- ❑ router periodicky informuje o možných multicastových skupinách, o ktorých vie (napríklad preto, že iné stanice sú pripojené do týchto skupín)
- ❑ ak stanica neodpovie na informácie od routra, je automaticky odpojená zo skupiny.

Multicastové smerovanie

- **Ciel'**: nájsť strom prepojených routrov, ktoré majú v lokálnej podsieti členov multicastovej skupiny
- ❖ **strom**: nepoužijeme všetky cesty medzi routrami
- ❖ **závislé od zdroja**: strom môže byť pre každého odosielateľa v skupine iný
- ❖ **zdieľaný strom**: rovnaký strom pre všetkých členov

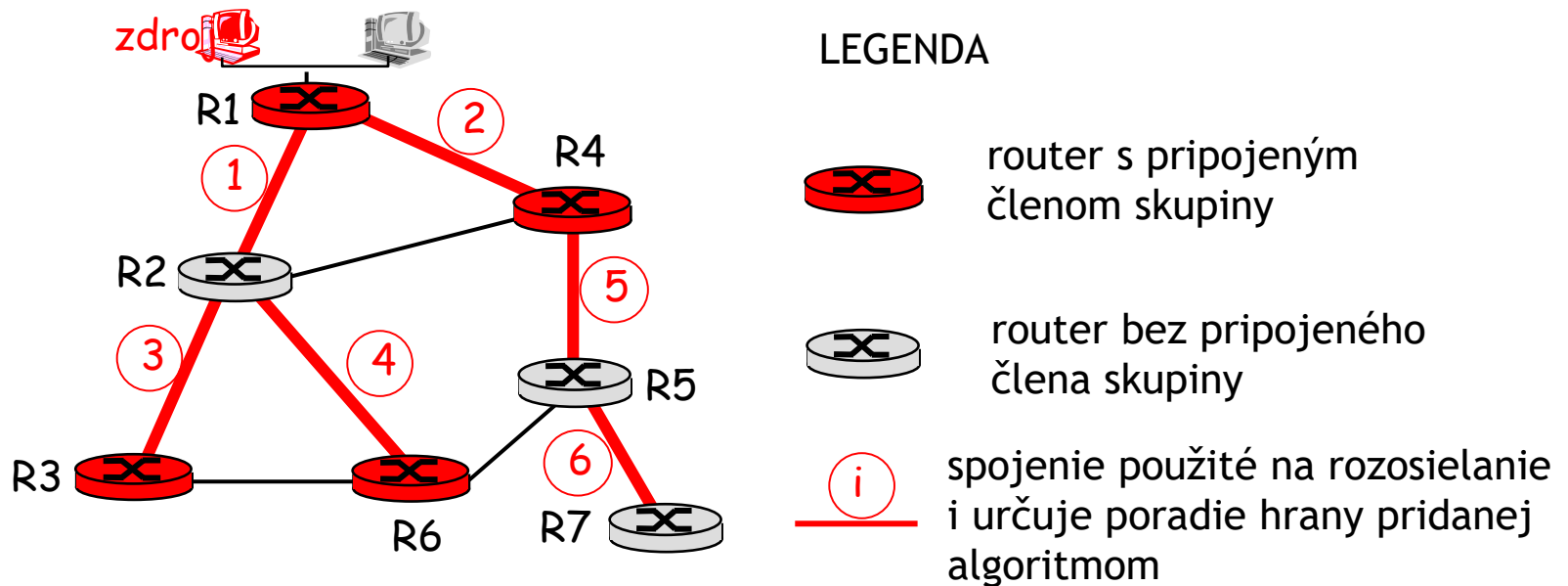


Prístupy k budovaniu multicastových stromov

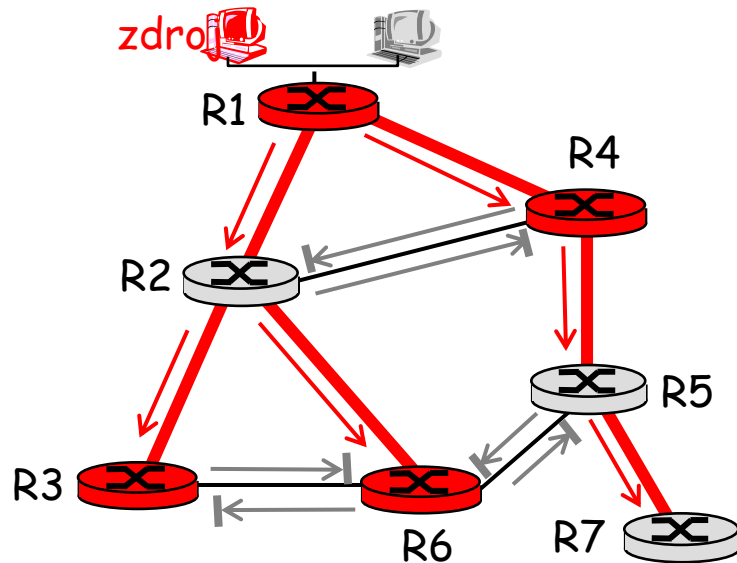
- ❑ **závislé od zdroja:** jeden strom pre každý zdroj vysielania
 - ❖ stromy najkratších ciest
 - ❖ reverse path forwarding
- ❑ **zdieľaný strom:** všetci vysielatelia majú rovnaký strom
 - ❖ stromy založené na centrálnom uzle

Strom najkratších ciest

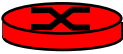



- Dijkstrov algoritmus (potrebné poznať topológiu)



Reverse Path Forwarding: príklad



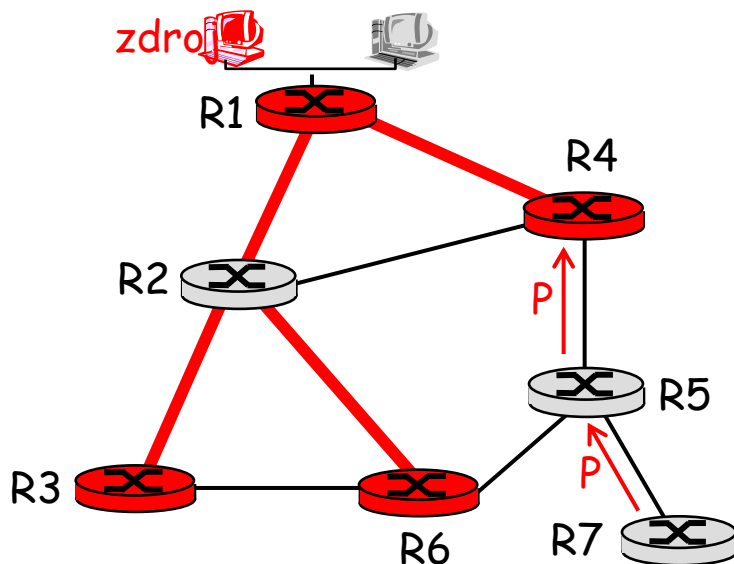
LEGENDA

-  router s pripojeným členom skupiny
-  router bez pripojeného člena skupiny
-  paket bude rozosielaný
-  paket nebude rozosielaný

- výsledkom je strom závislý od zdroja
- môže to byť nevýhoda pri nesymetrických spojeniach

Reverse Path Forwarding: orezávanie

- vzniknutý strom obsahuje podstromy, ktoré nemajú členov multicastovej skupiny
- ❖ nepotrebuje do nich rozposielať multicastové správy
- ❖ uzly bez pripojených členov multicastovej skupiny v podstrome posielajú “orezávacie” správy (prune message)



LEGENDA

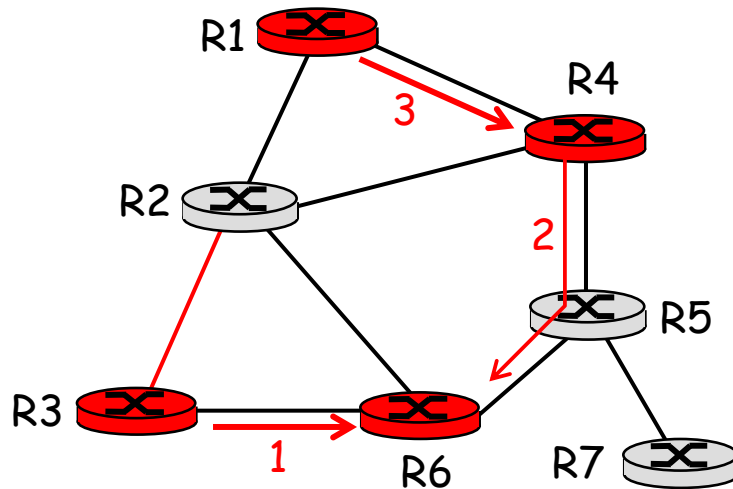
- router s pripojeným členom skupiny
- router bez pripojeného člena skupiny
- orezávacía správa
- spoje, ktoré budú prenášať multicastové správy

Stromy založené na centrálnom uzle




- ❑ **jediný strom** rozosielania **pre všetky zdroje**
- ❑ jeden z routrov je určený ako **”stred”** stromu
- ❑ napojenie (ako spanning tree):
 - ❖ router s pripojeným členom pošle pripájaciu správu smerom k centrálnemu uzlu
 - ❖ pripájacia správa sa spracuje na routroch na ceste
 - ❖ cesta, ktorou išla pripájacia správa, sa stane novou vetvou stromu
 - ❖ ak sa pripájacia správa dostane na router, ktorý už je pripojený v strome (v najhoršom centrálny uzol), správa sa už neposiela ďalej

Stromy založené na centrálnom uzle

Nech R6 je vybraný za centrálny uzol:



LEGENDA

-  router s pripojeným členom skupiny
-  router bez pripojeného člena skupiny
-  pripájacie správy s poradím odoslania v čase

Prehľad prednášky

- ❑ Smerovacie algoritmy
 - ❖ LSA
 - ❖ DVA
- ❑ Protokoly smerovania vnútri autonómnych systémov
 - ❖ RIP
 - ❖ OSPF
- ❑ Hierarchické smerovanie

- ❑ Protokol smerovania medzi autonómnymi systémami : BGP
- ❑ Princípy smerovania broadcastu
- ❑ Princípy smerovania multicastu
- ❑ Protokoly multicastového smerovania
 - ❖ **DVMRP**
 - ❖ PIM

Multicastové smerovanie na Internete: DVMRP

- ❑ **DVMRP**: distance vector multicast routing protocol, RFC1075
- ❑ **zaplav a orež**: reverse path forwarding, strom závislý od zdroja
- ❖ DVMRP má vlastné smerovacie tabuľky v routoch na komunikáciu medzi DVMRP routrami
- ❖ nespolieha sa na normálne smerovacie tabuľky
- ❖ úvodný paket na pripojenie je rozoslaný všetkými smermi cez reverse path forwarding
- ❖ routre, ktoré nechcú byť súčasťou danej multicast skupiny, posielajú späť orezávacie správy

DVMRP: ďalšie vlastnosti

- ❑ DVMRP router periodicky (1 min.) “zabúda”, ktoré vetvy boli orezané:
 - ❖ multicastové dáta sú opäť odoslané do pred tým orezaných vetiev
 - ❖ router, ktorému začínajú prichádzať tieto dáta môže opäť poslať žiadosť o odrezanie, alebo začať prijímať dáta
- ❑ routre sa môžu pripojiť aj rýchlejšie, nemusia čakať na zabudnutie
 - ❖ ak sa naňho napojená stanica chce pripojiť (cez IGMP)

Prehľad prednášky

❑ Smerovacie algoritmy

- ❖ LSA

- ❖ DVA

❑ Protokoly smerovania vnútri autonómnych systémov

- ❖ RIP

- ❖ OSPF

❑ Hierarchické smerovanie

❑ Protokol smerovania medzi autonómnyimi systémami : BGP

❑ Princípy smerovania broadcastu

❑ Princípy smerovania multicastu

❑ Protokoly multicastového smerovania

- ❖ DVMRP

- ❖ PIM

PIM: Protocol independent multicast

Dva režimy:

Hustý

- routre sú automaticky pripájané do skupiny pokiaľ sa tieto routre explicitne neodpoja
- predpokladáme, že takmer všetky routre sú zapojené do skupiny
- podobné ako DVMRP
- nepripojené routre sú “zbytočne” zaťažované

Riedky

- routre sa musia sami aktívne pripájať
- stromy založené na centrálnom uzle
- ideálne pre skupiny s jediným odosielateľom
- nepripojené routre nemusia robiť nič

PIM-DM (hustý režim)

zaplav a orež, podobné ako DVMRP, ale

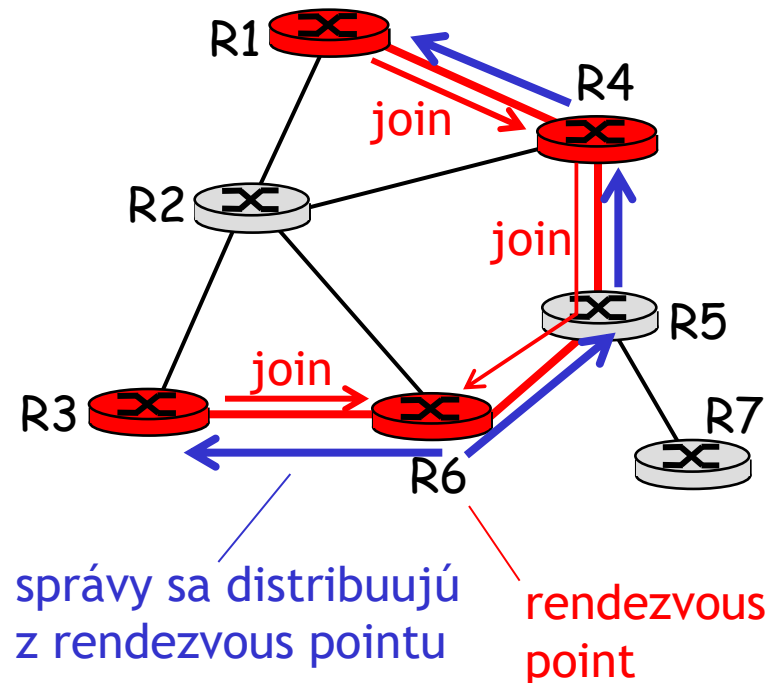
- využívajú sa tradičné smerovacie tabuľky routra na určenie, či správa prišla z najkratšej cesty

- menej komplikované a menej efektívne zaplavovanie ako DVMRP

- má mechanizmus na to, aby router zistil, že je listom multicastového stromu

PIM - SM (riedky režim)

- centrálny uzol = rendezvous point
- pripájanie sa do skupiny paketom smerom k tomuto uzlu
- odosielateľ najprv pošle správu na centrálny uzol a tá sa odtiaľ distribuuje do celého multicastového stromu



Zhrnutie

- ❑ Smerovacie algoritmy
 - ❖ LSA
 - ❖ DVA
- ❑ Protokoly smerovania vnútri autonómnych systémov
 - ❖ RIP
 - ❖ OSPF
- ❑ Hierarchické smerovanie

- ❑ Protokol smerovania medzi autonómnymi systémami : BGP
- ❑ Princípy smerovania broadcastu
- ❑ Princípy smerovania multicastu
- ❑ Protokoly multicastového smerovania
 - ❖ DVMRP
 - ❖ PIM

Ďakujem za pozornosť

Modifikované slajdy z knihy:

Computer Networking: A Top Down Approach ,
4th edition.

Jim Kurose, Keith Ross
Addison-Wesley, July 2007.