UvPS - 1. semestr

revision:
20.06-21.06: go through everything
22.06: dig into details, do exercise questions
23.06-24.06: go through everything once again

1. Přednáška

Vznik počítačových sítí, Internetu

1. Sálový počítač - zadávání velkých množství dat: **děrné štítky** - pro přenos dat mezi počítači
2. Terminál (s obrazovkou) - **point-to-point komunikační kanál**
3. PC - jsou navzájem propojeny Local Area Network. **Klient-server**. WAN nahradil point-to-point kanály

- ARPANET - první síť založena na point-to-point spojení pronajaté telefonními linky
- teď: TCP/IP

Požadavky na síť

na Odolnost

- **přepojování okruhů**
  - v minulosti
  - vytvoří se posloupnost uzlů (okruh) spojující startovní a koncové zařízení. Po vytvoření *okruhu* se **všechna** zvuková data přenáší touto cestou
  - *výhody a nevýhody*: rychlejší, plynulejší, ale při výpadku uzlu se spojení rozpadne
- **přepojování paketů**
  - nové sítě
  - data se rozdělí na bloky, tzv. *pakety*
  - každý paket najde vlastní cestu k cílovému uzlu
  - *výhody a nevýhody*: pomalejší, ale výpadek uzlu není fatální (či *odolné vůči chybám*)

na Bezpečnost

- minulost:
  - *fyzická* bezpečnost - 1. priorita
  - absence šifrování, otevřená komunikace, důvěra v obsah dat
- nyní:
  - bezpečnost dat - ověřování uživatelů / počítačů, kryptografie, inspekce dat

na Rozšiřitelnost

- usnadnění přidání nového počítače/sítě do lokální sítě/internetu

LAN

1. **Core** = hlavní část sítě
   - připojená k ISP (Internet Service Provider).
   - zařízení jsou umístěna v místnosti s klimatizaci a záložním zdrojem napětí
   - **Router / Směrovač** = uzel propojující sítě
   - **Switch / Přepínač** = uzel propojující hlavní router se zbytkem lokální sítě (je jich několik)
2. **Distribution** = vertikální vrstva
   - část sítě, distribuje konektivitu do všech částí budovy/kampusu
3. **Access** = horizontální vrstva
   - umožňuje přístup k síťovým službám všem zařízením

WAN

vrstvy

- **Tier1**
  - (globální) společnosti s přímým přístupem k páteři internetu (IXP - Internet Exchange Point) - či má vlastní páteřní síť
  - jejich IXP propojuje všechny kontinenty
  - př. AT&T, Deutsche Telekom (T-Mobile)
- **Tier2**
  - národní společnosti (př. O2 Czech Republic)
  - nemá přímý přístup k páteři - závisí na Tier1
- **Tier3**
  - koncové zákazníky - společnosti, organizace, domácnosti
  - připojují své LAN

Na kvalitu služeb

Přenosové parametry sítě

- **Latence** = zpoždění dat
- **Jitter** - pravidelnost doručování dat
- **Ztrátovost dat** - jak často nějaký paket se ztratí a nebude doručen
- **Šířka pásma** - kolik dat se dá přenést pomocí konkrétních fyzických signálů

Různé aplikace - různé požádavky:

- multimédia - pravidelné doručení (jitter)
- www, pošta - nízká ztrátovost dat (ztrátovost dat)

Kvalita služeb - způsoby

- **QoS** - klasifikace dat dle jejich přenosu
  - označení všech dat vysokou prioritou, - nefér
- **vyhrazená šířka pásma** - pro veškerý potřebný provoz
  - zaručená kvalita
  - plýtvání kapacitou kanálu
- **Best Effort** - zaručuje dostatečnou rychlost doručení zpráv pomocí prioritních front
  - neplýtváme kapacitou kanálu
  - nevyhoda: nezarucuje lepsi kvalitu

Základní dělení sítí

Lokální sítě / LAN (Local area network)

- menší vzdálenosti
- sdílení zdrojů mezi vzdálenými počítači (servery, tiskárny)
- jednotná - vlastník ji spravuje

Rozlehlé sítě / WAN

- přenos dat na větší vzdálenosti
- vzdálený přístup
- (distribuovaná) mnoha vlastníky

Veřejné a privátní sítě

- LAN - privátní
- non-LAN - veřejné
- VPN (Virtuální privátní síť)
  - dva uzly, každý je umístěn v jedné ze dvou poboček LAN (privátní síť), jsou propojeny VPN tunelem, který vede přes veřejnou síť

RFC (Request for Comments)

- řada dokumentů popisující internetové protokoly
- dokument se nikdy nemění, aktualizace mají nové číslo
- jsou volně šiřitelné

2. přednáška

Síťový model, síťová architektura

- Síťový model
  - popisuje počet vrstev, jejich strukturu a funkce
  - např. OSI model
- Síťová architektura
  - síťový model + protokoly, rozhraní mezi vrstvami, konkrétní služby/technologie
  - např. TCP/IP (sada protokolů)

OSI (Open Systems Interconnection)

- je to **síťová architektura**, která se skládá z:
  - základní model (vrstva + funkce)
  - a sada protokolů
- *nevýhody*: architektura jako celek byla budovaná shora (7 -> ... -> 1), předidealizovaná a nepraktická

OSI	Název	Funkce
1	Fyzická	fyzický přenos bitů mezi uzly

OSI	Název	Funkce
2	Linková	přenos dat mezi dvěma přímo propojenými uzly
3	Síťová	přenos a směrování paketů mezi uzly, které se nacházejí v různých sítích. Délka paketů - **proměnlivá, omezená**
4	Transportní	přenos datových paketů s *neomezenou* délkou
5	Relační	řídí dialog mezi aplikacemi
6	Prezentační	skryje rozdíly mezi implementacemi sémantiky dat (=můžou mít různý význam)
7	Aplikační	vrstva nejbližší koncovému uživateli

X.400, X.500

- OSI protokoly
- implementace služeb na základě OSI protokolů
- Sada standardů OSI se označují X.číslo

Staré standardy

- **X.400** (sada standardů)
  - definuje protokoly např. Message Handling System (pošta)
  - nad transportní vrstvou
  - adresy jsou *jednoznačné* - mají hodně komponent (jako pošta)
  - později byly nahrazeny kratšími, nejednoznačnými adresami (e-mail)
- **X.500** (sada standardů) -
  - definuje protokoly např. Directory Access Protocol (DAP)
  - první implementace telefonního seznamu
  - jednoznačná identifikace osob (podle jejích atributů)

Nyní

- Asymetrická kryptografie - infrastruktura pro správu veřejných klíčů
- LDAP - protokol, který získává a spravuje informace o uživatelích a službách

Rodina protokolů TCP/IP

- architektura navržena odspodu
- protokoly řeší jeden problém

OSI	definice	protokoly
1, 2	síťové rozhraní (není v TCP/IP)	Ethernet, WiFi (=média)
3	síťová	IP (--v4, --v6)
4	transportní	TCP, UDP
5,6,7	aplikační	HTTP, DNS, NFS
mimo hierarchii		ARP, ICMP

TCP a UDP

TCP (Transaction Control Protocol)

- určen pro **Spojované služby** (telefon)
- spolehlivé doručení dat
- aplikace je jednodušší, průběh přenosu dat závisí na spodní vrstvě. Aplikace nemůže řídit průběh komunikace
- TCP segmentuje data na menší bloky, odesílá je v jednotlivých paketech, v případě selhání je znovu odesílá

UDP (User Datagram Protocol)

- určené pro **Nespojované služby** (pošta)
- není zaručeno správné pořadí ani doručení paketů - se posílají jednotlivé pakety
- aplikace (vy) řídí komunikaci, odpovídá za správnost přenosu dat

Aplikační modely

- Model Klient-Server
  - klient navazuje komunikaci, zadává požadavky, server ho obsluhuje (i více klientů)
  - klientovi je daná pevná adresa serveru
  - př. WWW, SMTP
- Model Peer-to-Peer
  - není dana pevná adresa zdroje dat
  - každý je zároveň klientem a serverem
  - šíří data ilegálně
  - př. BitTorrent, Napster

Adresování počítačů

- potřebujeme znát **adresu serveru**, abychom ho kontaktovali
- ve vrstvě, ve které probíhá komunikace, se používá **konkrétní adresa**
- adresy:
  - **MAC adresa** - je v linkové vrstvě
    - dřív: daná výrobcem, dnes: *nastavitelná* (síťové karty mají ji uloženou v paměti)
    - nelze použit pro komunikaci mezi různými sítěmi, protože nerespektuji síťovou topologii
    - př. ethernetové adresy (6B) - prefix vyrobce, cislo sitove karty
  - **IP adresa** - je v síťové vrstvě
    - přidělována podle topologie sítě
    - každému počítači je přiřazena adresa podle toho, kde je připojen k síti
  - **doménová adresa/jména** - je v aplikační vrstvě
    - uživatelsky přívětivé
    - přidělována podle organizace
- DNS - převod mezi doménovými jmény a IP adresami
- ARP - převod mezi síťovými a MAC adresami

Domény

Doménový systém

- hierarchická struktura zón, které obsahují informace o podřízených počítačích a zónách
- informace jsou v databázi sdílená množinou nameserverů pro danou zónu

Správa domén

- doména nejvyšší úrovně (TLD)
  - spravovány ICANN, jejich struktura definovaná v RFC 920
  - rezortní (com, org, ..., info, aero,), ISO kódy zemí (cz., uk/eu), internacionalizované kódy (.рф)
  - nyní lze mít privátní TLD
  - .cz - spravována CZ.NIC (korporace českých ISP)
- SLD a nižší domény
  - spravovány jejich majiteli [ms.[mff.[cuni.cz]]]

IP adresy

- každý koncový uzel v TCP/IP musí mít IP adresu
- IP adresa má 2 části:
  - adresa sítě (pro směrování mezi sítěmi)
  - adresa počítače (používá se uvnitř sítě)
- IP v4 (4 byty)
  - v ČR nejvíc používané
  - dekadická notace (. . .)
- IP v6 (16 bytů)
  - dvoubajtové bloky oddělené ":"
  - skupina nulových bloků "::"
- přiřazení adresy síti:
  - ISP přiděluji síti **veřejné adresy**
  - správce sítě v LAN přiděluje síti jeden/více bloků **privátních adres**
  - pro přístup k Internetu, na začátku se musí použit **překlad adres (NAT)**
- přiřazení adresy počítači:
  - v místní síti: správce sítě zvolí přiřazení IP adres. Přiřazení může být
    - statické (uzel má předdefinovanou IP adresu) vs. dynamické (adresa je přiřazena na vyžádání)
    - volné (může se připojit kdokoliv) vs. omezené (autentikace)
  - platí i pro **privátní adresy**
  - **link-local adresy** - každý počítač si je volí sám (duplicitní adresy) -> slouží pro komunikaci v místní síti (jenom pokud jiné počítače používají link-local adresy)

Port, socket

- **Socket** = jeden konec komunikačního kanálu mezi klientem a serverem. <IPadresa, port>
- **Port** = 16bitové číslo, které slouží k rozlišení různých služeb/aplikací
  - **well-known services** = předdefinované číslo portu, např. port 80 - webové servery
  - **destination port** = některý z well-known services, musí je klient znát
  - **source-port** = přiřazení neobsazeného čísla
- příklady: 80, 443/TCP: HTTP
  - přenos web stránek

Překlad adres (NAT)

- lokální síť používá privátní adresy, a ven se představuje veřejnými adresami
- průběh:
  - směrovač v LAN zachytí paket ,si uloží jeho socketovou adresu, nahradí v paketu vlastní IP adresou a nějakým volným portem. Server odešle odpověď na tuto upravenou socketovou adresu, směrovač vyhledá původní adresu socketu a změní hodnoty paketu zpět na původní hodnoty IP a port z požádavku

Adresování služeb

- URI (Uniform Resource Identifier)
  - jednotný systém odkazů
  - 1 klient pro více služeb (FTP, WWW)
  - členění: URL (umístění zdroje), URN (název služby bez explicitního umístění)
- struktura adresy URI
  - **schéma**":" má název protokolu, je to schéma odkazu
  - "//"**autorita**: představuje server / doménu
  - **cesta**= cesta k požadovanému zdroji v rámci serveru (něco jako souborový systém)
  - **dotaz** -ho může obsahovat HTTP URI např. data převzata z HTML formuláře
  - **fragment** - identifikátor bodu uvnitř stránky
- http://1.2.3.4:8080/q?ID=123#Local

Datový tok v TCP/IP

- komunikace 2 aplikací (www klient, www server)
- z aplikační do linkové vrstvy, a zpátky

OSI	co se děje
5-7	uživatel zadá URL (takhle adresuje server). Aplikační vrstva předá data se cílovou socketovou adresou OSI 4
4	Transportní vrtsva předá data spolu s cílovou IP adresou OSI 3
3	Síťová vrstva. Pokud cílová IP adresa ve stejné síti - next-hop uzel = cílový server. Jinak: next-jop uzel = router, přesměruje data OSI 2
2	cílová adresa na Linkové vrtsvě: MAC adresa next-hop uzlu
1	přenesení dat na jiný konec kom. kanálu

Pak zpracování dat vyššími vrstvami. Jinak pokud požadavek neskončil, jsme pořád na síťové vrstvě

Multiplexing

- několik kom. kanálů v určité vrstvě používá stejný komunikační kanál v nižší vrstvě
- obě strany kom. kanálu dodržují protokol
  - na jedné vrstvě může probíhat komunikace ve více vrstvách současně
- PDU (Protocol data unit) - formát dat nějaké vrstvy
  - můžeme PDU zapouzdřit do jakékoliv vrstvy
  - sestává z těla obsahujícího PDUn, v záhlaví: řídicí informace

- Interface = rozhrání, či výměna dat mezi vrstvami
  - formát jejich dat = Interface Data Unit
- např. odešleme paket IPv6 přes síť IPv4 = PDU3 zapouzdřit do PDU3

Typ PDU v TCP/IP

OSI	PDU
5-7	stream, message
4	**data**, socketová adresa
3	**segment**, IP adresa
2	**packet**, mac adresa
1	**frame / rámec**

rámec - Frame Check Sequence - ten se vypočítá z obsahu zbytku rámce. Linková vrstva porovná tuto hodnotu s hodnotou uloženou v rámci, a zkontroluje MAC adresu, zda rámec patří ke konkrétnímu uzlu

3. přednáška

Kyberbezpečnost

Autentikace, autorizace

- Autentikace = proces ověření osoby
  - **lokální** autentikace podle: *znalostí* (heslo, PIN), *technických prostředků* (klíč, HW token), *biometrie* (otisky prstů, face-id)
  - **vzdálená** autentikace: *ochrana proti odposlechu* (systém jednorázových hesel), *přenos dat v protokolu* (pomocí SASL), *použití autentikačního serveru/protokolu* (LDAP, RADIUS)
- Autorizace = zpřístupnění služeb pro verifikovanou osobou

One-Time Password (OTP)

- = systém jednorázových hesel

1. vygenerování seznamu jednorázových hesel (nešifrované spojení)
2. **challenge-response**: server pošle výzvu (=náhodný řetězec), na ní uživatel odpoví heslem a HW/SW kalkulačkou a výsledek z kalkulačky napíše serveru jako odpověď
3. **token** - se serverem generuje jednorázový časově omezený kód

Kryptografické algoritmy

Symetrické šifrování

- pro šifrování a dešifrování se používá **stejný klíč**
- **výhody**: rychlé, vhodné na *velká data*
- **nevýhody:** partneři si musí klíč předat bezpečnou cestou
- př. DES, Blowfish, AES, RC4 (algoritmy)

Asymetrické šifrování

- pro šifrování a dešifrování se používá dva různé klíče
- mat. základ: **jednocestné funkce**,
  - příkladem takové funkce: **diskrétní logaritmus** - $m = p^k \mod q$
    - máme daný součin - chceme najít jeho dva dělitele
- **výhoda**: jeden klíč lze veřejně šířit, druhý tajně uschovat
- **nevýhoda:**
  - pomalé algoritmy
  - lze šifrovat jen malá data
  - veřejný klíč je třeba pečlivě ověřovat
- př. RSA, DSA (Digital Signature Algorithm)

Hashovací funkce

- vytvoření kódu z daného textu. Kód je krátký a pevné délky
- malá změna textu = velká změna hashe (**skoro jednoznačný**)
- hash je jednocestný, tedy je z textu **neodvoditelný**
- nalezení textu se shodným hashem je obtížné
- široké uplatnění - kontroly shody, výběr z tabulky, ...
- př. CRC, MD5

Využití kryptografie

Šifrování dat

- používá kombinaci symetrických a asymetrických algoritmů
- průběh:
  - **předpoklad**: text zašifrovaný přes nezabezpečený kanál (např. e-mail)
  - vygenerujeme náhodný klíč a symetricky jim zašifrujeme text
  - zašifrujeme tento klíč asymetricky pomocí veřejného klíče příjemce
  - zašifrovaný klíč je připojen k zašifrovanému textu
  - **po doručení**: příjemce dešifruje zašifrovaný klíč svým soukromým klíčem, a pak jim symetricky dešifrujeme text
- chrání před přečtením kýmkoliv

Elektronický podpis

- kombinace asymetrických a hashovacích algoritmů
- průběh:
  - odesílatel zvolí hashovací funkci, vezme libovolný text a vypočítá jeho hash
  - pak vezme svůj soukromý klíč a zašifruje hash, ten je pak připojen k původnímu textu
  - příjemce převezme text, použije stejnou hashovací funkci a vypočítá hodnotu hashe
  - pak vezme veřejný klíč odesílatele a dešifruje hash
- zabraňuje komukoliv měnit data

Diffie-Hellmanův algoritmus

- způsob sdílení tajných informací nezabezpečeným kanálem
- základ řady protokolu založených na symetrické kryptografii
- průběh:

- vygenerování tajných čísel a, b
- veřejná prvočísla p, q
- co pošlou obě strany: $A = p^a \mod q$, $B = p^b \mod q$
- pak porovnají $s = B^a \mod q, s = A^b \mod q$
- s=s

Autenticita veřejných klíčů

- musíme ověřit, že jmenovka patří klíči
- autenticitu ověří třetí strana (známý člověk / certifikační autorita) a připojí svůj podpis / identifikační značku

Certifikát

- veřejný klíč doplněný o identifikaci vlastníka a podepsaný vydavatelem

SSL, TLS

- SSL - Secure Socket Layer, TLS - Transport Layer Security
- mezivrstva mezi transportní a aplikační vrstvou TCP/IP
- umožňuje autentikaci a šifrování
- protokoly např. HTTPS port 443
  - "htttps" = http provozované přes SSL

Aplikační vrstva TCP/IP

- popisuje
  - průběh dialogu
  - formát zpráv - textové, binární
  - typy zpráv - požadavky a odpovědi
  - sémantika dat
  - interakci s transportní vrstvou
  \
- protokoly:

DNS (Domain Name System)

Definice

- =služba pro překlad doménových jmen na adresy, a naopak (je realizovaná stejnojmenným protokolem)
- protokol využívá TCP a UDP (port 53)
  - UDP - běžné dotazy (512 B - jsou krátké. EDNS protokol umožňuje větší velikosti)
  - TCP - větší výměny dat
    - Pokud odpověď přesahuje datový limit, server nastaví příznak TC (truncated), pak můžeme dotaz zopakovat
- protokol je binární
  - každá zpráva obsahuje hlavičku a resource record (záznam)

DNS záznamy

- obsahují:

- jméno domény, TTL (doba platnosti v sekundách), typ záznamu, kanonické jméno

Typ	Obsah
SOA	obecné info o doméně
NS	název nameserveru domény
A	IPv4 adresa počítače
AAAA	IPv6 adresa počítače
PTR	reverzní záznam - doménové jméno počítače
CNAME	záznam pro aliasy (jméno počítače)
MX	jméno poštovního serveru

Servery DNS

- typy serverů:
  - **primární** - spravuje záznamy o doméně
  - **sekundární** - stahuje z primárního serveru obsah záznamů, a obnovuje ji
  - **caching-only** - udržuje nevyřešitelné dotazy po dobu platnosti
- primární + sekundární = autoritativní servery
  - jejich data pochází z ověřeného zdroje
  - každá zóna v DNS má alespoň jeden autoritovních nameserverů

Vyřizování DNS dotazu

- po zadání www.mff.cuni.cz - dotaz je rekurzivní (nameserver má ho vyřídit a vrátit odpověď)
- pokud server nemá ve své cache žádnou informaci o doméně, obrátí se na některý z kořenových nameserverů a uloží je do své cache (ty nefungují rekurzivně. Pak na dotaz konečně odpoví autoritativní nameserver
- DNS dotaz a odpověď:
  - dotaz - v záhlaví: (ID - 2B náhodné číslo), příznaky (požádavek na rekurzivní odpověď), - a query (s jediným RR)
  - odpověď
    - v záhlaví: identifikátor dotazu, příznaky (napr. Authoritative Answer)
    - v query: zopakovaný dotaz
    - v answer: RR s odpověďmi
    - v authority: seznam autoritativních nameserverů
    - additional

Bezpečnost DNS

- někdy je obtížné útočníkovi se dostat ke znění dotazu
  - volba náhodného portu / ID
- **cache poisoning** = útočník legálně provozuje server pro nějakou doménu a donutí klienta, aby mu poslal DNS dotaz (např. vábivá reklama). Do sekce AUTHORITY a ADDITIONAL přidá falešné údaje o jiné doméně
  - řešení: rozšíření DNSSEC - podepisuje všechny záznamy domény klíčem. Je ale komplikované a rozšiřuje se pomalu

Diagnostika DNS

- program **nslookup**
  - příkazy pro nastavení serveru a dohledání jmén
  - lze volat i z příkazové řádky
- program dig (Linux)
  - `dig [@server] name [type_RR]`

4. přednáška

FTP (File Transfer Protocol)

- jeden z nejstarších protokolů
- anonymní přístup (user - anonymous / ftp, heslo - email)
- slouží pro přenos souborů z/na vzdálený počítač
- textový protokol - klient se přihlásí na port 21

Kódy odpovědi

- řídicí kanál - požadavky klienta a odpovědi serveru
- HTTP - 404

Číslice	význam
1xx	předběžná kladná odpověď (akce zahájena)
2xx	kladná odpověď (definitní)
3xx	neúplná kladná odpověď (jsou nutné další příkazy)
4xx	dočasná záporná odpověď (try again)
5xx	trvalá záporná odpověď (nepodařilo se)

Aktivní/pasivní datové spojení

- FTP používá dodatečné datové kanály
  - řídící spojení - posílají se požadavky klienta a odpovědi serveru
  - datové spojení - přenos veškerých dat
    - na každý nový přenos se otevře a uzavře nové TCP spojení
    - **aktivní** - navazuje server - port 20 ("ftp-data") - zahájení příkazem PORT
    - **pasivní** - navazuje klient - příkaz PASV
- server/klient potřebuje od druhé strany IP adresu a port
- použije se NAT (překlad adres + změna obsahu zpráv)

Aplikace pro FTP

- WWW prohlížeče
- správce souboru (Total Commander)
- příkaz `ftp`

SMTP (Elektronická pošta)

Elektronická pošta, SMTP (Simple Mail Transfer Protocol)

- existuje mimo internet - offline služba
- alias@domena / login@pocitac
- na Internetu - SMTP (přenos pošty)
  - **textový protokol** na TCP portu 25
  - pro přenos e-mailů
- ukázka SMTP
  - klient a server posílá jednotlivé příkazy jako textové řádky (včetně kódu odpovědi)
  - nový dopis - klient: MAIL FROM: <>, RCPT TO: <>
  - odpověď serveru = kód (250, 450, 550 | 354)
    - neúspěšné doručení: **DSN** (Delivery Status Notification) - generován MTA
  - DATA
  - klient posila text dopisu
  - QUIT

Příjem a odeslání pošty v SMTP

- SMTP *není vhodný* pro uložení velkých souborů
- dva způsoby:
  - A. přímé doručení - na cestě mezi klientem a cílovým serverem nestojí žádné překážky - navázaní SMTP spojení
  - B. doručení přes forwarder
    - proces "**mail-submission**": mail se předává pomocí SMTP serveru v lokální síti (=**mail-forwarder**)
    - **Mail Transfer Agent** = každý uzel, který přijímá (=server) a dále posílá poštu (=klient) pomocí SMTP
    - pokud doručení mailu posledním MTA není možné, zůstane ve frontě na tomto MTA
      - správce poštovního serveru zabrání tomu: *nastaví MX záznam v DNS*
        - MX obsahuje jméno Mail Exchangeru - dočasně přijímá poštu pro daný stroj/doménu, dle priorit

Přístup k poště z pohledu uživatele

- připojení z MUA (Mail User Agent) = poštovní program
  - A. přímé připojení
    - dopisy v mailboxu jsou ve frontě MTA
  - B. připojení přes POP/IMAP (čtení), SMTP (odesílání)

Elektronický dopis (E-mail)

- skládá se ze 2 částí:
  1. záhlaví: hlavičky, řádky s řídicími informacemi
  2. text dopisu: UTF-8 / UUENCODE (ESMTP), přílohy obsahující soubory

Hlavičky dopisu

- Cc: carbon copy - kopie adresátům
- Bcc: blind carbon copy - kopie tajným adresátům
  - poštovní program jej přidá do SMTP obálky, ale nikoliv do textu

Soubory a diakritika v poště

- původně: ASCII 7 bit, pak UUENCODE (vezmou se 24 bitů původního souborů, rozdělí se do 4 tisknutelných znaků)
  - UUENCODE: nevíme, jaké vložené soubory obsahuje dopis

MIME

- rozšíření **MIME (Multipurpose Internet Mail Extension)**
  - v řadě jiných protokolů, např. HTTP
  - dokument v MIME formátu:
    - je multipart (tělo je strukturované)
    - **typ dokumentu**, např. text/html
    - **znaková sada**: např. charset="UTF-8"
    - **způsob kódování**: quoted-printable
    - původní název souboru, předpokládaný způsob zpracování (např. zobrazit jako přílohu)
  - definuje 2 typy kódování:
    - **Base64**: vychází z UUENCODE - jiná tabulka (52 znaků (místo 26), číslice,atd.). formát řádek
    - **Quoted-Printable**: nonASCII znaky - "=HH" (hexadecimální hodnota)
      - do hlavičky možné vkládat nonASCII znaky - "=?"

Bezpečnost pošty

- **šifrování obsahu dopisu** - např. PGP (Pretty Good Privacy)
- nikdy není jistý odesílatel - řešení: **elektronický podpis**, systém výzva/odpověď
- **open-relay server** - dovolí komukoliv, aby se připojil a poslal dopis komukoliv - pro rozesílání hromadných (spam) mailů, zablokování jiných poštovních serverů
  - správně nastavený poštovní server rozlišuje dobré dopisy od špatných
- při prvotním vložení mailu, **mail-submission server** (MSA) požádá klienta, aby se autentikoval pomocí **AUTH** (příkaz ESMTP)
- příkaz STARTTLS - zahájení SSL/TLS spojení

Ochrana proti SPAMu

- **Gray-listing** = server udržuje databázi tripletů (klient, server, recipient). Napoprvé mail odmítne (odp. 450), opakované doručení už akceptuje
- **Sender Policy Framework** = algoritmus, při které doména ověří, jestli server má povolení na odesílání dopisu z dané domény
- **DomainKeys Identified Mail (DKIM)** = všechny odesílací MTA pro danou doménu možné vybavit vlastním klíčem. Příjímací server klíč zkontroluje a podle výsledku dopis -> do cílového mailboxu, nebo do spamu.
- **Antispam** = server na základě heuristiky odhaduje pravděpodobnost, že mail je spam
  - riziko false-positive

5. přednáška

	POP	IMAP
Doba	- starší	-mladší

	POP	IMAP
Dopisy	-nutné stahovat ze serveru celé	-server uchovává informace o nich
		-vyžaduje pouze část dopisů
		-je možné nechat v dopisu vyhledávat
Složky		-podpora více složek

POP (Post Office Protocol)

- starší protokol, pro přístup uživatelů k poštovní schránce
- port 110
- *nevýhody*:
  - otevřené posílání hesla
  - dopisy je nutné stahovat ze serveru celé - pak se smažou ze serveru
  - nelze pracovat se strukturou dokumentů
- nahrazen protokolem IMAP

IMAP (Internet Message Access Protocol)

- *výhody:*
  - šifrované spojení
  - server uchovává informace o dopisech, je možné v dopisech vyhledávat, vyžádá pouze část dopisu,
  - podpora vice schránek / složek
- **šifrování**
  - navazování spojení na port 993
  - příkaz STARTTLS
- IMAP je ve většině MUA
- protokol je textový, uživatelsky přívětivý

Princip distribuované databáze

- Gopher (1991) - první rozšířená služba, která umožňuje uživateli pomocí odkazů přecházet z jednoho serveru na jiný.
  - Servery uchovávají v sobě informace, tž. Gopher je databáze informací
- teď: Gemini, HTTP - přístup ke vzdáleným dokumentům

WWW

- =distribuovaná hypertextová databáze
- **základní jednotka** = hypertextový dokument (stránka) - v HTML
  - jsou statické (cesta v URL = relativní cesta na disku serveru)
  - dynamické - JavaScript (požadavky klienta)
- přenos stránek - HTTP

HTTP (Hypertext Transfer Protocol)

- zabezpečená verze: HTTP + SSL = **HTTPS**
- **klient** pošle textový požadavek

- server - kód odpovědi (200), MIME typ zaslaného dokumentu, požadovaný dokument v HTML

HTTP v.1

- port 80
- formát zprávy - klient:
  - úvodní řádka: metoda, cesta (část url), verze protokolu
  - hlavička: Host (na jaký server se klient obrácí)
- form odpovědi - server:
  - kód
    - 4xx - chyba na straně klienta
    - 5xx - chyba na straně serveru
  - číslo protokolu
  - slovní popis

Metody HTTP

Metoda	fce	Pozn.
GET	zobrazení web stránky/dokumentu.	Je idempotentní = opakované použití má stejný efekt
HEAD	hlavička (nikoliv dokument)	jestli je stránka dostupná, jak je velká atd.
POST	(parametry) => obnovený obsah dokumentu	posílá na server parametry
PUT	přepsat obsah dokumentu na serveru, který jsme poslali	idempotentní
DELETE	smázání dokumentu na serveru	
CONNECT	otevření spojení (tunelu)	

Vlastnosti HTTP v1

- odpověď na 1 požadavek = 1 dokument
- po jednom spojení může jít více požadavků (text, 3 obrázky)
  - požadavky jsou tedy nezávislé -> komunikace je bezestavová
  - 1 TCP-Spojení nemusíme zavírat
- stav = cookies
  - generované serverem na základě dat z logického dialogu (požádavky klienta) - pošle je v hlavičkách klientovi
  - prohlížeč je potom ukládá při dalších požadavcích
  - data z cookies = shromáždění informace o uživateli

HTTP v2

- větší propustnost dat
  - vlastní multiplexing - více streamů v rámci 1 TCP spojení (streamy se dají prioritizovat)
  - server pushne více dat, než klient potřebuje - např. reklamy ??
- binární protokol
- nemá ale šifrování -> HTTPS

Telnet (Telecommunication Network)

- protokol pro přihlašování na vzdálené zdroje, port 23
- také navazuje spojení na nějaký server pracující v jiném protokolu (HTTP, SMTP)
- **síťový virtuální terminál** (NVT): echo -> zobrazení znaku
- **nevýhoda**: otevřený přenos dat

Secure Shell (SSH)

- protokol pro vzdálené přihlašování a přenos souborů
  - aktuální verze 2, port 22
- komunikace je šifrovaná (asymetricky), ověříme vzdálený server
- SSHv2 umožňuje také
  - otevírat paralelně více zabezpečených kanálů
    - např. být přihlášen na virtuální stroj a současně přenášet soubory
  - SSHFS - zpřístupnit souborový systém
  - tunelování = komunikace z jedné strany SSH se přenáší kanálem a na druhé straně na nějaký tamější server

Bezpečnost SSH

- klient ověřuje server na základě: kontroly klíče (potvrzuje uživatel), nebo ověřeného certifikátu
- server ověřuje uživatele pomocí hesla / výzev a odpovědí (OTP) / veřejného klíče
- strategie používání klíčů
  - důkladně ověříme klíč serveru
  - přihlášení: místo hesla, privátní klíč s heslem
  - bez hesla, pokud nerecipročně (a->b, b->a) = ochrana proti červům

Voice over IP

- nástroj pro přenos hlasu pomocí TCP/IP sítě
- použití: H.323, SIP (standardy), aplikace (Skype)
- nevýhody:
  - digitalizace hlasu
  - propojení s běžnou telefonní síti
  - nalezení partnera

H.323

- standard pro multimediální komunikace
- zahrnuje **binární protokoly**, např. RTP kanály, RTCP
- základ: Abstract Syntax Notation 1
- nahrazen SIP

SIP (Session Initiation Protocol)

- jeho architektura se podobá HTTP, informace se přenášejí ve formě hlaviček
- používá TCP i UDP
- řeší jen signalizaci - vyhledá partnera a naváže spoj

    - **proxy** = část komunikačního kanálu, který usnadňuje komunikaci přes hranice různých sítí, včetně privátních
- zprávy SIP protokolů obsahují SDP (Session Descripton Protocol). Ten má podobu keyword=value
    - SDP řeší vlastnosti zařízení a parametry datových kanálů
- průběh SIP session:
    - INVITE příkaz - vyslaný volající zařízení. INVITE obsahuje volané URL a SDP zprávu (nabídka datových kanálů)
    - příkaz dorazí na nejbližší proxy - ten vyhledá další uzel v cestě k volanému zařízení. Na konci proxy zkontroluje obsah SDP a ho upraví (NAT)
    - 100 Trying
    - volané zařízení zpracuje požadavek - odešle dočasnou odpověď 100 Trying (proxi tu zprávu nepošle volajícímu)
    - 180 Ringing (proxy přepošle)
    - odpověď 200 OK - s SDP zprávou (nabídka datových kanálů na volaném zařízení) - proxy ho pak upraví
    - ukončení hovoru = BYE (200 OK)

6. přednáška

NFS (Network File System)

- připojení cizího disku k počítači a pracovat s ním, jako s lokálním diskem
- UDP (pro malou výměnu dat) a TCP*, port 2049
- **identifikace připojeného disku**: server:cesta
- **relační** (RPC - Remote Procedure Call) a **prezentační** (XDR - Exchange Data Representation) vrstva

SMB (Server Message Block)

- **identifikace připojeného disku**: \server\cesta
- **autentikace**: login a heslo

NTP (Network Time Protocol)

- **synchronizace času mezi uzly sítě** - klíčová vlastnost LAN
- UDP, port 123
- zdroj **stratum 0** - má absolutně přesné hodiny
- server stratum N: řízený podle zdroje stratum N-1
- problém: **odpovědi od severů mají různé zpoždění**
    - Marzullův algoritmus: nejlepší průnik intervalů (obrana před zacyklením)

BOOTP, DHCP

- umožňuje klientovi získat **IP adresu sítě** (do které se připojil), kterou smí používat, a další informace o lokální síti

BOOTP (Bootstrap Protocol)

- přiděluje IP adresu bezdiskovým stanicím, které nemají trvalé úložiště a nemůžou nikam uložit konfiguraci (včetně IP adresu)
- stanice pošle MAC adresu síťové karty, BOOTP server najde klienta (stanici) v seznamu a pošle IP adresu, a jméno

- pokud BOOTP server v **jiné síti**:
  - cílová IP adresa - **limited broadcast**
  - žádost se pošle **všem** uzlům, ale některé ji budou ignorovat
  - nebo routery je nepropouští mimo síť -> **BOOTP forwarding**= router přeposílá BOOTP dotazy ze sítě
- nahrazen DHCP

DHCP (Dynamic Host Configuration Protocol)

- dynamická alokace IP adres = jejich přepisování
- časově omezený pronájem IP adres
- možnost zapojení více serverů
- v MS Windows: "získat IP adresu automaticky"

Průběh DHCP

- Klient: DHCPDISCOVER (broadcastový požadavek)
- DHCP servery: DHCPOFFER (nabídky)
- Klient: posuzuje nabídky během timeout
- Klient: DHCPREQUEST (adresa, kterou si zvolil)
- Server: DHCPACK (potvrzení, že adresa je opravdu stále volná)
- *doba pronájmu*
- Klient: DHCPREQUEST (pouze zvolenému serveru, chce zjistit, zda má stále přístup k adrese)
  - **odpověď je**: nový interval pronájmu
  - **není**: DHCPREQUEST (broadcastem)
    - ne -> nová procedura

6. OSI vrstvy - pokračování

Prezentační vrstva (OSI 6)

- skryje rozdíly mezi reprezentacemi dat (kódování znaků, čísel atd.)
- problémy:
  - konce řádek: CRLF
  - pořadí bytů: BE, LE
- v TCP/IP konverze dat probíhá v aplikacích

Relační vrstva (OSI 5)

- řídí dialog mezi 2 aplikacemi
  - 1 dialog může obsahovat více TCP spojení (SIP - přenos audio/video dat)
  - 1 spojení - více dialogů (např. 1 SMTP spojení - několik mailů)

Transportní vrstva (OSI 4)

- end-to-end přenos dat
- umožňuje provozování více aplikací (klientů a serverů) na stejném uzlu sítě
- zprostředkovává služby sítě aplikačním protokolům
- TCP a UDP , doplnění dole
- TCP (**Transmission** Control Protocol)

    - klient - naváže spojení, data tečou ve formě proudu
    - TCP řídí a zabezpečuje spojení
    - méně pravidelné, ale spolehlivé doručování
- UDP (**User** Datagram Protocol)
    - neexistuje spojení, data se posílají jako nezávislé zprávy
    - aplikace (my) řeší průběh přenosu zpráv
    - pravidelné, za cenu vyšší ztrátovosti

Struktura UDP datagramu

- v hlavičce: informace u multiplexingu (zdrojový a cílový port, řídicí informace - délka a kontrolní součet)
- zapouzdření aplikačních dat do PDU OSI 4 je jednoduché

Struktura TCP paketu

- zdrojový, cílový port
- **Sequence number (Seq)** = identifikace paketu (každý obsahuje relativní posun (offset) vůči počátku streamu)
- **Acknowledgement number (ACK)** = potvrzení doručení paketu
- **Flags** / příznaky
- Urgent pointer - aplikace: označení určitých dat za **urgentní** příznakem URG (např. FTP klient chce přerušit datový přenos). Je to funkce *out-of-bend* přenosu
- **TCP okno** / window

TCP okno

- potvrzení doručení a řízení toku dat
- paket s příznakem ACK a hodnota Ack. number = offset konce dat, která byla doručena
- **velikost okna** = rozsah dat, které smí odesílatel odeslat, aniž by čekal na potvrzení

Zahájení a ukončení TCP spojení

Navázaní TCP-Spojení

- = "3-way handshake"
- na začátku spojení se odehrává dohoda pomocí **3 speciálních paketů** (prázdná datová část, v hlavičce - informace)
- klient a server se dohodnou na sekvenčních číslech
- offset začíná od náhodného čísla (ne 0)
- 1. paket - SYN (klient), 2. paket - SYN ACK (server), 3. paket - ACK (klient)

Uzavření spojení

- klient už nebude posílat žádná data
- 1. paket - FIN (klient), 2. paket - ACK (server)

TCP příznaky

TCP flag/packet	Vyznam
SYN	synchronizace čísel paketu

TCP flag/packet	Vyznam
ACK	potvrzuje doručení všech paketů
PSH	příjemce obdržel poslední segment bloku dat, má ho předat aplikaci (push)
FIN	odesílatel zavírá svou stranu spojení
RST	reset - odesílatel odmítá přijmout spojení
URG	paket obsahuje urgentní (out-of-band) data

- `tcpdump`

Výpis existujících socketů

- command `netstat` -a
- jaké spojení jsou teď na našem počítači otevřená
  - TCP: stav spojení (LISTENING server), otevrene spojeni(ESTABLISHED), adresa lokalniho socketu, adresa socketu protistrany
  - UDP: bezici se servery (neni zadne spojeni)

Síťová vrstva (OSI 3)

- přenáší data, které byla předána transportní vrstvou od zdroje k cíli
- základ vrstvy:
  - **adresace** - způsob, jak identifikovat jednotlivé uzly v síti tak, aby bylo možné rozpoznat, do jaké sítě patří
  - **směrování** - způsob, jak na základě adresy najít správnou cestu od zdroje k cíli (viz forwarding)
  - **enkapsulace** = řídicí data se musí vložit do PDU
  - **dekapsulace**: vybalení dat, a předání transportní vrstvě

IP (Internet Protokol)

- Vlastnosti
  - nespojovaná služba - datagramy se doručují nezávisle
  - nespolehlivá - negarantuje doručení
  - nezávislá na médiu (nezávisí na technologiích)
- Adresa
  - sítě
  - uzlu/počítače
- Přidělování
  - IANA (Internet Assigned Numbers Authority), spadá pod ICANN
  - region
  - lokální správa sítě

Struktura IPv4 datagramu

- Verze: polovina bajtu (u IPv16 jiný format)
- **Délka hlavičky / net**: 32 bitová slova - max. délka hlavičky: 60 bajtů.
- **Service Type:** QoS (priorita), zbytek prvního slova = délka celého datagramu
- Druhé slovo - **fragmentace** (subnet)

    - postup, kdy síťovou vrstva dostane paket, při jeho zapouzdření rámec bude delší než max. povolená délka pro danou linkovou vrstvu **(MTU, Maximum Transmission Unit)**
    - síťová vrstva musí fragmentovat paket na více datagramů => komplikace
    - **Path MTU** - metoda: pakety se posílají s příznakem *"Do not fragment"*
- Třetí slovo - **Time-to-live**, číslo protokolu, který je v datagramu zapouzdřen, kontrolní součet hlavičky
- poté následují **IP adresy odesílatele a příjemce**

IPv4 adresy

- **původně**: 1 byte
- teď: třídy A, B, C - 4 byty
    - **třída A**: původní dělení adresy 1:3 (síť : počítač)
    - **třida B:** 1:1 (2B : 2B)
    - **třida C**: 3:1
        - má max. 254 počítačů, 2 mil. sítí, ostatní dvě adresy mají vyhrazený význam
    - třida D: multicastová adresa (chybí část pro počítač - pro speciální služby, jako videokonference atd.)
    - třida E: experimentální adresa

Speciální IPv4 adresy

Dle Architektury IP

- **this host** = zdrojová adresa jako 0.0.0.0/8
    - kdy neznáme svou adresu, ale potřebujeme komunikovat
- **loopback** = adresa lokálního počítače (běží na něm klient a server)
- **adresa sítě** = adresa site . same nuly
- **network boradcast** = adresa site. same jednicky
    - chceme oslovit všechny počítače v dané sítii
- **limited broadcast** = 255. 255. 255. 255
    - zdrojová a cílová adresa jsou ve stejné síti

Dle definice (organizační rozhodnutí)

- **privátní adresy** - pro provoz v lokální síti, přiděluje správce, nesmí opustit síť (NAT). 1 síť A, 16 sítí B, 256 sítí C
- **link-local adresy** - počítač si vezme libovolnou adresu a zahájí konverzaci na segmentu sítě. Se pomocí nich nedá komunikovat mimo vlastní síť

Subnetting

- rozšíření síťové části adresy: **net subnet host**
    - musíme specifikovat obsah sítě
- z jedné sítě můžeme vytvořit několik podsítí
- **síťová maska**: jedničky - obsah adresy sítě
    - vyANDujeme síťovou masku (naší adresu) a dotycnou adresu -> zjistime, zda adresa patri do dane site
- ignoruji se tridy - pocitaji se bity prefixu
- pokud se v siti pouzivaji ruzne masky - sit s **variable length subnet mask (VLSM)**
- **supernetting** = posun hranice site opacnym smerem
- nezajímá nás subnetting v cizí sítl

IPv6 adresy

- **podoba adres**: 128 bitů (16 bytů)
- druhy adres:
  - unicastová - adresa jednoho uzlu
    - loopback, linkscope (linklocal), unique-local
  - multicastová - adresa skupiny uzlů (rozhraní) => 1 zpráva se pošle více uzlům
  - anycastová - unicastová adresa přidělená více uzlům
  - chybějí broadcastové (nahrazeny multicastovou)

7. přednáška

Směrování v síti

- next-hope router = **gateway**
  - používá směrovací tabulku, která je pod kontrolou OS
- cíl = **destination**
  - je to adresa sítě včetně jejího rozsahu zadaného **síťovou maskou**
- defaultní záznam:
  - destination: default (router)
  - geteway: 1.0.0.1

Příklad směrovací tabulky

- **první záznam**: síťové rozhraní s loopback adresou - gateaway: naší vlastní adresa, kterou jsme připojeni do sítě
- **druhý záznam:** hlavní síť LAN s adresou třídy C
- **třetí záznam:** záznam pro point-to-point síť, kterou jsme připojeni k ISP routeru - host s maskou /32
- nepřímý záznam - ukazuje na jednu z podsítí v naší LAN
  - gateaway - vnitřní router v naší síti
- default - veškerý ostatní provoz směřuje na ISP router

Principy směrování / Záznamy

- každá stanice v TCP/IP by měla umět směrování (paketů)
- záznam: **cíl maska gateaway**
  - **cil** - adresa site
  - **maska** - rozsah teto site
  - **gateway** - next-hope router (do cizí sítě - záznam nepřímý), nebo adresa místní sítě (záznam přímý)
- maska: uvažovaná část adresy sítě
- typy záznamu:
  - direct (gateaway - vlastní adresa)
  - indirect, default
- vznik záznamu:
  - implicitní - automaticky (po nakonfigurování síťového rozhraní)
  - explicitní - ručně zadaný (příkazem)
  - dynamický - pomocí informací od dalších uzlů v síti

Směrovací algoritmus

- najdi ve směrovací tabulce všechny vyhovující záznamy. Existuje?
  - ne - není cesta (pokud neobsahuje defaultní záznam)
  - ano - zvol záznam podle nejširší masky/nejdelšího prefixu. Můj počítač?
    - ano
    - ne - moje síť?
      - ano - poslat příjemci (nějaký server)
      - ne - poslat směrovači (opakuj)

ICMP (Internet Control Message Protocol)

- posílá řídicí informace pro IP. Zprávy:
  - **ICMP Echo, ICMP Echo reply**
    - testování dosažitelnosti počítače (program ping)
  - **Destination Unreachable**
    - router nemá jak doručit paket, a ho zahazuje (po neúspěšném vyhledání směrovacích tabulek)
  - Time Exceeded
    - vypršel Time-To-Live (chyba v routování)
  - Source Quench
    - žádost o snížení rychlosti toku diagramů
  - Redirect
    - změní záznam v routovací tabulce
  - Parameter problem
    - chyba v zahlaví datagramu
  - Router Solicitation
    - vyhledávání routerů
- používá IP datagramy, není v OSI 4

Ping

- program, který slouží k diagnostice sítě
- zavoláme: vysílá ICMP Echo
- cílový zdroj: ICMP Echo reply
- PING url vypíše:
  - Time-To-Live
  - počet bytů z cílové IP adresy (url)
- 3 paketů přeneseny, 3 pakety přijaty, ztrata 0%

Time-to-live

- prostředek, který zabrání zacyklení v případě chyby v routovacích tabulkách
- vyjadřuje počet **routerů/hopů**, které smějí paket forwardovat
- každý router sníží hodnotu TTL a změní obsah IP hlavičky, přepočítá kontrolní součet
  - pokud hodnota = 0 => router paket zahodí, vyšle ICMP Time Exceeded

Diagnostika směrování

- netstat -r[n] , route pint
- ping (nepomůže) - směrovací tabulky fungují jednosměrně

- traceroute (TTL)

Statické řízení směrovacích tabulek

- statická metoda řízení tabulky = počítač má uložené informace pro všechny záznamy
- vhodné pro jednodušší, stabilní sítě
  - připojíme se s počítačem do sítě a on přes DHCP dostane adresu defaultního routeru pro LAN

Redirekce

- pro složitější, rozsáhlejší sítě
- ICMP Redirect - když chceme poslat paket z routeru (první záznam) zpatky do stejné sítě, pak do jiného routeru (který ho přepošle cílové síti), vytvoříme nový záznam s příznakem D

Dynamické řízení směrovacích tabulek

- routery si navzájem vyměňují informace o síti (pomocí routovacích protokolů)
- routovací protokoly (musí běžet na uzlu):
  - BIRD (MFF), routed, gated
- nevýhody:
  - citlivější na útoky / chyby software
  - zátěž zprávami routovacích protokolů

Routovací protokoly

Distance vector protokoly

- router má u záznamů v směrovací tabulce "vzdálenosti"
- svou tabulku často posílá sousedním routerům, ti si upraví svou tabulku => na konci tabulky (matice) budou stejné
- **výhody:** jednoduché, snadná implementace
- **nevýhody:** pomalá reakce na chyby, omezený rozsah sítě, chyba ve výpočtu jednoho routeru ovlivňuje celou síť

RIP (Routing Information Protocol)

- nejstarší směrovací protokol
- vzdálenost/metrika = počet routerů/hopů v cestě k cíli
  - omezen na 15 routerů
- pro výpočet nejkratších cest: **Bellman Fordův Algoritmus**
- nepoužívá se pro velké, dynamické sítě

Metrika a kvalita linek

- routery posílají **updaty** do svých tabulek
- např. update pošle router A -> jeho tabulka dorazí na oba routery B a C, upraví vzdálenosti do cílových sítí-> pak update pošle router B -> psk C
- Kvalita linek: dle šířky pásma (rychlostí)

Counting to infinity

- **race condition** = situace, kdy dva nezávislé jevy mohou nastat v nesprávném pořadí
- např. dojde k výpadku připojení do nějaké sítě na routeru. Výměnou update paketů se postupně metrika zvyšuje, a nezastaví se
- řešení:
  - split horizon - router neposílá informace o sítích routerovi, od níchž jej převzal
  - tirggered updates - jenom sníží metriku, neodstraní

Link state protokoly

- každý router místo tabulek posílá informace o stavech svých linek
  - si pamatuje tzv. "mapu sítě"
- výhody:
  - každý počítá sám za sebe: chyby neovlivní nikoho ze sousedů, na rozdíl od předchozích protokolů
  - pružná reakce na změnu síťové topologie
  - síť je možné rozdělit na menší podsítě
  - výměna dat probíhá pouze při změnách

OSPF (Open Shortest Path First)

- představitel link-state protokolu
- Dijkstrův algoritmus - nalezení nejkratší cesty
- výpočet vždy probíhá jen v rámci oblasti, na menší množině uzlů
- síť má páteř (oblast 0), ostatní oblasti se připojují pouze na páteř

Autonomní systémy

- jednotlivé bloky sítí tvoří **autonomní systémy (AS)**
  - skupina směrovačů se společnou směrovací politkou
- rozděluje routovací protokoly na
  - interní
  - externí - např. BGP (Border Gateway Protocol)
    - pracují s path-vector, posloupnosti cisel AS, pres nez cesta vede -> brani vzniku smycek

IP filtrovani

- směrovač, který připojuje LAN do internetu, je bod, který provádí IP filtrování (bezpečnost sítě)
- filtrování probíhá v OSI 4
  - TCP pakety musí chodit oběma směry
  - dobré pro protokoly s jedním kanálem (HTTP, SMTP)
  - s více kanály (FTP, SIP): filtr musí spolupracovat s aplikační vrstvou

Proxy server

- software, který kontroluje provoz určitého protokolu
- odděluje LAN od internetu
- provoz transparentně:
  - sw na routeru zachytí klientův požadavek, předá ho proxy serveru, ten ho zkontroluje podle bezpečnostních pravidel, navaže jako klient spojení na skutečný server -> ten po zkontrolovani (antivirem) odesle odpoved klientovi

- provoz netransparentně:
  - klienta musíme nakonfigurovat, aby se požadavky posílaly proxi-serveru v lokální sítí (ten nemusí byt router) -> nutná podpora protokolu

8. přednáška

Address Resolution Protocol

- konverze MAC (Ethernet) a IP (síťových) adres
- pokud cílovou MAC adresu nezná, použije **broadcastovou MAC adresu** (FF:FF:FF:FF:FF:FF)
- ARP server (hledaný uzel) na dotaz zareaguje unicastovou odpovědí (přidá informace o klientovi do tabulky)
- ARP klient si přiřazení IP a MAC adresy uloží do **ARP cache**
  - je tam více IP adres se stejnou MAC adresou (protože v síti je Proxy ARP)
- problém: chybějící zabezpečení (jakýkoliv uzel v síti může odpovědět)
  - např. nevyžádaná ARP zpráva (gratuitous ARP)
    - pokud v síti není Proxy ARP
  - řešení: některým serverům/routerům zakážeme používat ARP protokol

Proxy ARP

- server A pošle ARP dotaz serveru B, který se však nachází v jiné podsítí (podle masky jsou ve stejné síti). Na routeru spustíme ARP proxy, pošle serveru A MAC adresu routeru. Server A uloží MAC adresu do ARP cache. Pro další komunikaci s počítačem B použije MAC adresu routru

Linková vrstva (OSI 2)

- 2 podvrstvy
  - **Horní**: Logical Link Control - umožňuje různým protokolům OSI 3 (IP) přístup ke stejnému médiu (=**multiplexing**), kde se uloží jejích data
  - **Spodní**: Media Access Control (LAC) - řídí adresaci uzlů a přístup k fyzickému médiu v rámci linkové sítě
- množina uzlů sdělující stejné médium
- PDU na linkové vrstvě: **rámec(frame)**
  - **Synchronizační pole** - sekvence bitů, které odliší data od šumu. Teď pole se už nepočítá do obsahu rámce
  - **Hlavička** - MAC adresa klienta/servera, řídící informace LLC
  - **Data** (payload)
  - **Patička** - hodnota, která slouží ke kontrole správnosti doručení

Typy síťových topologií

zásuvka = port

1. Multipoint (více uzlů)
   - sběrnice (např. Ethernet) - koaxiální kabel
   - hvězda - strukturovaná kabeláž, UTP kabelem, sběrnicová
   - kruh - FDDI, Token-ring
2. Point-to-point (pár uzlů)
   - přímé propojení kabelem - Ethernet
   - modemem
   - bezdrátové propojení - na báti laserů a rádiových vln.

Způsob řízení přístupu uzlů k médiu

- Multipoint:
  - Deterministický - určuje, kdy uzel může vysílat
    - Token-ring: pokud uzel chce vysílat, počká na **token** (řídicí prvek v síti) a pošle data příjemci. Příjemce datový paket odstraní a pošle do sítě token
  - Nedeterministický - nastanou kolize
- Point-To-Point:
  - half duplex - pokud uzel neumí současně přijímat i vysílat, nastanou kolize
  - full duplex - pokud naopak, nebudou nastávat kolize

Řešení kolizí

- **CSMA (Carrier Sense With Multiple Access)**: uzel zkontroluje, jestli na nosné (přenosovém médiu) probíhá nějaký přenos, počká, až nebude volno
- **CSMA/CD (Collision Detection), např. Ethernet**
  - během vysílání uzel současně detekuje připadnou kolizi
  - při kolizi: zastavení vysílání -> prodlužování intervalu čekání
- CSMA/CA (Collision Avoidance), např. WiFi
  - když je volná nosná, vysílá se celý rámec a čeká se na ACK -> jinak exponenciální čekání

Ethernet

- technologie pro lokální sítě
  - je v hw počítače - Ethernet Adapter
- řízení přístupu: **CSMA/CD** metoda
  - při detekci kolize vysílá **jam signál** -> (stanice zastaví vysílání, upozorní ostatní, počká určitou dobu a pokus opakuje)
  - exponenciální čekání končí chybu
- adresy
  - 3 byty prefix výrobce, 3 byty vlastní číslo síťové karty

Struktura ethernetového rámce

- destination, source mac address, type: IP

Virtuální sítě (VLAN)

- provozuje po jedné fyzické síti více nezávislých lokálních sítí
- sítě - 12 bitový indentifikátor (VLANID)
- rámec se prodlouží o tag dlouhý 32 bitů => typ rámce = VLAN (tato operace se odehraje transparentně)
  - 0x8100
  - QoS priorita
  - VLANID

Cyklický kontrolní součet (CRC)

- hashovací funkce, která kontroluje konzistenci dat např. FCS - Frame Check Sequence v IP hlavičce
- myšlenka:

    - posloupnost bitů -> polynom s binárními koeficienty
    - tento polynom má tolik stupňů, kolik bitů má kontrolní pole
    - polynom se vydělí charakteristickým polynomem, s pevnou velikosti
    - zbytek po dělení se převede zpět na bity a použije se jako hash

WiFi

- WLAN (Wireless LAN) - bezdrátová síť
- CSMA/CA, hvězdicová topologie sítě (ve středu - AP access pointy)
- problém: Bezpečnost
- SSID (Service Set Identifier) - slouží k rozlišení WiFi sítí

Fyzická vrstva (OSI 1)

- přenos datového signálu po konkrétnám médiu
- převod digitální informace (1001010...) na analogovou (elektrické pulzy, rádiové vlny)
- typy médií
    - metalické kabely: elektrické pulzy
    - optické kabely: světelné pulzy
    - bezdrátové: modulace radiových vln

Druhy přenosu dat

- Analogový, digitální
    - **digitální**: zda hodnota signálu spadá do intervalu
    - D->A: modem
    - A->D: codec
- Baseband, Broadband
    - baseband - přenáší signál a kóduje ho (používá hodnotu signálu)
    - broadband - přenáší signál v širokém pásmu a moduluje ho (jeho frekvenci, nebo změnu amplitudy)

Nestíněná kroucená dvoulinka (UTP)

- pro připojení stanic v lokální síti
- 8 vodičů !!! (4 páry měděných pravidelně zakroucených)
- Ethernet používá jen 2 páry
- STP (Shielded Twisted Pair) - má kovové zastínění
- kabel může být
    - přímý
    - křížený

Optická vlákna

- signál se šíří jako viditelné světlo
- vysoká frekvence, velká šířka pásma (rychlost)
- nevýhoda - dražší, náročnější ohyb
- 2 druhy:
    - Jednovidová / singlermode - svít s laserem, přenosy na velké vzdálenosti

- Mnohovidová / multimode - LED diody, páteřní rozvody LAN

Segmentace sítě

- Repeater - spojuje stanice daleké od centra
  - větší dosah
  - neřeší propustnost (zhoršuje počet kolizí)
  - ve strukturované kabeláži: hub
    - strukturovaná kabeláž = označení metalických a optických prvků, umožňující propojení uzlů v síti
- Bridge - spojuje segmenty na linkové vrstvě
  - řeší: větší propustnost (rozděluje kolizní doménu)
  - ve strukturované kabeláži: switch
- centrální router a centrální síť jsou propojeny full-duplex linkou, která je maximálně propustná

Learning bridge, BUM (BUS)

- switche si pro každý port udržují **tabulku MAC adres stanic**, které jsou za tímto portem připojené
- **BUS/BUM** = switch pošle všechny rámce do správných portů, s výjimkou broadcastů/neznámých unicastů/multicastů
- pokud stanice B leží na stejném segmentu, proto switch nemusí rámec už nikam posílat
- pokud stanice C ne, switch ho přepošle pouze na port, kam patří

Spanning Tree Algoritmus

- přidáním druhého switche se vytvoří kružnice / cyklus
- náprava: najít kostru **grafu (=sítě)**
- switche si musí dohodnout, který bude forwardovat, a který bude monitorovat provoz druhého (jestli dojde k výpadku druhého switche)
- STP (Spanning Tree Protocol) vypustí hranu - převeden některé porty switche do režimu **blocking**
  - start portů je pomalý
  - má nezbytné timeouty