### *1. přednáška = síťové vrstvy*

1. Jaké jsou vlastnosti jednotlivých topologií sítí (sběrnice, hvězda, kruh, …)?
2. Jaké jsou vrstvy v OSI modelu?
3. Jaké jsou vrstvy v TCP/IP modelu?
4. Jaké má vlastnosti/Co zajišťuje (fyzická, linková, první, druhá, …) vrstva OSI modelu?

### *2. přednáška = Fyzická vrstva*

1. Významné vlastnosti kódování.
2. Typy kódů a jejich vlastnosti.
3. Metalická vedení a jejich vlastnosti
   a) symetrická a asymetrická vedení,
   b) typy symetrických vedení (UTP, STP, FTP)
   c) orientační znalost tříd UTP kabelů,
   d) přeslechy (NEXT, FEXT, útlum a přeslechy).
4. Optická vedení
   a) konstrukce vlákna (jádro, plášť, …),
   b) maximální úhel navázání a numerická apertura,
   c) jednovidová a vícevidová vlákna,
   d) vidová a chromatická disperze a její význam.

### *3. přednáška = Linková vrstva (protokoly, potvrzování, přístupové metody)*

1. Kódová vzdálenost a schopnost detekce a oprav chyb (znáte z jiných předmětů).
2. Potvrzovací schémata - popis a vlastnosti.
3. Přístupové metody (znáte z jiných předmětů)
   a) rozdíl mezi náhodným přístupem a deterministickými metodami,
   b) CSMA metody.
4. Protokoly linkové vrstvy - především Ethernet II, IEEE 802.3 a IEEE 802.2

### *4. přednáška = Síťová vrstva (směrování)*

1. datagramová služba, virtuální kanály
   a) vlastnosti
   b) výhody a nevýhody (porovnání)
   c) propojovací tabulky
2. směrovací metody
   a) vlastnosti
   b) porovnání
3. distance vector algoritmy
4. link state algoritmy
5. RIP
6. OSPF

### *5. přednáška = Propojování sítí (huby, bridge, routery)*

1. opakovač, přepínač, směrovač, brána - vlastnosti, funkce, rozdíly
2. přepínač
   a) učení přepínacích tabulek
   b) režimy provozu
      - cut-through
      - store-and-forward
   c) STA
      - chování s/bez STA
      - algoritmus STA

### *6. přednáška = Protokoly transportní vrstvy. Protokolová rodina TCP/IP v 4*

1. zběžná znalost hlaviček všech uváděných protokolů
2. IP
   a) k čemu se používá
   b) fragmentace
3. ICMP
   a) použití
   b) znalost některých „služeb“ např. echo, redirect …
4. RARP, BOOTP, DHCP
   a) použití
   b) základní rozdíly

5. UDP
   a) použití (i příklady aplikací)
   b) vlastnosti
   c) porty
   d) programování!!!
6. TCP
   a) použití (i příklady aplikací)
   b) vlastnosti
   c) porty
   d) TCP spojení
      - navázání
      - komunikace
      - ukončení
      - rámcová znalost řízení toku
   e) programování!!!
   f) rozdíly oproti UDP, kdy použijete UDP, kdy TCP

### *7. přednáška = IPv6 (vlastnosti, adresace, bezpečnost, mobilita)*

IPv6
1. formát diagramu
2. řetězení hlaviček
3. adresace
4. fragmentace
5. automatická konfigurace
6. mobilita
7. rozdíly oproti IPv4

### *8. přednáška = Řízení toku. QoS. Protokol SCTP*

Řřízení toku na síťové a transportní vrstvě
1. použití
2. způsoby
3. plánovací mechanismy
   - FIFO, Round Robin …
4. řízení toku v protokolu TCP

### *9. přednáška = Adresářové služby (DNS, X.500)*

1. jmenné služby - použití
2. LDAP
   a) co je to
   b) typy dat
   c) aplikace
3. DNS
   a) organizace dat v DNS
      - typy dat
      - distribuce
      - domény
   b) typy serverů
   c) rekurzivní a nerekurzivní dotaz
   d) reverzní dotaz

### *10. přednáška = Bezpečnost (principy, symetrické a asymetrické šifry, digitální podpis)*

1. vysvětlení pojmů
   a) šifrování
   b) autentizace
   c) podpis
   d) integrita
2. symetrické šifrování a šifrování s veřejným klíčem
   a) základní popis
   b) rozdíly
   c) typické použití
3. autentizace
   a) způsoby
   b) rozdíly

4. podpisování
5. distribuce klíčů
6. certifikát
   a) rámcový obsah
   b) použití

### *11. přednáška = Zabezpečení sítě (pravidla, firewally, NAT, ssh, ssl, ipsec, vpn)*

1. firewall
   a) popis
   b) zapojení (jen filtr, s DMZ …)
   c) pravidla
2. vzdálené připojování
   a) použití
      - k čemu to je?
      - požadavky
   b) protokoly
   c) aplikace

### *12. přednáška = Speciální sítě (FibreChannel, NAS, SAN)*

Způsoby připojení datových zařízení (DAS, NAS, SAN)

1. vlastnosti
2. porovnání
3. používané protokoly

### *13. přednáška = Správa sítí (SNMP, CMIP, RMON, aplikace pro dohled sítí)*

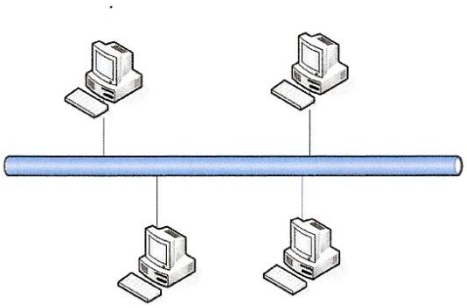
1. oblasti síťové správy
2. architektura a součásti síťové správy (server, agent, protokol)
3. SNMP
   a) volání (get, set …)
   b) verze
   c) MIB
   d) jaké proměnné může nabízet
4. RMON
   a) popis
   b) co nabízí

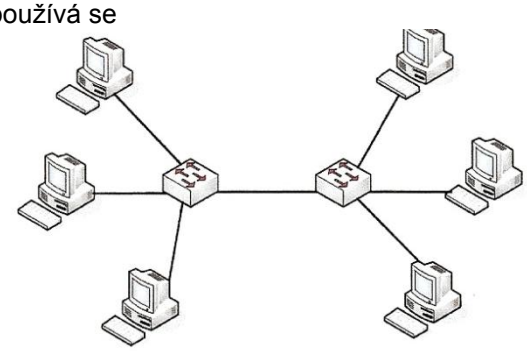
## *1. přednáška = síťové vrstvy*

**Taxonomie sítí**
- důvod použití (průmyslové aplikace, informační systémy apod.)
- rozlehlost = LAN (lokální), MAN (metropolitní), WAN (wave)
- rychlost
- topologie

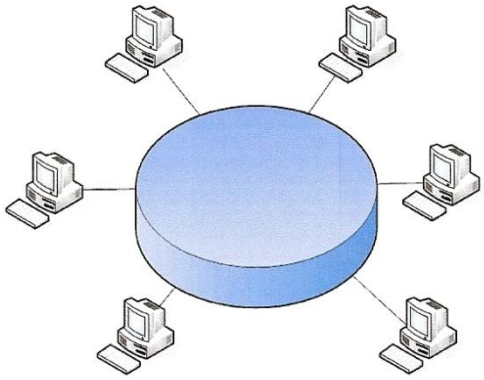
**1. Jaké jsou vlastnosti jednotlivých topologií sítí (sběrnice, hvězda, kruh, …)?**

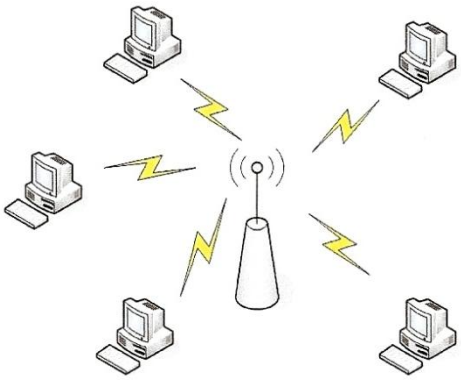
- **sběrnicová** - pasivní médium, médium je koaxiální kabel nebo kroucený dvoudrát
  - snadné připojení stanic (jedna sběrnice např. ethernet. kabel, na které jsou připojeny stanice)
  - odolné vůči výpadkům stanic
  - citlivost na poruchu média
  - data v síti jsou posílána všem počítačům
  - v daný okamžik může zprávy odesílat pouze jeden počítač
  - dnes se toto zapojeni téměř nepoužívá nicméně používá se





- **hvězda/strom** - stanice připojeny k centrálnímu uzlu samostatnými linkami
  - hub buď pasivní nebo aktivní
  - odolné vůči výpadku stanic i linek, citlivost na výpadek hubu
  - může komunikovat více stanice
  - snadná rozšiřitelnost





- **kruh** - dvoubodové jednosměrné spoje (lze tedy kombinovat média)
  - linky jsou zapojeny do uzavřeného kruhu (POZOR na záměnu s hvězdou!!!)
  - citlivost na poruchu spoje i stanice
  - u všech stanic je stejné zpoždění, protože stanice vysílají/přijímají jedna po druhé
- **bezdrát** - sdílené médium, snadná výstavba, citlivost na vnější rušení

**2. Jaké jsou vrstvy v OSI modelu?**

- Aplikační
- Prezentační
- Relační
- Transportní
- Síťová
- Linková (Spojová)
- Fyzická

3. **Jaké jsou vrstvy v TCP/IP modelu?** (Méně používané.)
   - Application (HTTP, FTP, SMTP, DNS, TFTP,...)
   - Transport (TCP, UDP)
   - Internet (IP)
   - Network Access

4. **Jaké má vlastnosti/Co zajišťuje (fyzická, linková, první, druhá, …) vrstva OSI modelu?**
   - ***Fyzická*** – specifikuje fyzickou komunikaci. Hlavní funkce poskytované fyzickou vrstvou jsou:
     - umožňuje přenos bitů kanálem.
     - definuje, co je 0 a co 1,
     - předepisuje vlastnost média (jaký drát bude použit, zapojení konektorů apod.),
     - definuje elektrické a mechanické vlastnosti rozhraní,
     - např. Ethernet 10BaseT, RS232,…
   - ***Linková*** (neboli spojová) - poskytuje spojení mezi dvěma sousedními systémy.
     - seřazuje přenášené rámce, stará se o nastavení parametrů přenosu linky,
     - zajišťuje funkci spolehlivého spojení (detekce a korekce chyb = oznamuje neopravitelné chyby),
     - rozpoznává rámce, formátuje fyzické rámce, opatřuje je fyzickou adresou,
     - poskytuje řízení toku na lince,
     - poskytuje jednoznačnou adresu v rámci segmentu (linkovou adresu),
     - např. PPP protokol, LLC 802.2,…

     Na této vrstvě pracují veškeré mosty a přepínače. Poskytuje propojení pouze mezi místně připojenými zařízeními a tak vytváří doménu na druhé vrstvě pro směrové a všesměrové vysílání.
   - ***Síťová*** vrstva – stará se o směrování v síti a síťové adresování. Poskytuje spojení mezi systémy, které spolu přímo nesousedí. Obsahuje funkce, které umožňují překlenout rozdílné vlastnosti technologií v přenosových sítích. Síťová vrstva poskytuje
     - adresaci a směrování dat přes mezilehlé prvky,
     - jednoznačnou adresu v rámci sítě (síťovou adresu),
     - síťovou službu se spojením,
     - síťovou službu bez spojení,
     - reportuje o problémech při doručování dat,
     - např. X.25, IP (Internetový protokol),…

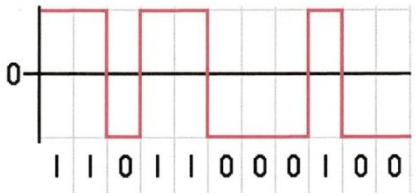
     Veškeré směrovače pracují na této vrstvě a posílají data do jiných sítí. Zde se již pracuje s hierarchickou strukturou adres.
   - ***Transportní*** (přenosová) vrstva – poskyzuje
     - rozklad dat na pakety,
     - uspořádání dat podle pořadí, tj. srovná pakety zpět do celku když dojdou zpřeházeně
     - multiplexuje a demultiplexuje data mezi transportními spoji, tj. adresace aplikace
     - transportní adresy (adresa, port),
     - koncové řízení toku,
     - hlavními protokoly této vrstvy jsou TCP a UDP.
   - ***Relační*** vrstva (session layer) – jejím úkolem je organizovat a synchronizovat dialog mezi spolupracujícími relačními vrstvami obou systémů a řídit výměnu dat mezi nimi. Umožňuje
     - vytvoření a ukončení relačního spojení,
     - synchronizaci a obnovení spojení,
     - oznamovaní výjimečných stavů
     - vytváření rozhraní pro aplikace a synchronizace spojení pomocí transakcí,
     - např. RPC, sdílení disků,…
   - ***Prezentační*** – transformace dat do tvaru použitelného aplikacím, vrstva se zabývá strukturou dat. Poskytuje
     - sjednocení prezentace informace,
     - dohodu o syntaxi,
     - transformaci dat
     - šifrování,
     - komprese,
     - např. kódování (ASCII/EBCDIC), big a little Indian, XDR, ASN.1,…
   - ***Aplikační*** – poskytuje aplikacím přístup k nižším vrstvám. Poskytuje
     - podůrné funkce aplikacím ASE (Application Service Element)
       - SASE = specifická podpora = přenos souborů, pošta, terminály
       - CASE = univerzální podpora = vytváření aplikačního spojení, obsluha transakcí
     - např. knihovny pro tvorbu síťových aplikací

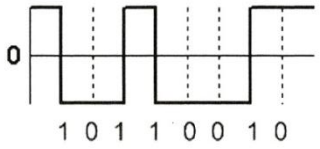
*2. přednáška = Fyzická vrstva*

**Multiplex** – frekvenční (FDMA) a časový (TDMA). Důležitý ukazatel pro určení kapacity přenosového kanálu.

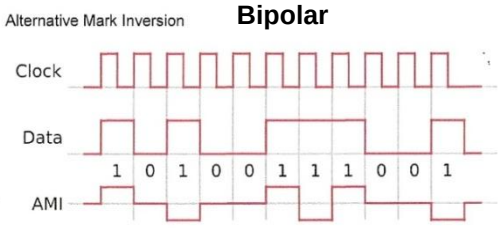
**Kapacita přenosového kanálu** – vychází z multiplexu. Jakýkoli signál lze popsat sadou sinusovek.

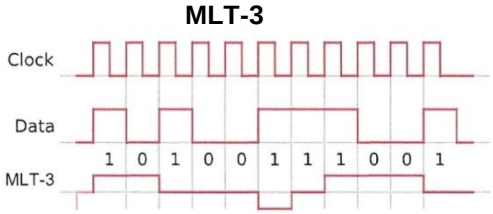
1. **Významné vlastnosti kódování.**
   - stejnosměrná složka – ideální je, aby byla 0, jinak signál „ujíždí“ do +/-
   - časová synchronizace – u některých typů kódování jsou hodiny vedeny samostatným drátem
   - šířka pásma
   - přenosová rychlost
2. **Typy kódů a jejich vlastnosti.**
   Non Return, Bipolar a MLT-3 jsou typy kódování, které nemají stejnosměrnou složku. U stejných hodnot se tak narušuje časová synchronizace. Stejných hodnot by mělo být max. 10 za sebou. „Hodiny“ je nutno přenášet samostatným drátem.



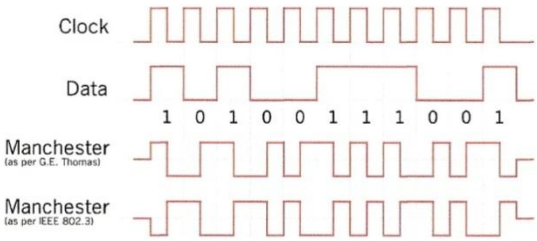

   - **Non Return to Zero (Unipolar)**
     - 1 a 0 se vyjadřují pomocí dvou předem definovaných napěťových úrovní,
   - **Non Return to Zero Inverted**
     - 1 je změna stavu, 0 je beze změny stavu
     - plus – pokud bude dost 1, bude zaručena časová synchronizace a stejnosměrná složka
     - 0 je „problém“, zde je nutno pomoct si kódováním a vkládat 1



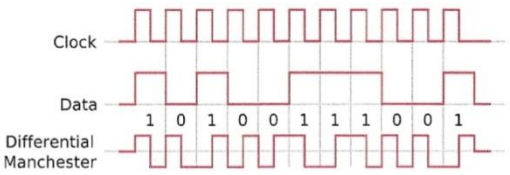




   - **Bipolar Non Return to Zero**
     - 1 je vždy plus
   - **Bipolar encoding**
     - 0 je vyjádřena shodně jako v unipolárním kódování
     - 1 je potom vyjádřena jako střídavě kladná a střídavě záporná napěťová úroveň
   - **MLT-3**
     - MLT-3 cykluje neustále mezi stavy +1, 0, -1, 0, kdy 0 nechává aktuální stav a 1 překlápí do stavu -1/+1
   - **Manchester**
     - 1 = jedna hrana, 0 = druhá hrana – existují dvě protichůdné normy, co je 1 a co 0, takže nelze říci jednoznačně – IEEE 802.3 – Ethernet a IEEE 802.4 – Token Bus
     - stejnosměrná složka = 0
     - má automatickou synchronizaci (z jednoho signálu jdou číst data i hodiny)
     - každý bit se přenáší stejně dlouhý časový úsek, tzn. drží stabilní frekvenci
     - případná hrana na začátku časového úseku se ignoruje
     - nevýhoda – má dvojnásobnou frekvenci oproti vstupu, proto vyžaduje pro přenos dvojnásobnou šířku pásma
     - používá v pomalejších ethernetech
   - **Diferenciální manchester**
     - 1 = otáčí hranu, 0 = drží stav
     - ALE pokud je hrana i na začátku daného časového úseku, znamená to nulu, absence pak znamená jedničku
     - má dvojnásobnou frekvenci, tzn. je nutný dvojnásobná šířka kanálu – tady to dělají nuly

**Manchester**




**Diferenciální manchester**




- **4B5B** - kódování 4B5B mění sekvekci 4b na sekvenci 5b tak, že v nové sekvenci jsou vždy alespoň dvě jedničky. To zaručuje časovou synchronizaci tam, kde se signál synchronizuje pouze v jedničce – na hraně (bipolar, MLT-3)

## 3. Metalická vedení a jejich vlastnosti

**a) symetrická a asymetrická vedení,**

- ***symetrická***
  - založeno na krouceném dvoudrátu – POZOR, kroucení je nutné kvůli přeslechům. Pokud je kabel kroucený, pak jsou siločáry rušeny shodně na obou drátech,
  - nízká cena
  - na vzdálenost 100 m, rychlost do cca 150 Mb/s,
  - signál odvozen z napětí obou kabelů 1 pár pro přenos tam, druhý pár pro přenos zpět
  - využívá se hlavně u lokálních sítí – líp se s tím pracuje
  - norma pro zapojení konektorů 1, 2 / 3, 6 (pozice 4, 5 je nepárová!!!)
- ***asymetrická*** = koaxiální kabely
  - 1 - 50 Mb/s: stovky metrů,
  - 40 Mb/s: kilometry (kabelová televize)
  - jeden kabel stíní, signál odvozen z druhého kabelu proti zemi

**b) typy symetrických vedení (UTP, STP, FTP)**

UTP - nestíněný, 4 páry
FTP - jako UTP, ale celé stíněné fólií
STP - stíněný, např: 2 dvoudráty samostatně stíněné

**c) orientační znalost tříd UTP kabelů** – různé kategorie = různá kvalita, dnes kabely od Cat5

Cat3: 16 MHz, 10 Mb/s
Cat4: 20 MHz, 16 Mb/s
Cat5: 100 MHz, 100Mb/s
Cat5e: 100 MHz, 1 Gb/s
Cat6: 250 MHz, 10 Gbit/s
Cat6a: 500 MHz, 10Gb/s (2/2008)
Cat7: 600 MHz, 100 Gb/s na vzdálenosti kolem 70 metrů (aktuální hodnota Listopad 2007)

**d) přeslechy (NEXT, FEXT, útlum a přeslechy)**

**Přeslech** = představuje nezáměrné prostupování elektrického signálu z jednoho telekomunikačního kanálu do druhého. V symetrických liniích přeslech vzniká, neboť každý vodič nesoucí elektromagnetický signál přenáší proud do přilehlého konduktoru. Tento zvukový signál je vždy měřitelný na obou koncích dané linky.

***2 typy přeslechů***:

- NEXT = near-end crosstalk = chyba vznikající na bližším konci vodiče od zdroje signálu
- FEXT = far-end crosstalk = chyba, která se projevuje na konci vzdálenějším

Silné přeslechy vedou k falešným kolizím na NEXT, na FEXT vedou k poškození signálu.

**Útlum** = znamená pokles amplitudy signálu způsobenou ztrátami na vedení.

## 4. Optická vedení

**a) konstrukce vlákna**

- jádro a plášť – jsou z opticky vodivých materiálů (sklo-sklo, plast-plast, apod., NE zrcadlo)
- primární ochrana
- tahové prvky
- sekundární ochrana
- pro komunikaci jsou zpravidla 2 vláka – jedno tam, jedno zpět.

**b) maximální úhel navázání a numerická apertura**

- úhel vstupujícího paprsku nesmí přesáhnout maximální úhel navázání, jinak se v jádře nebude odrážet a ztratí se
- numerická apertura - míra schopnosti vlákna navázat paprsek do jádra, závisí na okolí

**c) jednovidová a vícevidová vlákna**

- jednovidová (Singlemode) - vykazují nejlepší parametry optické přenosové cesty. Mají nejmenší průměr jádra do 10 mikro-metrů. Takto malé jádro má za následek velký úhel odrazu ve vlákně, to vede k menšímu prodloužení dráhy paprsku. Mají jen chromatickou disperzi. Plus – prochází jen jeden paprsek.
- vícevidová (Multimode) - zásadní rozdíl oproti jednovidovým vláknům je v průměru jádra (ale 1000x větší opravdu není!). S velkým jádrem se zvětšuje i počet drah po nichž paprsky procházejí. Proto se vlákna nazývají mnohavidová (vícevidová).

**d) vidová a chromatická disperze a její význam**

- Disperze je příčinou zkreslení přenášeného signálu, dochází ke zpožďování impulsů a změně jejich tvaru.
- vidová disperze - různé vidy mají různé rychlosti šíření signálu vláknem
- chromatická disperze - spektrální složky téhož vidu se šíří různou rychlostí

***3. přednáška = Linková vrstva (protokoly, potvrzování, přístupové metody)***

**1. Kódová vzdálenost a schopnost detekce a oprav chyb.**

- Hammingova vzdálenost dvou slov je dána počtem odlišných bitů těchto slov a značí se vzd(a, b). Platí, že vzd(a, b) = váha(a xor b), kde váha je počet jedniček ve slově. Hammingova vzdálenost kódu se označuje jako kódová vzdálenost kvzd – minimální vzdálenost dvou slov v kódu. Pokud máme kód s kódovou vzdáleností kvzd, můžeme provádět detekci dch < kvzd bitů nebo detekci a korekci dch + och < kvzd bitů.
- $kvzd = ch_d + ch_o + 1; ch_d \geq ch_o; ch_d, ch_o \in \mathbb{N}$
- Pokud máme kód s kódovou vzdáleností kvzd, můžeme provádět detekci dch < kvzd bitů nebo detekci a korekci dch + och < kvzd bitů.

**2. Potvrzovací schémata - popis a vlastnosti.**

- **Synchronní simplexní protokol**
  - synchronní - nelze pozastavit, simplexní - provoz jedním směrem
  - bez zpětného potvrzení - nereaguje na chyby
  - vyžaduje použití samoopravných kódů
- **Simplexní protokol s pozitivním potvrzováním**
  - nutný alespoň poloduplexní(vždy jenom jedním směrem, ale může se střídat) kanál, efektivní pro kanály s malou chybovostí
  - vysílač pošle zprávu, čeká (do timeoutu) na potvrzení - nepřijde-li, zopakuje zprávu
  - přijímač pošle potvrzení v případě bezchybného příjmu
- **Simplexní protokol s čistě negativním potvrzováním**
  - pošle NACK pokud dorazila chybná data
  - rychlá reakce na chybu, ale
  - nedokáže reagovat na ztracený rámec i ztracené potvrzení (odmítnutí)
  - vysílač pošle zprávu a čeká na odmítnutí; nepřijde-li do timeoutu, považuje zprávu za doručenou
  - přijímač pošle odmítnutí při chybě
- **Simplexní protokol s negativním potvrzováním**
  - pošle ACK pokud je vše OK, NACK pokud zpráva je poškozena
  - možná duplikace
  - možná záměna ACK/NACK

**3. Přístupové metody**

Popisují pravidla, kterými je řízen přístup pracovních stanic na vedení. Těmito metodami je vlastně nadefinován, jak zabezpečit, aby v jednom okamžiku vysílala jenom jedna stanice. Kdyby v jednom okamžiku vysílalo několik stanic docházelo by ke vzájemnému rušení.
**Rozlehlost sítě** – důležitý faktor pro přístupové metody v linkové vrstvě.

**a) rozdíl mezi náhodným přístupem a deterministickými metodami,**

- nejrozšířenější **DETERMINISTICKÉ** metody přístupu stanice k síti jsou založeny na principu předávání tzv. řídícího znamení. Toto řídící znamení je nazýváno TOKEN a přístupová metoda k médiu se nazývá TOKEN PASSING. Stanice pracující na principu této metody si navzájem předávají krátkou zprávu. Obdržením této zprávy TOKEN získává stanice právo vysílání této zprávy, pokud nějakou k vysílání má. Toto vysílání musí stanice ukončit do předem stanoveného časového limitu a na závěr vyslat TOKEN další stanici.
- z **NEDETERMINISTICKÝCH** metod je nejrozšířenější CSMA/CD. U této metody se přístup ke sdílenému médiu řídí dvěma pravidly. Stanice smí vysílat zprávu pouze tehdy, pokud nevysílá žádná jiná stanice. Jestliže stanice zjistí, že její vysílání je rušeno, okamžitě přeruší vysílání a o nové vysílání se pokusí až po uplynutí stanoveného časového limitu. Pro odstranění kolizí je tento limit volen **náhodně**. Základ této doby je dán generátorem náhodných čísel. Opakování se provádí zhruba 16x. V

systémech používaných pro administrativu se používají obě metody. V systému řízení technologických procesů je výhodnější používat metody deterministické.

**b) CSMA metody.**

- **NENALÉHAJÍCÍ** - pred odesláním paketu testuje stav kanálu. Je-li kanál volný, stanice zahájí vysílání. Pokud je kanál obsazen, stanice pocká náhodne zvolený casový okamžik a znovu testuje stav kanálu. Postup opakuje do odeslání paketu.
- **NALÉHAJÍCÍ** - pred odesláním paketu testuje stav kanálu. Je-li kanál obsazen, stanice odloží vysílání na okamžik jeho uvolnení.
- **P-NALÉHAJÍCÍ** - pred odesláním paketu testuje stav kanálu. Je-li kanál volný, stanice zahájí vysílání. Pokud je kanál obsazen, stanice pocká na uvolnení kanálu. Byl-li kanál volný nebo se práve uvolnil, zacne stanice s pravdepodobností p vysílat a s pravdepodobností q = 1 - p odloží další cinnost o krátký casový interval (muže odpovídat délce šírení signálu médiem). Po uplynutí této doby celou cinnost opakuje až do úspešného odeslání paketu.
- **CSMA/CD** = Carrier Sense Multiple Access with Collision Detection = metóda "odpočúvania" siete - Karta najprv načúva či je premávka na prenosovom médiu a ak nezachytí nosnú frekvenciu "carrier" tak kábel je voľný a môže vysielať. Môže sa však stať, že toto urobia dve sieťové karty súčasne a tým nastane kolízia keďže začnu spoločne vysielať a tým sa rušiť. Tento stav "kolíziu" musia zaregistrovať aj ostatné zariadenia a preto zastavia vysielanie až po chvíľke. Po zastavení vysielania počkajú nejaký náhodný okamih a potom začnú celý tento proces odznova.Túto metódu využíva štandard IEEE 802.3 (technológia Ethernet)
- **CSMA/CA** = Carrier Sense Multiple Access with Collision Avoidance = modifikácia metódy CSMA/CD - nedetekuje sa kolízia. princím spočíva v odhade kedy asi môže nastať kolízia a vtedy nevysielajú, týmto sa odstráni problém kolízie, ale výrazne sa spomalí premávka na sieti. Túto metódu využíva napr. štandard IEEE 802.11

**4. Protokoly linkové vrstvy - především Ethernet II, IEEE 802.3 a IEEE 802.2**

- rozdíl ve formátu linkového rámce na úrovni podvrstvy MAC (Media Access Control)
- zde se Ethernet II a IEEE 802.3 liší ve významu dvou konkrétních bytů v hlavičce rámce: verze Ethernet II v těchto dvou bytech očekává údaj identifikující datový obsah rámce (neboli jeho typ), verze IEEE 802.3 má na stejném místě dvoubytový údaj o délce celého rámce
- naštěstí je ale vždy možné korektně rozpoznat obě verze, protože největší ethernetový paket má délku 1500 bytů, zatímco všechny identifikátory obsahu rámce (u Ethernetu II) jsou číselné konstanty větší než 1500. Praktickým důsledkem pak je skutečnost, že oba typy ethernetových rámců mohou vedle sebe koexistovat "na jednom drátě"
- významným rozdílem mezi Ethernetem II a "Ethernetem" IEEE 802.3 je to, že veškerý další vývoj se odehrává již jen na půdě společnosti IEEE a jejích pracovních skupin. Větev Ethernetu II se od roku 1982 již dále nevyvíjí, a veškeré nové verze "Ethernetu" tedy patří do vývojové větve IEEE 802.3.
- IEEE 802.2 je starší protokol, který v hlavičce obsahuje údaje jako přístupový bod, identifikaci organizace nebo protokol

***4. přednáška = Síťová vrstva (směrování)***

---

**1. datagramová služba, virtuální kanály**

***a) vlastnosti***

- **DATAGRAMOVÁ SLUŽBA** - každou jednotlivou zprávu (paket) označit adresou cílového účastníka a předávat ji polygonální sítí nezávisle na předávání ostatních zpráv (paketů),
- **VIRTUÁLNÍ KANÁL** - při otevírání spojení koncových účastníku uložit v paměti přepojovacích uzlů poznámku o zvolené cestě, a pouze tuto cestu pak využívat pro další komunikaci.

***b) výhody a nevýhody (porovnání)***

- **DATAGRAMOVÁ SLUŽBA**
  - + vhodná na sítích, kde se počítá s možnými výpadky prvků
  - - nezachovává pořadí paketů - nutno pořešit v transportní vrstvě
  - - výpadek prvku může způsobit ztrátu paketů
- **VIRTUÁLNÍ KANÁL**
  - + zachovává pořadí paketů
  - + ztráta paketů se snadno detekuje
  - - složitější a náchylná na výpadek
  - - po uzavření spojení nutno mazat údaje u propojovacích tabulek

***c) propojovací tabulky***

- Linka zabezpečená procedurou řízení je při výstavbě virtuálních kanálu využita vícenásobně. Uzel identifikuje příslušnost přenášeného paketu ke konkrétnímu virtuálnímu kanálu na lince podle logického čísla kanálu v záhlaví paketu. Logická čísla jsou kanálům přidělována při jejich otevírání. Uzel po příchodu speciálního řídícího paketu, který odpovídá žádosti o otevření kanálu, vybere vhodný

neobsazený výstupní kanál ve směru určeném směrovací strategií, a logická čísla příchozího i odchozího virtuálního kanálu si poznamená do přepojovací tabulky. Pro další pakety přicházející s daným logickým číslem kanálu pak jednoduše podle přepojovací tabulky zamění logické číslo kanálu a paket předá k odeslání do výstupní linky. Strategie FIFO ve frontě této linky zajistí, že celá síť zachovává pořadí paketů.

2. **směrovací metody** - popisují chování paketu v určitém bodě sítě (kam půjde paket dál?)
   - **záplavové** - každý uzel kromě příjemce vyšle přijatý paket do všech směrů
     - + nejkratší cesta
     - + spolehlivost
     - - zahlcení sítě
     - - potřeba likvidovat nadbytečné pakety
   - **náhodné** - odesílání paketu na náhodný výstup
     - + odolnost proti změně topologie
     - + při částečné znalosti topologie lze její chování upravit, pak už je užitečná
     - - nezaručuje omezenou dobu doručení
     - - potřeba dodatečné informace
   - **izolované** - bere v úvahu pouze lokální informace a už ne informace ostatních uzlů sítě
     - horký brambor
     - odeslání paketu na výstup s nejkratší frontou
     - nezaručuje omezenou dobu doručení
     - potřeba dodatečné informace
     - zpětné učení
     - využití informací o čase / počtu průchodů v paketu, tabulka obsahuje odesílatele, nejkratší čas, směr odkud paket přišel
     - nutné kombinovat s jinou metodou
     - pomalá konvergence při chybě - potřeba zapomínání
   - **adaptivní** - směrovací tabulky se přizpůsobují momentálnímu stavu
     - + bezpečnost
     - + reakce na změnu
     - možnost kombinace s předchozími metodami
     - link-state (OSPF)
     - distance-vector (RIP)
   - **statické** - nastavení tabulek při návrhu sítě, vhodné určit i alternativní směry
   - **hierarchické** - rozdělení adresy: prefix oblasti + adresa uzlu uvnitř
     - adresování respektuje topologii

**Distance vector algoritmy - výměna kompletních směrovacích tabulek**
- - vzdálenost měří pouze pomocí předem definovaných hodnot, nereagují na aktuální změny zatížení
- - routery si vyměnují kompletní směrovací tabulky -> větší zátěž sítě
- - pomalá konvergence - obzvlášť při výpadku
- Ford-Fulkersonův algoritmus
- Příklady: RIP, IGRP, EIGRP, BGP

**Link state algoritmy - výměna změn sítě -> každý uzel má informace o síti**
- + rychlá konvergence
- + nízké zatížení sítě
- příklad: OSPF

**RIP - Routing Information Protocol**
- komunikace se sousedy - uzly si v pravidelných intervalech vyměňují informace o vzdálenostech (odhadech) k dalším uzlům
- pouze jedna cesta
- techniky zrychlení konvergence (tzn. adaptace na vyřazení nějakého prvku ze sítě)
  - Split horizon - uzel nepředává nové informace zpět uzlu, od kterého je získal
  - Poison reverse - uzel nepředává nové informace zpět uzlu, od kterého je získal a místo toho mu podstrčí hodnotu nekonečna (16)

**OSPF - Open Shortest Path First**
- OSPF je typickým představitelem směrovacího protokolu typu Link State. Vytváří tedy v paměti směrovače kompletní mapu celé sítě, označovanou jako topologická databáze (někdy se jí říká Link State Database). Nad touto databází potom pomocí algoritmu označovaného jako Shortest Path First (SPF) provádí výpočty potřebné k nalezení nejvýhodnější cesty do jednotlivých sítí.

- Uplatňuje se rozdělení na oblasti, 0 je páteřní oblast. Komunikace mezi dvěma autonomními systémy musí vždy jít přes páteř

## *5. přednáška = Propojování sítí (huby, bridge, routery)*

1. **opakovač (repeater, hub)**
   - elektronický pasivní síťový prvek
   - přijímá zkreslený, zašuměný nebo jinak poškozený signál - vysílá ho dále opravený, zesílený a správně časovaný
   - -> zvýšení dosahu média bez ztráty kvality a obsahu signálu
   - patří do první (fyzické) vrstvy referenčního modelu OSI (pracují přímo s elektrickým signálem)
   - nemá žádnou vyrovnávací paměť
   - může propojovat libovolný počet segmentů sítě, ale pouze segmenty se stejnou přenosovou rychlostí
   - opakovač musí šířit kolize (!)
   - v Ethernetu jich nemůže být libovolně mnoho kvůli době šíření kolize (CSMA/CD) - mezi každými 2 body maximálně 2 opakovače
2. **přepínač (switch)**
   - aktivní síťový prvek
   - propojuje jednotlivé segmenty sítě
   - obsahuje větší či menší množství portů (až několik stovek), na něž se připojují síťová zařízení nebo části sítě
   - analyzuje procházející pakety a podle informací v nich obsažených (adres, identifikátorů apod.) rozhoduje, kam paket předat dál
   - nepropaguje kolize (má paměť)
   - všesměrově šíří pouze broadcasty
   - často se používá jako náhrada opakovače
   - přes jeden přepínač může probíhat více přenosů až do maximální kapacity přepínače

   ***a) učení přepínacích tabulek***
   - aby mohl přepínač (most) filtrovat, musí znát topologii sítě (na jakém portu je která stanice)
   - pokud neví, chová se podobně jako opakovač
   - můžeme nastavit ručně (pracné, síť se může často měnit - notebooky atd.)
   - automatické učení
     - nejdříve se chová jako opakovač a učí se
     - ukládá si do tabulky dvojice [port, adresa]
     - po určité době začne filtrovat provoz

   ***b) režimy provozu***
   - cut-through
     - paket odchází ihned po přečtení cílové adresy (měl by se používát pouze v plně duplexních sítích s mikrosegmentací)
     - přečte pouze cílovou adresu a zbytek již rovnou přeposílá
   - store-and-forward
     - celý paket je uložen, spočtena checksum a poté je odeslán, používá se u sítí s velkým rušením a nebo při asymetrickém switchování
     - přepínač celý rámec přijme, analyzuje a potom odešle
   - Fragment free
     - switch začne přeposílat rámec až po přijetí 64bitů, kdy se ujistí, že na daném segmentu nevznikla kolize - ma význam v případě, kdy je do switche připojen Hub.

   ***c) STA - Spanning Tree Algorithm***
   - chování s/bez STA
   - algoritmus STA
     - algoritmus, pomocí kterého se najde kostra dané síťové topologie
     - přeruší se případné kružnice zablokováním některých portů - tím se zachová stromová struktura
     - v případě výpadku se zablokované porty opět uvolní
     - všechny moderní přepínače tento algoritmus podporují (IEEE 802.1d)
     - Simulace STA
     - Popis STA v češtině
     - Popis STA od CISCO s obrázky

3. **směrovač (router)**
   - přeposílá datagramy směrem k jejich cíli (routování - směrování)
   - pracuje na síťové vrstvě, propojuje sítě
   - pracuje se síťovými adresami
   - nešíří broadcasty z jedné sítě do druhé
   - přenáší data i mezi sítěmi, které používají naprosto odlišné linkové technologie, jsou tedy naprosto nezbytné pro Internet

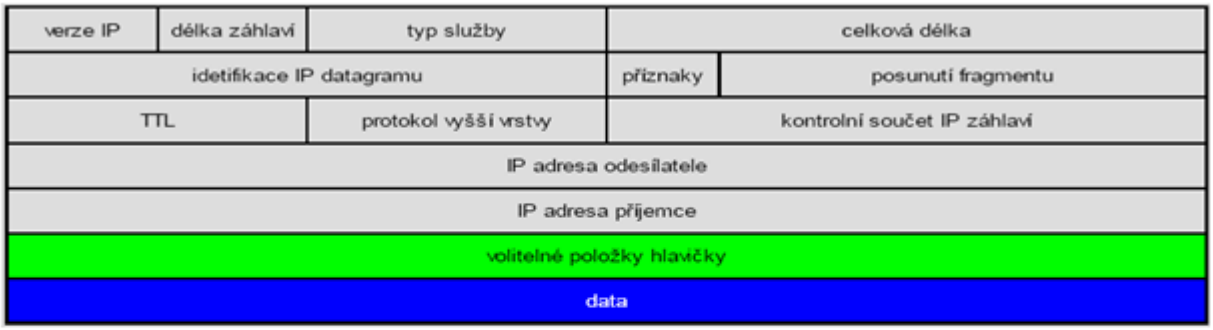
- kromě směrování může podporovat i další funkce (firewall, NAT, VPN)
- směrování je statické nebo dynamické (např. RIP, OSPF, ...)

**brána (gateway)**
- 3 typy - směrovač
  - aplikační brána
  - brána pro překlad protokolů z jedné množiny protokolů do jiné
- příklady aplikační brány - software pro poštovní aplikace, proxy WWW server apod.

## *6. přednáška = Protokoly transportní vrstvy. Protokolová rodina TCP/IP v 4*

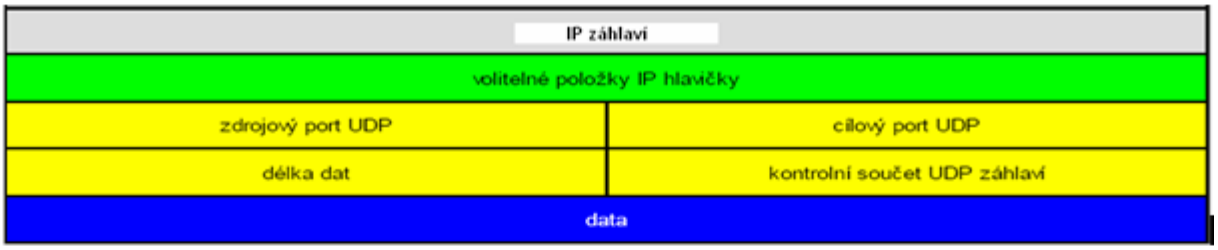
### 1. zběžná znalost hlaviček všech uváděných protokolů

IP Hlavička

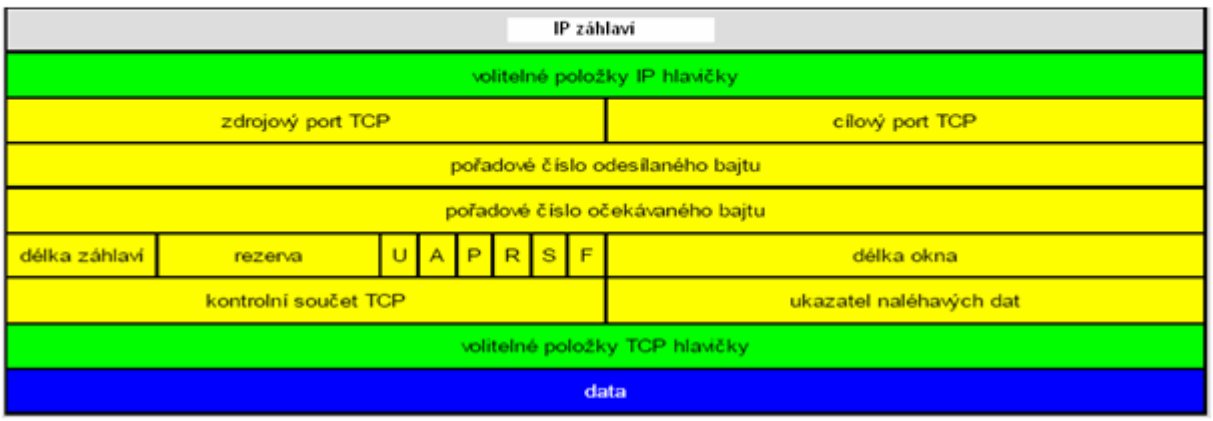



ICMP hlavička


UDP hlavička




TCP hlavička




- TCP:
  - pořadové číslo odesílaného bajtu – číslo prvního bajtu
  - pořadové číslo očekávaného bajtu – poslední dobře přijatý bajt + 1
  - příznaky: U urgentní data A potvrzení P aplikační data R odmítnutí spojení S nová sekvence číslování F ukončení spojení

### 2. IP

**a) k čemu se používá**
- Internet Protocol je základní protokol síťové vrstvy a celého Internetu
- provádí vysílání datagramů na základě síťových IP adres obsažených v jejich záhlaví
  - každý datagram je samostatná datová jednotka, která obsahuje všechny potřebné údaje o adresátovi i odesilateli a pořadovém čísle datagramu ve zprávě

        - datagramy putují sítí nezávisle na sobě a pořadí jejich doručení nemusí odpovídat pořadí ve zprávě
        - doručení datagramu není zaručeno, spolehlivost musí zajistit vyšší vrstvy (TCP, aplikace)
    - poskytuje vyšším vrstvám síťovou službu bez spojení
    - funkce a činnosti vykonávané protokolem IP:
        - adresace koncových uzlů a sítí v IP intersíti
        - vytváření IP paketů z paketů protokolů vyšší vrstvy
        - směrování IP paketů přes IP intersíť
        - fragmentace IP paketů
    - dále se stará o segmentaci a znovusestavení datagramů do a z rámců podle protokolu nižší vrstvy (např. Ethernet)
    - protokol IP je základním přenosovým prostředkem pro protokoly TCP/IP

**b) fragmentace**

- umožňuje vložení IP paketu do kratších rámců nižší vrstvy (MTU - Maximum Transmission Unit)
- fragmentaci provádí libovolný směrovač
- defragmentaci provádí koncový uzel
- možnost zakázání fragmentace

**Internet Protocol version 4** (**IPv4**) je v informatice čtvrtá revize IP (Internet Protocol) a zároveň jeho první verze, která se masivně rozšířila. Spolu s IPv6 vytvářejí základ pro komunikaci v rámci sítě Internet. IPv4 je popsána IETF v RFC 791 (září 1981), které nahradilo RFC 760 (leden 1980) a je standardizována jako MIL-STD-1777 Ministerstvem obrany USA.
IPv4 je datově orientovaný protokol, který je používán v sítích s přepojováním paketů (např. Ethernet). Jde o protokol přepravující data bez záruky, tj. negarantuje ani doručení ani zachování pořadí ani vyloučení duplicit. Zajištění těchto záruk je ponecháno na vyšší vrstvě, kterou v představuje protokol TCP. Stejně tak je na vyšší vrstvě ponechána kontrola integrity dat, protože IPv4 datagram nese pouze informaci o kontrolním součtu hlavičky datagramu se služebními údaji.

**Formát IPv4 datagramu**

**Formát IP datagramu**

| Bajty | 0 | | 1 | 2 | 3 |
|---|---|---|---|---|---|
| Bajt 0 až 3 | Verze | Délka hl. | Typ služby | Celková délka | |
| Bajt 4 až 7 | Identifikace | | | Příznaky | Offset fragmentu |
| Bajt 8 až 11 | TTL | | Protokol | Kontrolní součet hlavičky | |
| Bajt 12 až 15 | Adresa odesílatele | | | | |
| Bajt 16 až 19 | Adresa cíle | | | | |
| Bajt 20 až 23 | Volby | | | | Výplň |
| ... | Data | | | | |

Datagram IPv4 obsahuje hlavičku se služebními údaji nutnými pro přepravu a za ní následují data. Konec hlavičky je zarovnán na násobek čtveřice bajtů pomocí výplně (anglicky *padding*). Strukturu IP datagramu vystihuje tabulka uvedená v pravé části. Následuje popis jednotlivých polí:

- **Verze**: verze IP, zde 4.
- **Délka hl.**: délka hlavičky ve čtyřbajtových slovech; hlavička může být kvůli volbám různě dlouhá.
- **Typ služby (TOS, Type of Service)**: podle původních představ měla tato položka umožnit odesilateli, aby zvolil charakter přepravní služby ideální pro dotyčný datagram. Jednotlivé bity znamenaly např. požadavek na nejmenší zpoždění, největší šířku pásma či nejlevnější dopravu. Směrování pak mělo brát ohled na hodnotu TOS a volit z alternativních tras tu, která nejlépe odpovídala požadavkům datagramu. V praxi však k realizaci nedošlo. V současnosti se položka používá k podobným účelům – nese značku pro mechanismy zajišťující služby s definovanou kvalitou (QoS).
- **Celková délka**: délka datagramu v bajtech.
- **Identifikace**: odesilatel přidělí každému odeslanému paketu jednoznačný identifikátor. Pokud byl datagram při přepravě fragmentován, pozná se podle této položky, které fragmenty patří k sobě (mají stejný identifikátor).
- **Příznaky**: slouží pro řízení fragmentace. Definovány jsou dva: *More fragments* ve významu „nejsem poslední, za mnou následuje další fragment původního datagramu“ a *Don't fragment* zakazující tento datagram fragmentovat.

- **Offset fragmentu**: udává, na jaké pozici v původním datagramu začíná tento fragment. Jednotkou je osm bajtů.
- **TTL (Time To Live)**: představuje ochranu proti zacyklení. Každý směrovač zmenší tuto hodnotu o jedničku (případně o počet sekund, které datagram ve směrovači strávil, pokud zde čeká déle). Pokud tím TTL nabude hodnotu nula, datagram zahodí, protože vypršela jeho životnost.
- **Protokol**: určuje, kterému protokolu vyšší vrstvy se mají data předat při doručení. Čísla protokolů definována v RFC 1700 (TCP: 6, UDP: 17, ICMP: 1, EGP: 8, …). RFC 1700 je již překonáno novým standardem RFC 3232, jež odkazuje na databáze organizace IANA a její stránky: http://www.iana.org .
- **Kontrolní součet hlavičky**: slouží k ověření, zda nedošlo k poškození. Počítá se pouze z hlavičky a pokud nesouhlasí, datagram bude zahozen.
- **Adresa odesílatele**: IPv4 adresa síťového rozhraní, které datagram vyslalo.
- **Adresa cíle**: IP adresa síťového rozhraní, kterému je datagram určen.
- **Volby**: různé rozšiřující informace či požadavky. Například lze předepsat sérii adres, kterými má datagram projít. Volby obvykle nejsou v datagramu použity (v tabulce jsou barevně odlišeny).
- **Výplň**: nenese žádnou informaci, slouží k zaokrouhlení délky hlavičky na násobek čtyř bajtů, pokud jsou použity volby uvedené výše.
- **Data**: obsahuje přepravovaná data.

**ICMP - Internet Control Message Protocol**

**použití**

- slouží k přenosu řídících hlášení, která se týkají chybových stavů a zvláštních okolností při přenosu
- používá se např. v programu ping pro testování dostupnosti počítače, nebo programem traceroute pro sledování cesty paketů k jinému uzlu
- přenos chybových i řídících informací
  - testování dostupnosti (Echo Req/Rep)
  - řízení zahlcení a toku
  - změna směrovací tabulky
  - informace o masce
  - časová synchronizace
- omezená implementace
- zahazování z bezpečnostních důvodů

**znalost některých „služeb“ např. echo, redirect...**

- Echo Request … požadavek na odpověď, každý prvek v síti pracující na IP vrstvě by na tuto výzvu měl reagovat. Často to z různých důvodů není dodržováno.
- Echo Reply … odpověď na požadavek
- Destination Unreachable … informace o nedostupnosti cíle, obsahuje další upřesňující informaci
  - Net Unreachable … nedostupná cílová síť, reakce směrovače na požadavek komunikovat se sítí, do které nezná cestu
  - Host Unreachable … nedostupný cílový stroj
  - Protocol Unreachable … informace o nemožnosti použít vybraný protokol
  - Port Unreachable … informace o nemožnosti připojit se na vybraný port
- Redirect … přesměrování, používá se především pokud ze sítě vede k cíli lepší cesta než přes defaultní bránu. Stanice většinou nepoužívají směrovací protokoly a proto jsou informovány touto cestou. Funguje tak, že stanice pošle datagram své, většinou defaultní, bráně, ta jej přepošle správným směrem a zároveň informuje stanici o lepší cestě.
  - Redirect Datagram for the Network … informuje o přesměrování datagramů do celé sítě
  - Redirect Datagram for the Host … informuje o přesměrování datagramů pro jediný stroj
- Time Exceeded … vypršel časový limit
  - Time to Live exceeded in Transit … během přenosu došlo ke snížení TTL na 0 aniž byl datagram doručen
  - Fragment Reassembly Time Exceeded … nepodařilo se sestavit jednotlivé fragmenty v časovém limitu(např pokud dojde ke ztrátě části datagramů)

**RARP, BOOTP, DHCP**

použití

- Přidělení IP adresy

základní rozdíly

- RARP – Reverse Address Resolution Protocol, rfc 903
  - přidělení adresy bezdiskové stanici
  - nepoužívá se
- BOOTP – Bootstrap Protocol, rfc 951, rfc 2132
  - starší protokol
  - statické přidělení parametrů

- Protokol BootP slouží k získaní konfigurace. Klient ve svém požadavku (BOOTREQUEST) uvádí svou hardwarovou adresu a volitelně svou IP adresu, jméno serveru a požadovaný boot soubor. Server v odpovědi (BOOTREPLY) pošle požadovanou IP adresu, IP adresu gateway, svou IP adresu a jméno.
- DHCP – Dynamic Host Configuration Protocol, rfc 2131
  - více DHCP serverů
  - dynamické přidělení parametrů
  - Protokol DHCP rozšiřuje starší protokol BootP, zachovává stejný formát zpráv. Slouží také k získaní konfigurace, navíc má informace jako: maska podsítě, broadcastová adresa, adresa routeru, doména, adresa DNS serveru atd.

**UDP - User Datagram Protocol**

použití (i příklady aplikací)
- stream - proud - tok dat.
- asi nejlepším příkladem je vysílání internetových rádií a televizí, u těch je jedno jestli nějaký packet nepřijde, prostě se musí jet dál. Další věcí je rychlost.

vlastnosti
- rychlejší jak TCP
- neřeší kontrolu došlých packetů a ani pořadí ve kterém přicházejí, může se stát že z řady odeslaných packetů 1,2,3,4,5 vám dojdou ve stavu 4,2,5,3 s tím že 1 vůbec nedošla

porty
- UDP používá porty, aby bylo možné rozlišit v počítači jednotlivé aplikace a správně jim doručit data, i když jich komunikuje v počítači více. Port je 16 bitová hodnota, která umožňuje používat porty z rozsahu 0-65535. Port 0 je rezervován, ale je možné ho použít, pokud odesílající proces neočekává žádnou odpověď.
- Porty 1-1023 jsou tzv. dobře známé (anglicky well known ports) a na Unixech a odvozených operačních systémech jsou potřeba práva uživatele root, aby je bylo možné použít. Porty 1024-49151 jsou registrované porty. (Nemělo by se jich používat pro jiné aplikace než jsou registrované u IANA.) Porty 49152-65535 jsou používány pro komunikaci klienta se serverem.

programování!!!

**TCP**

použití (i příklady aplikací)
- posílání souborů (záleží na aplikaci, třeba u Torrentů funguje jak TCP tak UDP)
- HTTP protokol, neboli WWW stránky

vlastnosti
- kontrola došlých dat
- využívá algoritmu klouzající okénko (sliding window) - kdo dělal 2. úlohu z UDP ví o čem je řeč

porty
- Několik příkladů: FTP (port 21 a 20), SMTP (port 25), DNS (port 53) a HTTP (port 80). Registrované porty jsou typicky používané aplikacemi koncových uživatelů při otevírání spojení k serverům jako libovolné čísla zdrojových portů, ale také mohou identifikovat služby. Dynamické/privátní porty mohou být také používány koncovými aplikacemi, ale není to obvyklé.

TCP spojení
- navázání - třífázový protokol - klient pošle SYN na server, ten pátky pošle SYN + ACK a klient zpátky pošle ACK - v ideálním případě se spojí
- komunikace - řazení dat do správného pořadí, znovuposílání ztracených rámců, odstranění duplikací, kontrola přetečení zásobníku pro zpracování, využití klouzavého okénka
- ukončení - třífázový protokol - klient pošle FIN na server, ten pátky pošle FIN + ACK a klient zpátky pošle ACK - v ideálním případě spojení skončí
- rámcová znalost řízení toku - změna velikosti okénka, Nagleův algoritmus (pro využití kapacity kanálu při malém zatížení), odesílání potvrzení po větších blocích, přenastavení timeoutu

programování!!!

rozdíly oproti UDP, kdy použijete UDP, kdy TCP
- TCP je spojově orientovaný protokol. Spojení může otevřít klient nebo server a pak mohou už být posílána jakákoliv data oběma směry. Charakteristické vlastnosti TCP protokolu jsou:
  - spolehlivost – TCP používá potvrzování o přijetí, opětovné posílání a překročení časového limitu. Pokud se jakákoliv data ztratí po cestě, server si je opětovně vyžádá. U TCP nejsou žádná ztracená data, jen pokud několikrát po sobě vyprší časový limit, tak je celé spojení ukončeno
  - zachování pořadí – jestliže se odešlou 2 zprávy, jedna po druhé, první dorazí nejdřív k serveru. Pokud data dorazí ve špatném pořadí, TCP vrstva se postará o to aby některá data pozdržela a finálně je předala správně seřazená
  - vyšší režie – TCP protokol potřebuje tři pakety jen pro otevření spojení, což však umožňuje zaručit spolehlivost celého spojení

- UDP je jednodušší protokol založený na odesílání nezávislých zpráv. Charakteristika protokolu:
  - bez záruky – protokol už neumožňuje ověřit jestli došla zamýšlenému příjemci. Datagram se může po cestě ztratit. UDP nemá žádné potvrzování, přeposílání ani časové limity.
  - nezachovává pořadí – jestliže odešleme dvě zprávy jednomu příjemci, nemůžeme předvídat v jakém pořadí budou doručeny.
  - jednoduchost – nižší režie, než u TCP (není zde řazení, žádné sledování spojení atd.)

***7. přednáška = IPv6 (vlastnosti, adresace, bezpečnost, mobilita)***

**1. formát datagramu**

- Datagram v IPv6 odráža snahu tvorcov vytvoriť čo najmenšie záhlavie a to len s najzákladnejším obsahom. Tvorcovia presunuli zriedka využívané časti datagramu z IPv4 do tzv. „rozširujúcich hlavičiek“ v IPv6. Ide najmä o položky potrebné k fragmentácii datagramu, ktoré boli v každom IPv4 datagrame. Pre základné záhlavie bola stanovená pevná veľkosť, takže položka „veľkosť záhlavia“ bola vynechaná. Pri dnešnej nízkej chybovosti a vysokej priepustnosti liniek, stratilo opodstatnenie pole „kontrolná suma“. Princípy použité v IPv4 (Fragmentácia, Smerovanie) prešli len s nepatrnými zmenami.

**2. řetězení hlaviček**

- K základnému záhlaviu v IPv6 je možné pripojiť rozširujúce hlavičky a to prostredníctvom položky „ďalšia hlavička“ v základnom záhlaví. Túto položku obsahuje aj každá rozširujúca hlavička, pričom je možné týmto spôsobom pripojiť niekoľko doplňujúcich záhlaví. Hodnota položky „ďalšia hlavička“ taktiež zastupuje položku „protokol“ z IPv4, ktorou sa určuje typ údajov nesený za celým záhlavím.

**3. adresace**

- délka adresy – 128b
- druhy adres
  - Individuální (unicast)-označují jedno rozhraní připojeného počítače či zařízení.
  - Skupinové (multicast)-představují adresu skupiny síťových rozhraní. Paket se skupinovou cílovou adresou bude dopraven všem členům skupiny. Tyto adresy se používají nejčastěji pro šíření zvukového či obrazového signálu, videokonference a podobně.
  - Výběrové (anycast)-také označují skupinu síťových rozhraní, ale datagram bude dopraven jen na jedno z nich (zpravidla to nejbližší). Výběrové adresy umožňují například realizovat některé speciální služby - klient odešle datagram s obecnou adresou a některý z dostupných serverů se jej ujme.
- broadcast adresy nejsou podporovány
- zápis adres
  - FEDC:1234:0000:ABCD:0F12:0000:0000:4567
- zkracování
  - FEDC:1234::ABCD:F12:0:0:4567
  - FEDC:1234:0:ABCD:F12::4567
- prefixy
  - FEDC:1234:0000:ABC0:0000:0000:0000:0000/60
  - FEDC:1234:0:ABC0::/60

**4. fragmentace**

- MTU (Maximum transfer unit) je hodnota udávající maximální velikost přenosové jednotky na úrovni vrstvy síťového rozhraní (nejnižší vrstvy) komunikačního modelu TCP/IP.
- Protokol IP umožňuje rozdělení velkého IP paketu na menší celky (fragmenty). Každý fragment je samostatný IP paket - má svou (novou) IP hlavičku. Každý fragment potom putuje k cíli samostatně, nezávisle na ostatních. U příjemce jsou jednotlivé fragmenty poskládány a je z nich sestaven původní celek - IP paket. Každý fragment je samostatný IP paket, z čehož vyplývá, že může být podle potřeby opět rozdělen na další fragmenty.
- provádí se pouze u odesilatele
- informace o fragmentaci v rozšiřující hlavičce
- hlavičky až k fragmentační – nefragmentovatelné
- minimální MTU (Maximum Transmission Unit) pro IPv6 je 1280B (ale očekává se, že klient provede MTU Path discovery)
- vyhledávání MTU cesty (MTU Path discovery)
  - ICMPv6
  - pravidelné opakování (cca 10min)
  - nemusí se implementovat

**5. automatická konfigurace**

- dva typy automatickej konfigurácie:

- stavová konfigurácia - charakterizuje ju v IPv4 používaný systém DHCP, ktorý je upravený pre potreby IPv6 a nesie názov DHCPv6, nevyužívá broadcast
  - DHCP Unique Identifier (DUID)
    - jednoznačně identifikuje uzel
    - linková adresa a čas
    - přiděleno výrobcem
  - Identity Association (IA) - jednoznačně identifikuje rozhraní
  - vyhledání všech serverů (FF02::1:2 – adresa agenta) –> solicit - advertise
  - oslovení zvoleného serveru podle DUID (FF02::1:2)! –> request – reply
  - obnovení - renew, rebind, release, confirm
  - rekonfigurace vyvolaná serverem - reconfigure
  - bezstavová konfigurace
    - Kreativní novinkou je konfigurace bezstavová, která nevyžaduje žádné servery. Jejím základním pilířem je tak zvané objevování sousedů (neighbor discovery). Základní myšlenka je celkem prostá: každý směrovač v určitých intervalech rozesílá do sítí, k nimž je připojen, tak zvané ohlášení směrovače. V něm jsou obsaženy základní informace - především prefixy adres dané sítě a zda on sám může sloužit pro předávání paketů ven (jako implicitní směrovač, default gateway).
    - Z ohlášení směrovačů (o které může při startu aktivně požádat pomocí výzvy směrovači) se počítač dozví, jaké adresy používá zdejší síť. K nim si doplní zbývající část (typicky 64 bitů), která se jednoznačně generuje z jeho Ethernetové adresy. Tak získá platné IPv6 adresy pro své rozhraní. Také si vytvoří seznam implicitních směrovačů, kterým bude předávat pakety směřující mimo síť. Pokud je jich víc, zpočátku je prostě střídá a směrovací tabulku si postupně vylepšuje na základě jejich upozornění (ICMP přesměrování), pokud paket k určitému cíli poslal nevhodným směrovačem.
  - ohlášení směrovače
  - určení adresy
  - konfigurace směrování

**6. mobilita**

- Základní charakteristikou mobilních zařízení je, že za provozu přecházejí z jedné sítě do druhé (např. když komunikujete ze svého notebooku během cesty z Prahy do Vídně) a tudíž mění svou IP adresu. To nepředstavuje problém, pokud spojení navazuje mobilní počítač - prostě použije svou aktuální adresu.
- Potíže nastanou, pokud se někdo chce spojit s cestujícím počítačem. Řešení je poměrně prosté. Každý počítač má svou domácí síť a jí odpovídající domácí IP adresu. Tato adresa je zanesena v DNS a tu také použije externí stroj při pokusu o spojení.
- Pokud je počítač právě na cestách, zastupuje jej domácí agent. Tuto roli hraje jeden ze směrovačů v domácí síti mobilního stroje (podpora mobility zahrnuje i způsob jeho automatického výběru, aby nemusel být konfigurován staticky). Domácí agent na sebe přesměruje data určená mobilnímu stroji (při objevování sousedů odpovídá místo něj). Mobilní uzel průběžně informuje svého domácího agenta o aktuální IP adrese.
- Navázání spojení tudíž probíhá následovně: externí počítač zašle paket s žádostí o navázání spojení na domácí (trvalou) adresu mobilního uzlu. Jeho domácí agent ji zachytí a předá tunelem mobilnímu uzlu. Ten odpoví žadateli a zároveň zahájí proces nazvaný optimalizace cesty. Jeho cílem je informovat protějšek o aktuální adrese mobilního uzlu. Situaci komplikují bezpečnostní mechanismy, aby se kdokoli nemohl prohlásit za jiný počítač na cestách. Jakmile se protějšek dozví aktuální adresu mobilního uzlu, probíhá další komunikace přímo mezi nimi.

**rozdíly oproti IPv4**

- zjednodušení hlavičky
  - zjednodušuje původní hlavičku odstraněním přebytečných polí a zřetězuje volitelné podhlavičky se základní IP hlavičkou
  - mimo základní hlavičky se používají volitelné podhlavičky pro směrování, fragmentaci a ověřování přístupu - např. podhlavičky s názvem Hop-by-Hop Options, Routing, Fragment, Destination Options, Authentication, Encalupsulating Security Payload
  - tyto se používají jen v případě požadavku na danou funkci
- rozšíření adresního prostoru IP adres
  - IPv6 rozšiřuje adresní prostor z původních 32 bitů na 128 bitů. ($5.10^2$8 IP adres na jednoho současného člověka)
- automatická konfigurace uzlů
  - protokol má zapracované mechanismy automatického předělování konfiguračních údajů sítě jeho jednotlivým uzlům
  - odpadá tak manuální předělování IP adres na lokálních uzlech, respektive směrovačích sítě

- bezpečnostní procedury
  - nový protokol má přímo zabudované bezpečnostní procedury ověřování přístupu (Authentication) a kodování (Encryption) na úrovni IP komunikace
  - IPv6 umožňuje volitelný výběr bezpečnostních metod a parametrů, implicitně se však používají metody autorizace přístupu MD5 a kódování podle DES (Data Encryption Standard)
  - IP má bezpečnostní vlastnosti implementované prostřednictvím volitelných podhlaviček AH (Authentication Head) a ESP (Encapsulating Security Payload)
- podpora multimediálních aplikací
  - nárust aplikací požadujících komunikaci v reálném čase (real-time) v síti Internet si vyžádal i zvýšení podpory tohoto druhu komunikace na straně IP protokolu - IP za tímto účelem používá motodu označení toku návěští (Flow label)
  - tokem rozumíme IP pakety přenášené mezi zdrojem a cílem v IP síti se speciálními požadavky na přenos
  - přiřazením číselného návěští danému toku prostřednictvím protokolu RSVP (Resource Reservation Protocol), může IP směrovač obsluhovat přenos paketů různých toků odlišným způsobem

## *8. přednáška = Řízení toku. QoS. Protokol SCTP*

**1. řízení toku na síťové a transportní vrstvě**

- QoS - Quality of Service
  - kvalita služeb
  - snaží se zaručit koncovému uživateli doručení dat v potřebné kvalitě
  - uplatňuje se v přenosu multimédií, IP telefonii atd.
- pomocí specfikace TOS (Type of Service) v hlavičce IP protokolu (obvykle se ale ignoruje)
- pomocí pozdržení potvrzení, což zvýší RTT (Round Trip Time = doba, za kterou přijde potvrzení odeslaného datového segmentu)
- pomocí změny velikosti TCP okénka

**2. plánovací mechanismy**

- FIFO - First in, first out
  - nejjednodušší přístup - pakety jsou zpracovávány v pořadí jejich příchodu
- prioritní FIFO
  - pakety jsou rozděleny do dvou prioritních tříd: důležité pakety a "méně důležité" pakety
  - pro každou třídu existuje speciální fronta, přičemž platí, že libovolný paket z fronty důležitých paketů má přednost před libovolným paketem z fronty méně důležitých paketů
  - vzájemné pořadí mezi všemi důležitými pakety (a stejně tak i mezi těmi "méně důležitými") je jednotlivými frontami zachováno
  - front (a tím pádem i prioritních tříd) může být i více než 2
- WFQ - Weighted fair queuing
  - každý datový tok má svojí frontu
  - pokud jeden tok např. posílá rychleji nebo větší packety než ostatní, projevuje se to jen na zpracování onoho toku a netrpí tím ostatní (všechny fronty mají stejnou prioritu)
- Leaky Bucket (nebo spíš Token Bucket)
  - v zásobníku na tokeny je určitý počet tokenů a když přijde packet o velikosti N bytů na řadu, je odebráno ze zásobníku N tokenů. Pokud jejich počet nestačí, je s packetem nakládáno dle implementace (zahození, ...). Jinak je odeslán.
- Round Robin
  - funguje jako bariéra v datovém přenosu
  - mechanismus Round Robin postupně po dobu krátkého časového kvanta propouští pakety z jednotlivých datových toků, které dorazily k bariéře
  - některé toky můžou mít přednost

**3. řízení toku v protokolu TCP**

- efektivně lze omezovat pouze odchozí tok dat
- pro příchozí tok musíme použít nepřímé metody
- můžeme pozdržet potvrzení a tím prodloužit RTT (Round Trip Time) - pozor na timeout, který působí opětovné odeslání dat
- můžeme měnit velikost okénka

## *9. přednáška = Adresářové služby (DNS, X.500)*

**1. jmenné služby - použití**

- specializovaná databáze
  - adresář
  - optimalizace pro čtení a vyhledávání

        - občasné aktualizace
        - distribuce
        - replikace
        - chybí podpora transakcí
    - úlohy
        - překlad
        - ověření
        - vyhledání (lokalizace)
        - poskytování podrobných informací (vlastností)

**2. LDAP - Lightweight Directory Access Protocol**

- co je to
    - protokol pro ukládání a přístup k datům na adresářovém serveru
    - podle tohoto protokolu jsou jednotlivé položky na serveru ukládány formou záznamů a uspořádány do stromové struktury (jako ve skutečné adresářové architektuře)
    - je vhodný pro udržování adresářů a práci s informacemi o uživatelích (např. pro vyhledávání adres konkrétních uživatelů v příslušných adresářích, resp. databázích)
    - je založen na doporučení X.500, které bylo vyvinuto ve světě ISO/OSI, ale do praxe se ne zcela prosadilo, zejména pro svou „velikost“ a následnou „těžkopádnost“
- typy dat
    - záznam - identifikován DN
- atributy
    - tvoří záznam
    - jméno
    - typ
    - hodnoty
- typy
    - int
    - bin
    - cis
    - ces
    - dn
    - tel
- třídy – objectclass
    - definice atributů
    - standardní třídy
    - uživatelské třídy
- DIT - Directory Information Tree
    - struktura adresářového stromu
- odkaz na jiný LDAP server
    - atribut ref a LDAP URL
    - ldap://ldap.cvut.cz/o=cvut,c=cz

**3. aplikace**

- servery
    - University of Michigan LDAP server
    - OpenLDAP server
    - iPlanet Directory Server
    - Active Directory
- email klienti
    - MS Outlook, MS Outlook Express - Adresář
    - Evolution
    - Thunderbird
- samostatní klienti
    - LDAPconf
    - OpenLDAP
- Samba
- LDAP PAM

**4. DNS - Domain Name System**

- organizace dat v DNS
    - typy dat
    - distribuce
    - domény

typy serverů (autoritativní znamená, že jeho odpověď je brána za správnou)
- primární
  - udržuje data o zóně
  - autoritativní server
- sekundární
  - kopíruje si data z primárního serveru
  - autoritativní server
- caching only
  - není autoritativní pro žádnou zónu (na rozdíl od pri. a sek.)
- root
  - udržuje záznamy root domény
- forwarding
  - předává rekurzivní dotaz (odlehčení linky), může sám resolvovat
- slave
  - jen forwarding (problém s terminologií)

rekurzivní a nerekurzivní dotaz
- rekurzivní řešení dotazu
  - server se chopí vyřízení dotazu, najde odpověď a pošle ji tazateli
  - rekurzivní přístup server zatěžuje (musí sledovat postup řešení, ukládat si mezivýsledky apod.), ale projde jím odpověď a tu si může uložit do vyrovnávací paměti
  - typicky se tak chovají lokální servery, aby si plnily vyrovnávací paměti a mohly dalším tazatelům poskytovat odpovědi rovnou
- nerekurzivní řešení dotazu
  - server dotaz neřeší, pouze poskytne tazateli adresy dalších serverů, jichž se má ptát dál
  - takto se chovají servery ve vyšších patrech doménové hierarchie (kořenové a autoritativní servery TLD), které by rekurzivní chování kapacitně nezvládaly

reverzní dotaz
- překlad IP adresy na doménu
  - DNS obrátí pořadí bajtů v adrese
  - obrácené adrese pak připojí doménu in-addr.arpa
  - výsledné „jméno“ vyhledává standardním postupem
- hledá-li například jméno k IP adrese 145.97.39.155, vytvoří dotaz na 155.39.97.145.in-addr.arpa

## *10. přednáška = Bezpečnost (principy, symetrické a asymetrické šifry, digitální podpis)*

**1. vysvětlení pojmů**
- šifrování
  - kryptografický algoritmus, který převádí čitelnou zprávu neboli prostý text na její nečitelnou podobu neboli šifrový text
- autentizace
  - ověření totožnosti odesílatele a příjemce
- podpis
  - odesilatel vytvoří otisk (hash) původního textu, tento hash zašifruje svým privátním klíčem a výsledek odešle společně s nezašifrovanými daty
  - příjemce dešifruje otisk veřejným klíčem odesílatele a porovná jej s vlastnoručně vytvořeným otiskem otevřených dat
  - podpis zaručuje, že nedošlo ke změně zprávy ani zfalšování odesilatele
- integrita
  - zaručení, že dokument nebyl během přenosu změněn
  - zajištěná zpravidla pomocí hashe či MAC (Message Authentication Code)
  - pokud jde o bezpečnost, není vhodné CRC (je lineární a po změně zprávy se dá odhadnout změna CRC)

**2. symetrické šifrování a šifrování s veřejným klíčem**
- symetrická šifra
  - šifrovací algoritmus, který používá k šifrování i dešifrování jediný klíč
  - nízká výpočetní náročnost
  - velkou nevýhodou je nutnost sdílení tajného klíče, takže se odesilatel a příjemce tajné zprávy musí předem domluvit na tajném klíči
  - často se používá společně s asymetrickými
  - obvykle se otevřený text zašifruje symetrickou šifrou s náhodně vygenerovaným klíčem, tento symetrický klíč se zašifruje veřejným klíčem asymetrické šifry, takže dešifrovat data může pouze majitel tajného klíče dané asymetrické šifry
- šifrování s veřejným klíčem

- pro šifrování a dešifrování se používají odlišné klíče
- šifrovací klíč pro asymetrickou kryptografii sestává z dvou částí:
- jedna část se používá pro šifrování zpráv (a příjemce zprávy ani tuto část nemusí znát)
- druhá pro dešifrování (a odesilatel šifrovaných zpráv ji zpravidla nezná)
- je vidět, že ten, kdo šifruje, nemusí s dešifrujícím příjemcem zprávy sdílet žádné tajemství, čímž eliminují potřebu výměny klíčů; tato vlastnost je základní výhodou asymetrické kryptografie
- typicky ale bývá problém s bezpečnou první výměnou veřejných klíčů

**3. autentizace**

- poskytnutí informace o identitě jednoho subjektu subjektu druhému
- provádí se před použitím dalších protokolů
- často se provádí oboustranně
  - způsoby
    - pomocí jména a hesla
    - pomocí biometrických údajů
    - pomocí něčeho, o čem se ví že uživatel má (specifický hardware, USB dongle,...)
    - pomocí správné odpovědi na náhodně vygenerovaný specifický dotaz
    - pomocí certifikátu - uživatel pošle certifikát podepsaný CA, který obsahuje osobní údaje a veřejný klíč. Server pomocí veřejného klíče zakóduje náhodně vygenerovaný token a pošle zpět. Uživatel token rozkóduje pomocí privátního klíče, odešle zpět a pokud se rozkódované tokeny rovnají, uživatel je vpuštěn do systému.
  - rozdíly

**podpisování**

- odesílatel udělá hash zprávy, ten zašifruje svým privátním klíčem. Následně odešle zprávu a tento vzniklý otisk příjemci.Příjemce dešifruje otisk veřejným klíčem, který získá od certifikační autority, udělá si vlastní hash zprávy a porovná je.

**distribuce klíčů**

- asymetriké šifry se často používají pro distribuci náhodně vygenerovaných klíčů pro následné symetrické šifrování, které je výrazně rychlejší. Příklad použití: SSH

**certifikát**

- rámcový obsah:
  - identifikace držitele
  - identifikace vydavatele
  - veřejný klíč držitele
  - podpis vydavatele
  - data vystavení/trvání
- použití
  - Držitel jej vytvoří a nechá podepsat CA. CA ověří totožnost a podepíše ho. Použití: certifikát dám protějšku (obsahuje veřejný klíč), protějšek může ověřit platnost klíče pomocí veřejného klíče (certifikátu) CA. Lze to i rekurzivně.

## *11. přednáška = Zabezpečení sítě (pravidla, firewally, NAT, ssh, ssl, ipsec, vpn)*

**Firewall**

- popis
  - síťové zařízení sloužící k řízení a zabezpečování síťového provozu mezi sítěmi s různou úrovní důvěryhodnosti a/nebo zabezpečení
  - zjednodušeně se dá říct, že slouží jako kontrolní bod, který definuje pravidla pro komunikaci mezi sítěmi, které od sebe odděluje
  - modernější firewally se opírají přinejmenším o informace o stavu spojení, znalost kontrolovaných protokolů a případně prvky IDS
- **zapojení (jen filtr, s DMZ …)**
  - jen filtr - firewall pouze filtruje příchozí nebo odchozí pakety
  - DMZ - demilitarizovaná zóna, typicky lokální síť, po které běhají pakety neomezeně, firewall kontroluje až všechno co jde ven/dovnitř DMZ
- pravidla
  - Nastavení pravidel pro komunikaci přes firewall se běžně označuje termínem „bezpečnostní politika firewallu“, zkráceně „bezpečnostní politika“. Bezpečnostní politika zahrnuje nejen samotná pravidla komunikace mezi sítěmi, ale u většiny dnešních produktů také různá globální nastavení, překlady adres (NAT), instrukce pro vytváření šifrovaných spojení mezi šifrovacími branami (VPN – Virtual Private

Networks), vyhledávání možných útoků a protokolových anomálií (IDS – Intrusion Detection Systems), autentizaci a někdy i autorizaci uživatelů a správu šířky přenosového pásma (bandwidth management).

### *12. přednáška = Speciální sítě (FibreChannel, NAS, SAN)*

**Způsoby připojení datových zařízení (DAS, NAS, SAN)**

- **DAS** (Direct attached storage)
  - vlastnosti - ulozne zarizeni (paska, disk) primo na serveru
  - nevyhody - decentralizovanost při větším počtu zařízení, a tím i mnohem větší náklady na správu systémů, nedostupnost dat při výpadku hostujícího serveru a omezená rozšiřitelnost
  - vyhody - velmi levne, jednoduchá instalace, údržba a provoz
  - používané protokoly - SCSI, SAS a Fibre Channel
- **NAS** (Network attached storage)
  - Klient ví, že se jedná o externí data a přistupuje k nim pomocí síťového protokolu. (FTP, HTTP, ...) Systém uložení dat určuje NAS. Výhodou je menší problémy se sériovým přístupem k jenomu médiu. Nevýhodou je možná zátěž sítě. (diskové pole v síti)
  - používané protokoly - NFS, HTTP, FTP, CIFS,...
- **SAN** (Storage area network)
  - Datové úložiště se pro operační systém jeví, jako by se jednalo o logický disk. Operační systém si na něm musí sám udělat systém souborů. Problémem je sdílení jedné logické jednotky mezi více systémy. Výhodou je snadné síťové bootování. SAN manipuluje s mnohem většími bloky dat než NAS a DAS, nedochází ke zpoždění, zátěž je rozložitelná na více serverů, úložná zařízení jsou připojena optickým kabelem. (diskové pole za serverem)
  - používané protokoly iSCSI, FCIP, IFCP

### *13. přednáška = Správa sítí (SNMP, CMIP, RMON, aplikace pro dohled sítí)*

**1. oblasti síťové správy**

- Management Information Base (MIB) popisuje sadu objektů, které jsou předmětem správy. Spravované zařízení může implementovat jednu nebo více MIB, v závislosti na jeho funkci. Tyto MIB databáze jsou velmi podobné standardním databázím v tom smyslu, že popisují jak strukturu, tak formát dat. Dělí se na tyto oblasti:
  - správa výkonu (performance management)
    - reaktivní a proaktivní
    - měření výkonnosti a zatížení
  - správa konfigurace (configuration management)
    - monitorování síťové konfigurace
  - účetní správa (accounting management)
    - monitorování využití sítě
  - správa poruch a chyb (fault maagement)
    - detekce chyb, logování a oznámení
  - správa bezpečnosti (security management)
    - nastavení a monitorování přístupu

**2. architektura a součásti síťové správy (server, agent, protokol)**

- Network Management Server využívá Net. Man. Protocol a posílá do sítě povely. Ty přebírá agent, který je v samotném uživatelově počítači (Managed device)

**3. SNMP**

- Simple Network Management Protocol (SNMP) je součástí sady internetových protokolů. Slouží potřebám správy sítí. Umožňuje průběžný sběr nejrůznějších dat pro potřeby správy sítě, a jejich následné vyhodnocování. Na tomto protokolu je dnes založena většina prostředků a nástrojů pro správu sítě. Má tři verze: druhá obsahuje navíc autentizaci a třetí šifrování. Nejvíce zařízení podporuje druhou verzi.
- volání (get, set …)
  - get-request -získání informace z MIB;
  - get-next-request - umožňuje managerovi získat informace o objektech v MIB bez znalosti jejich přesných jmen, umožňuje postupné procházení celým hierarchickým stromem;
  - set-request - změna hodnoty proměnné agenta;
  - trap - jediný typ příkazu vysílaný bez předchozího vyžádání, agent jej zasílá managerovi jako reakci na specifikovanou událost, zpráva zůstává nepotvrzená, proto agent nemá jistotu, zda byla doručena;
  - get-response - agent vykoná tuto operaci jako reakci na předchozí příkazy - je to vlastně odpověď agenta managerovi. Odpověď obsahuje i dotaz, protože protokol nezajišťuje souvislost mezi dotazem a odpovědí.
  - get-bulk - operace, která je součástí SNMP v.2. Umožňuje vyžádat si k přečtení celou skupinu informací z MIB, čímž se mnohdy urychluje komunikace.

- inform - umožňuje komunikaci dvou managerů mezi sebou.

- verze
  - SNMPv1 - Stanovovala jen příkazy get, get next, set a trap, ale využívala pouze ochranu heslem - community string. Toto byl pravděpodobně hlavní nedostatek, který přetrval ještě ve verzi 2.
  - SNMPv2 - Definovala příkaz get bulk pro snazší získávání informací z tabulek (např. routing table atp.). O efektivitě tohoto příkazu by se však dalo diskutovat. Ochrana heslem je nedostatečná, protože heslo lze poměrně jednoduše zjistit analyzátorem paketů
  - SNMPv3 - Současná specifikace protokolu SNMPv 3 pochází z roku 2002 a umožňuje ochranu dat i pomocí DES algoritmu.
- MIB
  - MIB (Management Information Base) je databáze, která dovoluje jednoznačně identifikovat informace využívané systémem správy. Aby mohl SNMP manager i agent tyto informace získat a předávat, tak je nutná znalost struktury MIB.
    - jaké proměnné může nabízet
  - Proměnné jsou definovány hierarchií (řadou) číslic, která je popsaná v MIB tabulce, kde je popsán význam jednotlivých proměnných, jejich formát a název. Pokud ale znáte hierarchii (řadu číslic – například „.1.3.6.1.4.1.21796.3.4.1.1.2.4“ – stav 4 binárního vstupu) pro konkrétní hodnotu, MIB tabulku nepotřebujete.

**4. RMON**

- popis
  - Vzdálené sledování sítě. V LAN úseku je sonda, která ho monitoruje. Na vyžádání monitorovací stanice jí posílá různé informace
- co nabízí
  - Statistic – okamžitý provoz (byte, kolize, chyby ...)
  - History – dtto, ale historie
  - Alarms – prahování měřených hodnot
  - Hosts – přenosová statistka pro každý uzel
  - Host Top N – setříděná předchozí statistika
  - Traffic Matrix – vzájemná komunikace mezi uzly
  - Filters – filtry pro následující skupinu
  - Packet Capture – pakety pro další dekódování
  - Events – záznamy událostí