Otázky k lekci č. 9: Síťová vrstva a směrování

1. Jaké jsou hlavní úkoly sítové vrstvy a jakým způsobem může síťová vrstva fungovat? Přepojování okruhů vs. paketů atd.

- doručovat data od zdroje ke koncovým adresátům (čili přes mezilehlé uzly)
- je třeba: vyhledat cestu / rouhodnutí o dalším odchozím směru - **routing**
  přenést přes mezilehlé uzly / předání paketu v odchozím směru) - **forwarding**
- další úkoly: **congestion control** (předcházení zahlcení) - podobné řízení toku, ale není identické
  může zajišťovat **QoS**
- přenos. služby, jež poskytuje transportní vrstvě by měly být:
  funkční
  nezávislé na konkrétní topologii sítě
  nezávislé na konkrétní přenosové technologii dílčích sítí
  adresování by mělo být jednotné (např. IP v TCP/IP)
- možnosti fungování:
  přepojování okruhů (nezapadá do vrstevnatého modelu)
  přepojování paketů
    virtuální okruhy (spojovaná varianta)
      spolehlivě/nespolehlivě
      QoS/BE
    datagramy (nespojovaná varianta)
      nespolehlivě + BE

2. Charakterizujte rozdíl mezi virtuálními okruhy a datagramovou službou

virtuální okruhy

- přenos paketů
- pakety obsahují identifikátor virtuálního okruhu
- pakety vždy stejnou cestou (čili správné pořadí při doručení)
- nutné navázat spojení (dojde k vyhledání cesty) i ukončit
- routing pouze jednou (při spojování)
- forwarding podle předem určeného směru (z routingu)
- stavový (navázání/ukončování/výpadek...)
- neumí reagovat na intenzivnější změny v síti (vhodné pro velké objemy dat)

datagram. služba

- přenos datagramů
- datagram opatřen celou adresou příjemce
- nemusí cestovat vždy stejnou cestou
- bezestavový (nenavazuje ani neukončuje spojení)
- spojení se nenavazuje (routing pokaždé znovu)
- umí reagovat na průběžné změny v síti

3. Co je směrovač a jaké úkoly plní?

- **router**
- zajišťuje propojení na úrovni síťové vrstvy
- v prostředí s přepojováním paketů řeší routing i forwarding
- propojuje 2 a více sítí
- **přepínač** (switch) je zařízení propojující na linkové vrstvě
- **opakovač**(repeater) propojuje na fyzické vrstvě

4. Jaký je účel směrovacích tabulek? Jaké údaje obsahují? Kdo aktualizuje jejich obsah?

- umístěny na síťové vrstvě
- směrovač musí mít informace o topologii sítě (kromě "zvláštních" metod)
- tyto informace udržuje v tzv. směrovací tabulce
- u adaptivních směrovacích algoritmů tabulky průběžně aktualizovány
- u neadaptivních lze naplnit jednorázově předem
- aktualizuje je aplikační proces (**route demon**), který také u adaptivních tabulku neustále aktualizuje
- tabulka obsahuje **<cílová síť; vzdálenost; odchozí směr>**
- paket obsahuje celou adresu -> použita jako klíč pro nalezení odchozího směru

5.Jaké existují varianty směrování? Uveďte základní způsoby klasifikace.

směrovací algoritmy

- neadaptivní:
  - nereaguje na průběžné změny
  - trasy spočítá předem
  - při výpadcích částí může způsobit výpadek celku
- adaptivní:
  - reaguje na průběžné změny
  - trasy počítá průběžně
  - odolnější vůči výpadkům

klasifikace směrování

- centralizované:
  - routing provádí centralizovaný server(route server)
  - směrovače provádí pouze forwarding
- izolované:
  - směrovače routují i forwardují
  - uzly nespolupracují na hledání cest
  - méně efektivní, než když spolupracují
- distribuované:
  - směrovače routují i forwardují
  - spolupracují na hledání cest
- hierarchické:
  - řeší obecný problém směrování - informací neúnosně mnoho
  - řeší dekompozicí - soustavu sítí rozdělí na menší podcelky, kde se směruje samostatně

6.Jak funguje centralizované směrování?

- počítá s existencí centrálního objektu (**route serveru**), jež počítá trasy a výsledek distribuuje ostatním směrovačům (ty si ukládají do cache)
- algoritmus může být adaptivní i neadaptivní
- s výpadkem route serveru síť nefunguje
- v praxi moc nevyužívané (spíše distribuované a izolované)

7.Jaké existují varianty izolovaného směrování?

- **záplavové směrování**
  - rozešle všemi směry, kromě příchozího
  - nepotřebuje směr. tabulky
  - problémy s opakováním
- **metoda horké brambory**
  - jsou-li vstupní fronty plné, pošli nejvolnějším směrem
  - doplňková metoda, kdy jiná vede k tvoření velkých front
- **metoda náhodného směrování**
  - odešli náhodným směrem
  - doplňková metoda
- **metoda zpětného učení**
  - napočátku směrovač neví nic a směruje záplavově
  - pak přijme paket od uzlu A ze směru S1 a odvodí, že A leží ve směru S1
- **source routing**
  - každý rámec si nese seznam uzlů, přes který má projít (Sestavuje odesílací uzel)
  - technika užívaná na linkové vrstvě!!! (ačkoliv "směrování" naznačuje síťovou vrstvu)

8.Jak funguje záplavové směrování?

- každý paket rozeslán do všech směrů, kromě příchozího
- existuje-li cesta k cíli, je nalezena(dokonce ta njkratší)
- nevyžaduje tabulky
- problémy s eliminací nadbytečných paketů
  - řešení vkládáním čítačů do všech paketů (dojde-li čítač k nule, je paket eliminován)
  - nebo pamatováním všech již prošlých uzlů v paketu a eliminací duplikátů
- využití:
  - pro běžná použití ve speciálních sítích (např vojenské)
  - pro speciální data (např. hledání cesty, aktualizační informace
  - pro speciální účely (např. pro aktualizaci distr. databází)

9.Jak funguje metoda zpětného učení (jako varianta směrování)?

- každý směrovač si zpětně získává informace z paketů jím procházejících
- na začátku neví nic, pak přijde paket od uzlu A ze směru S1, a uloží si, že A leží ve směru S1. Pak přijde paket od B pro A ze směru S2 a pošle směrem S1 a uloží si, že B leží ve směru S2
- používá se spíše na linkové vrstvě
- při směrování do velkých sítí nevhodné, spíše využíváno u ethernetových mostů a přepínačů

10.Jak funguje source routing (jako varianta směrování)?

- prováděné zdrojem
- každý rámec si v sobě od vysílacího uzlu nese celý seznam uzlů, kterými má projít
- má blíže k síťové vrstě, ale používá se na linkové vrstvě!!!
- odesílací uzel před odesláním vyšle průzkumný uzel. Ten se šíří záplavově, až dorazí ke svému cíli. Pak se vrátí a vrátí cestu, kterou se k cíli dostal
- záplavové směrování nešetrné k přenosové kapacitě, ale nalezne nejkratší cestu, není to ale příliš adaptivní

11.Jaké jsou základní varianty distribuovaného směrování?

- nejčastější
- směrovače vzájemně spolupracují - 2 možnosti:
    1) - výpočet je distribuovaný (každý počítá kus a vzájemně si předávají části výpočtů)
        - když jeden udělá chybu, pak splete i ostatní
    2) - každý počítá optimální cesty sám, posílají si jen podklady (informace o dostupnosti a změnách)
- výpočet cesty klasicky jako v teorii grafů (Ford-Fulkerson, nebo Bellman-Ford)
- důležité z hlediska zatížení sítě, jaké objemy si posílají a jak často

varianty

- **vector distance routing** (např. protokol **RIP**), vhodné pro menší sítě
- **link state routing** (např. protokol **OSPF**), vhodné i pro větší sítě

12.Jaké údaje si vyměňují uzly při směrování "vector distance" a jak často?

- každý směrovač si udržuje tabulku svých nejmenších vzdáleností od všech ostatních uzlů
- směrovače si tyto informace vyměňují (informace typu: já se dostanu k X za cenu Y)
    - jde vpodstatě o průběžnou výměnu obsahu celých směr. tabulek
- výměna ale probíhá jen mezi přímými sousedy!!!
- informace si posílají při změně(nalezení kratší cesty, neprůchodnost cesty)
- existence kratší cesty se šíří rychle
- neprůchodnost pomalu
- problém count-to-infinity: hodnota cesty přs neprůchodnou cestu se pořád zvyšuje o 1, trvá to dlouho, než se hodnota zvýší natolik, aby signalizovala neprůchodnost
- udělá-li někdo chybu, splete i ostatní

13.Jaké údaje si vyměňují uzly při směrování "link state" a jak často?

- po zapnutí si každý uzel zjistí, jaké má přímé sousedy
- uzel průběžně zjišťuje odezvy sousedů (ohodnocení hran)
- každý uzel pravidelně sestavuje paket, kam dá naměřené hodnoty odezvy svých přímých sousedů. Tento rozešle ostatním
- paket stačí rozesílat jen při změně
- každý směrovač postupně naakumuluje zprávy o stavu všech spojů v síti
- výpočet nejkratší cesty probíhá lokálně (pomocí Dijkstra)
- každý uzel počítá za sebe, případná chyba neovlivní ostatní uzly

14.Co je problém "count to infinity" u směrování vector distance a jak se řeší?

- problém **count-to-infinity**: hodnota cesty přs neprůchodnou cestu se pořád zvyšuje o 1, trvá to dlouho, než se hodnota zvýší natolik, aby signalizovala neprůchodnost
- řešení: - metoda **split horizon**
    - směrovač nebudei nzerovat zpátky (jde-li o cestu do A, kterou se Y dozví od X, pak Y
    - nebude cestu do A zpětně inzerovat uzlu X, od kterého ji získal)
  - doplněk **poisoned reverse**
    - uzel Y inzeruje zpět uzlu X dostupnost s hodnotou nekonečno
- v některých topologiích i toto řešení selhává

15.Jak funguje protokol RIP a jaká má omezení?

- Routing Information Protocol
- typu vector-distance
- metrikou je počet přeskoků (jen do maxima 15!!! nekonečno = 16)
- obsah směr. tabulky(distance vector) rozesílán každých 30 sekund všem sousedním směrovačům (obsahuje až 25 cílových sítí, rozesílá se jako UDP datagram na port 520)
- pokud není distance vector přijat do 180 sekund, je soused/spoj brán jako mrtvý, následně se použije metoda split horizon with poisoned reverse

- zpracování aktualizačních informací řeší routing démno na úrovni OS
- omezení: je špatně škálovatelný, nelze pro větší sítě(kvůli metrice 15), je málo stabilní

16.Jak funguje protokol OSPF a pro jaké sítě se hodí (srovnejte s protokolem RIP)

- **Open Shortest Path First**
- typu link-state (každý uzel testuje dostupnost svých sousedů)
- OSPF směrovač si udržuje :
    - databázi přímých sousedů
        - každý má jinou
        - každých 10 vteřin aktualizuje pomocí **HELLO paketů**
    - topologickou databázi - všichni by ji měli mít stejnou - pomocí této db počítá nejkratší cestu
    - směrovací tabulky - pro samotné směrování IP paketů
- nový OSPF směrovač tedy nejprve zjistí své sousedy, s každým si sesynchronizuje stou topologickou databázi,po úspěšné synchronizaci rozešlou všem ostatním směrovačům v síti informaci o existenci spojení.
- existující trvale monitoruje dostupnost přímých sousedů
    - pokud není změna, každých 30 minut opakuje všem své sousedství
    - pokud je změna, okamžitě informuje o změně ostatní směrovače
- OSPF/RIP
    - vnímá jen své sousedy / vnímá celou síť
    - výpočet distribuovaný / každý počítá sám
    - chyba ovlivní ostatní směrovače / neovlivní
    - aktualizace pravidělně (30 sekund) / stačí při změně(jinak jednou za 30 minut pro osvěžení)
    - aktualizační informace zasílá jen přímmým sousedům / všem uzlům v síti
    - hodí se pro malé sítě (díky omezení 15) / pro velké sítě vhodnější

17.Jak funguje hierarchické směrování a kdy je vhodné/nutné?

- ani link state algoritmy nejsou vhodné pro opravdu velké sítě
- problém je: ve velikosti sítí (dnes velmi velkné, že objem aktualizačních informací je neúnosný)
    - ve složitosti správy (směrování ve velkých soustavách sítí komplikované)
    - v rozdílných routovacích politikách různých provozovatelů
- řešení: rozdělit na menší části (jejich směrování řešeno **autonomně** (autonomní systém **AS**), nešíří podrobné směr. informace ven)
    - v rámci menších částí zachovat úplné směrování
    - do každé části vymezit jeden vstupní bod (nebo několik málo vstupních b.)
    - mezi částmi směrovat vždy přes jednotné vstupní body
- hierarch. směrování v OSPF
    - detailní směr. informace neopouští oblasti (areas)
    - hraniční směrovaše sbírají informace o dostupnosti sítí v oblasti a předávají ostatním hraničním s.
    - na úrovni páteřních sítí se vše opakuje

18.K čemu slouží směrovací protokoly IGP a EGP?

- Interior/Exterior Gateway Protocol

IGP

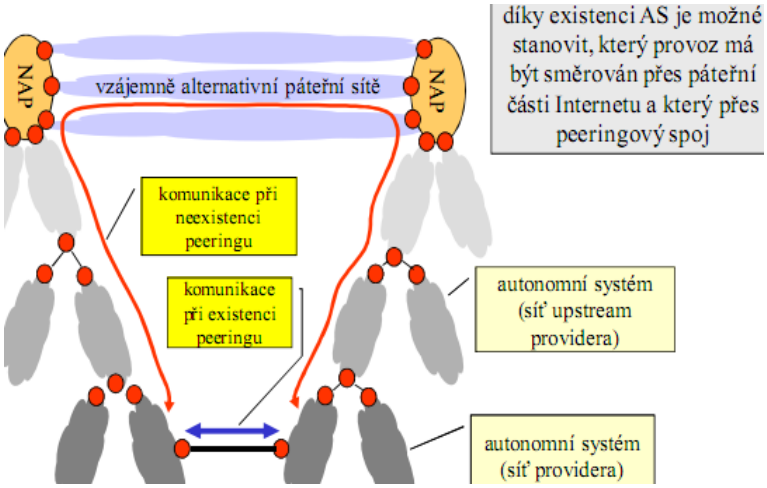
- provozovatelé samy mohou rozhodnout, jaké směrovací protokoly použijí v rámci **AS**(autonomního systému) i jednotlivých oblastech (areas)
- v úvahu připadají: OSPF, RIP, IGRP(Interior Gateway Routing Protocol), EIGPR (Enhanced ...)

EGP

- jsou nutné pro vzájemnou komunikaci mezi autonomními systémy (AS)
- umožňují definovat pravidla vzáj. komunikace mezi jednotlivými AS
- například: BGP (Border Gateway Protocol)

19.Jak souvisí autonomní systémy s peeringem?

- díky AS je možné stanovit, který proboz má být směrován přes páteřní části Internetu a který přes peeringový spoj



20.Jaké techniky se používají pro předcházení zahlcení?

- předcházení zahlcení: další úkol síťové vrstvy
- podobné řízení toku, ale jde o jiný problém
- **řízení toku**
    - point-to-point záležitost mezi jedním odesilatelem a jedním příjemcem (netýká se sítě mezi nimi)
- předcházení zahlcení
    - týká se zátěže celé sítě
    - dat. tok od všech odesilatelů se sčítá a neměl by překročit hranici, kterou je síť schopna zvládnout
- **zahlcení** - přenos. síť musí zahazovat pakety, neboť je nedokáže zpracovat
- jak řešit:
    - **dopředné techniky** (open loop)
        - neberou v potaz aktuální stav sítě
    - **zpětnovazebné techniky** (closed loop)
        - skrze zpětnou vazbu reagují na aktuální stav sítě

21.Jak fungují dopředné techniky pro obranu před zahlcením?

- řeší se snáze při užití virtuálních okruhů
- např. nově navazovano virtuální okruhy se vedou mimo zahlcené části. Při navazování spojení je uzavřen kontrakt se sítí (ten, kdo spojení navazuje říká, kolik zdrojů bude potřebovat - síť to buď akceptuje, nebo odmítne navázat spojení) -> takto funguje např. v ATM
- triviální řešení: předimenzovat síť
- **trafic conditioning**
    - obecně techniky ovlivňování dat. toku, aby lépe prošel
        - varianta **traffic policing** (co je nad limit, je eliminováno(zahozeno) již u odesilatele/po cestě)
        - varianta **traffic shaping**(snaha různě tvarovat dat. tok, např. pozdržet pakety, aby byly odeslány s rovnoměrnými odstupy

22.Jak fungují zpětnovazební techniky pro ochranu před zahlcením?

s explicitní zpětnou vazbou (odesilatel dostává od sítě explicitní informace, zda došlo/nedošlo/hrozí zahlcení) a podle toho mění své chování

- např. síť. vrstva TCP/IP
    - dat. pakety přenáší protokol IP
    - dojde-li k zahlcení, informuje odesilatele protokolem ICMP (internet control message protocol)
    - reakce odesilatele není definována (záléží na implementaci)

bez explicitní zpětné vazby (odesilatel sám usuzuje na to, zda došlo k zahlcení, či nikoliv, například z průběhu odezvy příjemce, z míry ztrátovosti paketů apod.)

- např. protokol TCP
    - zajišťuje spolehlivý přenos
    - dostává potvrzení od příjemce
    - pokud nedostane potvrzení v limitu, interpretuje to jako zahlcení