8. Počítačové sítě, ISO/OSI model, vlastnosti fyzických vrstev, topologie, řízení přístupu k médiu, kódování, spolehlivost datových přenosů, protokoly rodiny TCP/IP

https://moodle.fel.cvut.cz/enrol/index.php?id=3839

Počítačové sítě

Historie internetu

- 1969 Arpanet
- 1982 TCP/IP
- 1991 WWW

ISO/OSI model (International Organization for Standardization/Open Systems Interconnection model)

ISO OSI model je rozdělený na vrstvy. Teorie je taková, že v každé vrstvě se řeší něco jiného a není potřeba znát fungování jiných vrstev. Třeba emailový klient nepozná, jestli mu data přišla přes wifi nebo kabelem. Data vytvořená v aplikační vrstvě (packet) se obalí hlavičkou nižší vrstvy, postupně se uzavírá do dalších a dalších hlaviček, poslední je hlavička ethernetového rámce. Zařízení se vždy dívají jen na hlavičku z vlastní vrstvy - třeba router posílá packety dál podle IP adresy a je mu jedno, jestli to je TCP nebo UDP.

Vrstva	Vrstva anglicky	Protokoly	Název v TCP/IP	Zařízení	Pomůcka
Aplikační	Application	FTP, HTTP, SSH	Aplikační		All
Prezentační	Presentation	CSS, HTML, GIF	x		People
Relační	Session	SQL, SSL, PAP	x		Sleeping/Resting
Transportní	Transport	TCP, UDP	Trasportní		Through
Síťová	Network	IP	Síťová	Router	Networking
Spojová	Link	MAC	Spojová	Switch	Lectures
Fyzická	Physical	Ethernet	Spojová	Hub	Fail

Table 0.1: Vrstvy ISO/OSI modelu

V praxi to ale tak vždy není. Třeba FTP posílá IP adresy uvnitř svého packetu, takže když prochází FTP packet routerem na hranici sítě (bránou), ve kterém se adresy překládají (NAT níže), je potřeba packet rozbalit a upravit IP adresy i v něm.
TCP/IP je obecnější model, který má vrstvy jen čtyři.

Protokoly

Ethernet

- Na fyzické úrovni je nutné řešit, aby nevysílalo víc zařízení najednou, jinak se signál zaruší a nelze ho poslouchat (tomu se říká kolize)
- Kolizím lze předcházet, třeba tím, že každý připojený stroj má přidělený čas k vysílání (tak to řešil protokol token ring)
- Ethernet kolizím nepředchází, ethernet je detekuje. Každé zařízení během vysílání poslouchá, jestli nevysílá někdo jiný. Pokud ano, obě zařízení přestanou hned vysílat a budou opakovat vysílání po náhodné době.
- V ethernetu se data ověřují pomocí Cyclic Redundancy Check (CRC)

ARP (Address Resolution Protocol)

- Převádí IP adresu na MAC adresu
- Odesílatel pošle požadavek “Who has 192.168.37.178”, aby zjistil, kdo má danou IP. Dostane odpověď a MAC adresu si uloží do tabulky pro příště.

IP

- Slouží k adresaci strojů v síti

- IPv4: Čtyři čísla po 8 bitech, celkem 32 bitů, má 4 miliardy adres, ale dnes už nestačí
- Adresy jsou rozděleny do menších sítí, ty se udávají maskou
- Maska v IP verzi 4 označuje, kolik bitů na začátku adresy má celá síť stejně, ostatní bity identifikují zařízení uvnitř sítě (např. 192.168.0.0/16)
- IPv6: Osm čísel po 16 bitech, celkem 128 bitů, tj 3*10^38 adres, to už by stačit mělo

NAT

IP adresy uvnitř sítě (třeba 192.168.*.*) nejsou v celém internetu unikátní, taky se jim říká neveřejné. Routery v internetu packety z neveřejných adres mažou. Aby mohly počítače z vnitřních sítí komunikovat se serverem v internetu, musí jim být přidělena veřejná IP adresa, na kterou jim pak server pošle odpověď. Překlad neveřejných IP adres na veřejné obvykle zajišťuje hraniční router sítě (brána) protokolem Network Address Translation (NAT) s pomocí tabulky. Takový router (většinou patří poskytovateli internetu) má několik veřejných IP adres, a když přijde z vnitřní sítě požadavek něco poslat do internetu, uloží si do tabulky neveřejnou IP počítače a k tomu přidělí veřejnou IP. Jakmile přijde odpověď, podívá se do tabulky, aby zjistil, kterému počítači má packet poslat. Při pozdější komunikaci může být veřejná adresa přidělená počítači jiná. Pokud je adres málo, lze pracovat i s porty.

UDP

User Datagram Protocol slouží k posílání packetů přes síť. UDP neumí garantovat, že všechny packety budou doručeny, a že budou doručeny ve správném pořadí. Občas se prostě stane, že se nějaký packet úplně ztratí. UDP se používá u služeb, kde je prioritou čas, a není důležité mít kompletní informace - třeba u online streamů.

TCP

Transmission Control Protocol také řídí posílání dat, ale garantuje, že dorazí v pořádku. Spojení je navázáno “handshakem”, kdy nejprve stroj A vyšle žádost o spojení (SYN), spojení je potvrzeno strojem B (SYN ACK), a stroj A potvrdí, že bylo spojení úspěšně navázáno (ACK). Každý packet má své číslo, a přijemce musí potvrdit, že packet obdržel odesláním zprávy ACK. Pokud přijetí nepotvrdí (nebo se ACK ztratí), tak bude packet odeslán znovu. TCP se používá u služeb, kde chceme zaručeně správné a kompletní informace a nepotřebujeme je okamžitě - třeba email.

ICMP

Internet control message protocol je nástroj, který informuje o stavu sítě. Jeho součástí je třeba ping nebo traceroute. Zprávy “Destination unreachable” nebo “Time limit exceeded” jsou taky z ICMP - odesílá je router, když maže packet, který se pohyboval internetem moc dlouho. Stáří IP packetu (Time To Live - TTL v jeho

hlavičce) se počítá podle počtu navštívených routerů, číslo se postupně snižuje, na nule je packet smazán, protože zřejmě zabloudil.

HTTP

HyperText Transfer Protocol funguje na principu klient server. Klient posílá serveru požadavky, třeba GET, POST, PUT, DELETE, PATCH server je vykonává a odpovídá třímístným číselným kódem:

- 1..: Informativní
- 2..: Vše proběhlo ok (200 OK)
- 3..: Přesměrování (301 MOVED PERMANENTLY)
- 4..: Chyba na straně klienta (403 forbidden, 404 not found, 401 unauthorized)
- 5..: Chyba na straně serveru (500 internal server error, 503 unavailable)

SMTP

Simple mail transfer protocol, stará se o emaily.

A: MAIL FROM <...@...>
B: ok
A: RCPT TO <...@...>
B: ok
A: DATA
B: ok, ukonči data pomocí \n . \n
A: ..... \n . \n
B: ok
A: QUIT
B: bye

DNS

Domain name system překládá doménová jména (google.com) na IP adresy. Domény mají různé řády, třeba v cyber.felk.cvut.cz je cz doména prvního řádu a cyber doména čtvrtého řádu. Domény se registrují u autoritativního serveru. Při požadavku na identifikaci domény se nejprve server podívá do své paměti, a pokud doménu nemá, ptá se dalších serverů. Autoritativní servery sdílí své domény ostatním serverům společně s jejich dobou platnosti.

Typy DNS záznamu

- A (Host address)
- AAAA (IPv6 host address)
- ALIAS (Auto resolved alias)
- CNAME (Canonical name for an alias)
- MX (Mail eXchange)
- NS (Name Server)

- TXT (text - používá se napíklad, když po vás někdo chce verifikace že doména je vaše, máte za úkol dát nějakou dohodnutou hodnotu do DNS záznamu - analogie: jestli je ten FB account tvůj, napiš si na zeď 123456)

Sockety

Socket je způsob komunikace mezi procesy, v rámci pc i přes síť. Jeví se jako soubor a zapisuje se do něj jako do ostatních souborů v POSIX (read, write).

Bezpečnost

Bezpečnost na síti se řeší dvěma způsoby

- Symetrické klíče: Odesílatel má symetrický klíč, tímto klíčem zašifruje zprávu. Příjemce má stejný klíč a pomocí něj zprávu přečte. Je to výpočetně jednodušší než asymetrické klíče, ale je potřeba vyřešit bezpečné předání klíčů. Taky nejde vůbec poznat, kdo zprávu zašifroval. Jakmile klíč unikne, lze pomocí něj přečíst všechny zprávy.
- Asymetrické klíče: Klíč má dvě části, veřejnou a soukromou. Veřejná část je volně přístupná, a je výpočetně nemožné zjistit z ní soukromý klíč. Server, řekněme banka, má u sebe uložený svůj soukromý klíč. Ten dokáže přečíst všechny zprávy, které byly zašifrovány odpovídajícím veřejným klíčem. Veřejný klíč je registrovaný u certifikační autority danou bankou, certifikační autorita potvrzuje, že klíč opravdu patří bance. Klienti posílají svá data bance zašifrovaná pomocí veřejného klíče, a jediná banka je dokáže pomocí svého soukromého klíče číst. Je ale důležité, aby certifikační autorita (klidně i lokální databáze certifikátů) byla důvěryhodná - viz chyba Superfish u Lenova, hodně hezky to vysvětluje computerphile na youtube.
- Podepisování: Ověření pravosti dat. Můžete si to představit jako šifrování naopak, jen tomu nikde tak moc neříkejte. Chceme-li zajistit, že s daty(dokumentem) nebylo nijak manipulování nebo ověřit jejich původce, využijeme právě podpis. Obvyklý postup je vytvoření hashe dokumentu a následně zašifrování pomocí privátního klíče (ten může být součástí certifikátu). Osoba, která chce ověří pravost dokumentu, dešifruje hash pomocí veřejného klíče, spočítá nový hash dokumentu a porovná. Je vhodné mít různý certifikát pro podpis dokumentů a pro ověření identity. Pokud máte stejný, tak při ověřování identity pomocí certifikátu (podepisujete náhodná data), můžete nevědomky podepsat nějaký hash dokumentu a někdo vás tak klamem přinutí k podpisu nějakého dokumentu.

SSL a TLS

Transport Layer Security je novější verze SSL. SSL už je deprecated. Používá symetrickou kryptografii, ale k bezpečnému předání klíčů používá asymetrické klíče (nejčastěji DH key exchange). TLS se používá v HTTPS (bezpečné verzi HTTP), komunikace mezi klientem (K) a serverem (S) se navazuje takto:
K: posílá úvodní packet se seznamem podporovaných šifer a verzí TLS
S: odpovídá, posílá šifru a verzi TLS, která se bude používat a svůj veřejný klíč
K: ověří veřejný klíč u certifikační autority. Pokud je platný, tak vygeneruje symetrický klíč (ve skutečnosti vygeneruje Master secret, to je složitější), ten zašifruje pomocí veřejného klíče a pošle serveru
S: přečte zprávu pomocí svého soukromého klíče, tím získal symetrický klíč. Teď mají obě strany symetrický klíč a mohou komunikovat.
Důležitý poznatek: Zejména při používání free nebo sdílených certifikátů, zelený zámeček (HTTPS) neznamená nutně bezpečí. Znamená to jen, že komunikace je šifrovaná a po cestě ji nelze odposlechnout. Nicméně na druhé straně může být klidně satan.

RSA

Asymetrická šifra, která se používá třeba u SSH. Máme náhodná hodně velká prvočísla p a q, a číslo $N = p \times q$, které je veřejně známé. Najdeme e, které je nesoudělné s $\varphi(N) = (p-1)(q-1)$. Najdeme d, aby $e \times d = 1 mod \varphi(N)$. Veřejný klíč je (N, e), soukromý klíč je (N, d). Zprávu Z zašifrujeme $C = Z^e mod N$, a dešifrujeme $Z = C^d mod N$.

Diffie-Hellmann

Dá se použít k vytvoření symetrického klíče mezi dvěma stranami (Alicí a Bobem) bez pomoci certifikační autority. Máme veřejná čísla p, q, a tajná čísla a od Alice a b od Boba. Alice pošle Bobovi q^a mod p, a Bob pošle Alici q^b mod p. Z toho mohou oba spočítat klíč K = g^(a*b) mod p, ale po odposlechnutí komunikace nelze kód zjistit.

Man in the middle

Man in the middle (MIM) je útok, během kterého útočník přeruší komunikaci mezi oběma stranami a vede ji přes sebe, u toho pak mění obsah komunikace. V příkladu Diffie-Hellmannovy výměny klíčů může MIM vygenerovat vlastní číslo x, a komunikovat s Alicí pomocí klíče g^(a*x) a s Bobem pomocí klíče g^(x*b), uprostřed zprávy rozbalí, přečte a zašifruje pomocí druhého klíče. Ani jeden z komunikujících se nemůže dozvědět, že komunikace byla narušena. Lze se chránit například využitím VPN nebo obecně předem dohodnutým a bezpečně předaným symetrický klíčem.

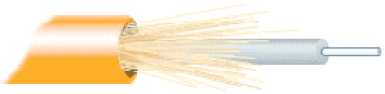
Vlastnosti fyzických vrstev

Přenosová média

- Metalické vedení
  - Koaxiální kabel
    - obtížnější instalace, nesnáší ostré ohyby
    - obtížnější připojování
    - dražší cena
    - nižší útlum na jednotku délky
    - vysoká odolnost vůči elektromagnetickému rušení
  - Kroucený dvoudrát (twisted pair)
- Optické vedení
  - odraz světla na rozhraní dvou prostředí s odlišným indexem lomu
  - standardně pouze pro spojení bod-bod
  - **vysoká přenosová kapacita**
  - vysoká odolnost vůči elektromagnetickému rušení
  - obtížný odposlech
  - obtížné spojování

Koax

Twisted pair



Optické vlákno

Koax

Twisted pair

Oops! We couldn't...

It may have been moved ...

https://www.hdt.cz/fotocache/bigorig/50520200.JPG

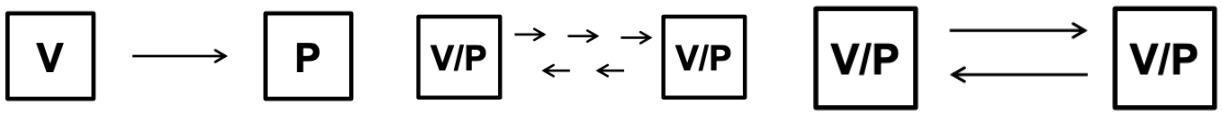
Retry

Optické vlákno

Typy datových přenosů

Podle směru přenosu

- jednosměrný
  - simplex
- obousměrný střídavý
  - poloviční duplex
  - half-duplex
- obousměrný současný
  - plný duplex
  - full-duplex



simplex *half-duplex* *full-duplex*

Podle počtu současně využitých kanálů

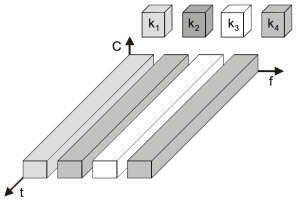
- sériový
- paralelní

Podle způsobu synchronizace

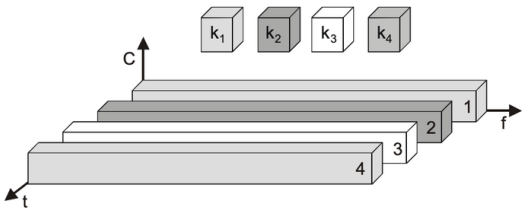
- paralelní synchronní přenos
- paralelní asynchronní přenos
- sériový asynchronní (arytmický) přenos
- sériový synchronní přenos
  - buď vyhrazený kanál pro přenos hodin
  - častěji je synchronizační signál zakódován přímo do datové posloupnosti (kanálová kódování – např. Manchester)

Sdílení kapacity kanálu

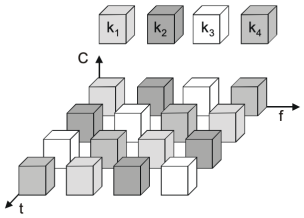
- Frekvenční multiplex - FDM
- Časový multiplex - TDM
- Kombinovaný časový a frekvenční multiplex



Frekvenční multiplex



Časový multiplex



Kombinovaný časový a frekvenční multiplex

Přepínání okruhů X přepínání paketů

Přepínání okruhů	**Přepínání paketů**
• mezi příjemcem a odesílatelem vzniká přímá souvislá cesta - “roura” • komunikace probíhá v reálném čase • data se nikam neukládají	• data od odesílatele k příjemci cestují přes “přestupní body” (podle principu “store & forward”) • není v reálném čase
Např.: klasická telefonie, TV, streamování	typické pro PC sítě

Topologie

Kruh

Data musí projít všechny uzly mezi odesílatelem a příjemcem. Výpadek jednoho uzlu ochromí celou síť.

Hvězda

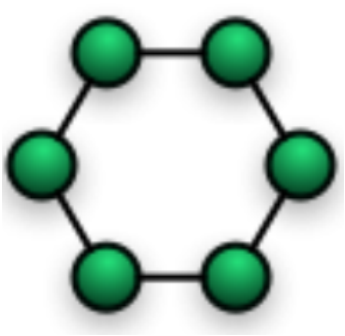
Každý počítač je připojený pomocí kabelu k centrálnímu prvku - hubu nebo switchi. Mezi každými dvěma stanicemi existuje vždy jen jedna cesta. To znamená, že selhání jedné stanice neomezí provoz sítě, ovšem kolaps centrálního prvku znamená kolaps i pro celou síť.

Strom

Vychází z hvězdicové topologie spojením aktivních síťových prvků, které jsou v centrech jednotlivých hvězd. V případě, že selže jeden síťový prvek, výpadek ovlivní pouze část sítě pod něj spadající. Ostatní části sítě ale mohou dále pracovat.

Sběrnice

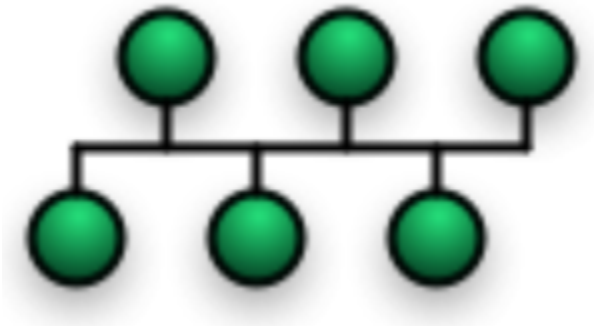
Spojení zprostředkovává jediné přenosové médium (sběrnice), ke kterému jsou připojeny všechny uzly sítě. Má nízké pořizovací náklady, ale omezenou rychlost přenosu a také v ní může docházet ke kolizím. Je vhodná spíše pro malé a dočasné sítě.



Kruh

Hvězda

Strom



Sběrnice

Řízení přístupu k médiu

Deterministický princip

Kolize vůbec nenastávají.

Master-Slave

- vyhrazený uzel Master se dotazuje uzlu typu Slave
- Slave nesmí samostatně vysílat, komunikace probíhá pouze přes Mastera
- např.: průmyslové distribuované systémy

-	+

závislost na výpadku Mastera	jednoduchá implementace

Token Passing

- jednotlivé uzly jsou rovnocenné
- oprávnění k vysílání má držitel pověření (tokenu), které si uzly v kruhu předávají mezi sebou
- držení tokenu je pro jeden uzel časově omezeno

-	+
dlouhý čas na zformování kruhu na začátku nebo při ztratě tokenu	nezávislost na jednom uzlu

TDMA (Time Division Multiple Access)

- Umožňuje více uživatelům sdílet stejný frekvenční kanál dělením signálu do různých časových slotů
- Uživatelé vysílají v rychlém sledu za sebou, jeden po druhém, každý používá svůj vlastní časový slot. To umožňuje více stanicím sdílet stejné přenosové médium (např. kanál rádiových frekvencí), při využití pouze části kapacity kanálu.

Delegated Token

- existuje vyhrazený uzel (arbitr), který vysílá speciální výzvu umožňující ostatním uzlům vysílat
- díky adresaci zpráv mohou všechny uzly přijímat současně
- nevýhodou je závislost na uzlu arbitra

Nedeterministický princip

Kolize nastávají a protokol s nimi počítá.

Protokol CSMA (Carrier Sense Multiple Access)

- všechny uzly jsou rovnocenné
- uzel před započetím vysílání čeká až skončí vysílání předchozí (Carrier Sense)
- na médiu vysílá a přijímá více uzlů. Vysílání jednoho uzlu je obecně přijímáno všemi ostatními uzly užívajícími médium. (Multiple Access)

CSMA/CA (Collision Avoidance)

- uzel informuje ostatní o úmyslu vysílat
- Wifi

CSMA/CD (Collision Detection)

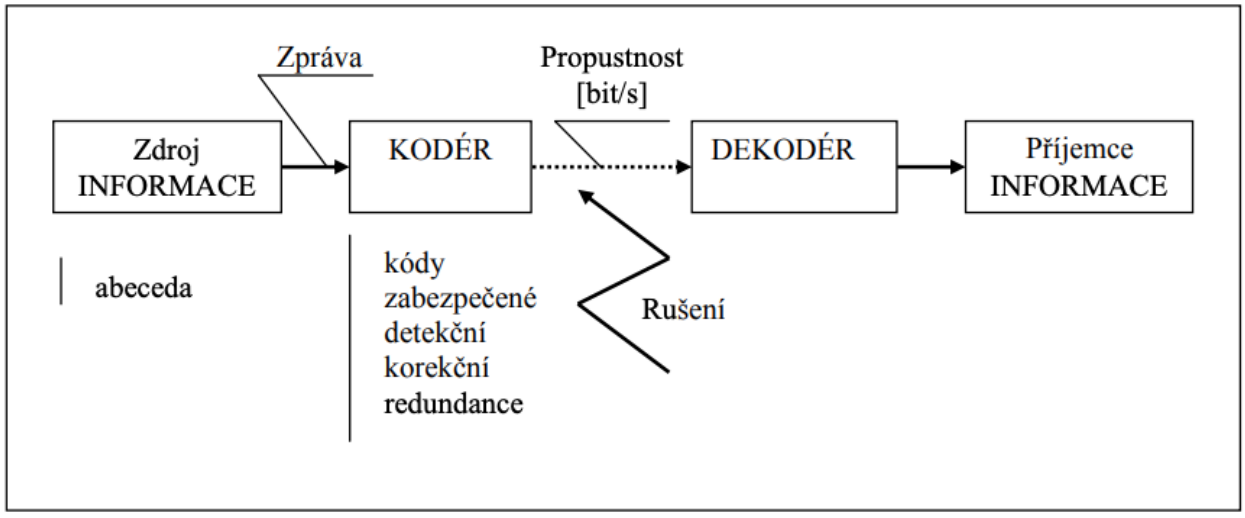
- po detekci kolize všechny uzly čekají náhodnou dobu před dalším pokusem o vysílání

CSMA/CR (Collision Resolution)

- všem uzlům je přiřazeno identifikační číslo či kód priority
- při výskytu kolize jeden z uzlů pokoušejících se vysílat současně dostane prioritu vysílat podle identifikačního čísla či kódu priority (oproti počkání náhodnou dobu a znovuvyslání jako v CSMA/CD)

Kódování

Kódování signálu



- **NRZ (Not Return To Zero)**
  Úroveň signálu přímo odpovídá 1/0
- **RZ (Return To Zero)**
  Třístavový, polovina intervalu +1 při bitu 1, -1 při bitu 0, druhá polovina intervalu nulová.
- **NRZI (Not Return To Zero Inverted)**
  1 - inverze signálu, 0 - úroveň zůstává
- **PSK (Manchester)**
  Fázová modulace, uprostřed intervalu: 0-sestup signálu, 1-vzestup signálu. Každý bitový interval má tedy uprostřed změnu. Dvojnásobné pásmo oproti přímému kódování. Použití v Ethernetu.
- **DPSK (Diferenciální Manchester)**
  1-změna na začátku intervalu, 0-absence změny na začátku intervalu *. Uprostřed intervalu změna vždy. Kóduje se změna/zachování úrovně posledního bitu (ne hodnota aktuálního bitu). Použití v Token-Ring.

Kódování dat

Prefixový kód
Žádný symbol jeho kódové abecedy není předponou (začátkem) jiného symbolu abecedy.
Př.: { 1, 21, 22, 231, 232, 24, 35, 535, 7 } je prefixový kód, { 1, 21, 22, 221, 222, 24, 35, 355, 7 } není prefixový kód.

Cyclic Redundancy Code (CRC)
- Přidat k bitů redundantních dat k n-bitové zprávě
- n-bitová zpráva je reprezentována jako polynom n-tého stupně, kde každý bit odpovídá příslušnému koeficientu v polynomu

Příklad
Generující polynom: $C(x) = x^3 + x^2 + 1 \quad \rightarrow \quad 1101$
Zpráva: $M(x) = x^7 + x^4 + x^3 + x \quad \rightarrow \quad 10011010$

Odesílatel:
1. Posuneme data o stupeň generujícího polynomu
   $M(x) = x^{10} + x^7 + x^6 + x^4 \quad \rightarrow \quad 10011010000$
2. Vydělíme
   $(x^{10} + x^7 + x^6 + x^4) : (x^3 + x^2 + 1) = \dots$
   $\vdots$
   $\overline{\qquad\qquad}$
   $x^2 + 1 \quad \rightarrow \quad 101 = R(x)$
   (zbytek po dělení)
3. Odešleme původní zprávu $M(x)$ následovanou zbytkem po dělení
   $P(x) = M(x) + R(x) = 10011010101$

Příjemce:
1. Přijme se polynom P(x) + E(x)
   - E(x) reprezentuje chyby
   - E(x) = 0 znamená bezchybný přenos
2. Dělení (P(x) + E(x)) polynomem C(x)
   - Pokud je výsledek= 0, buď
     - Nedošlo k chybě (E(x) = 0, P(x) je dělitelné C(x) beze zbytku)
     - (P(x) + E(x)) je beze zbytku dělitelné C(x), chyba nebyla detekována

Výtahy zpráv (hashovací funkce)

- ze vstupu proměnné délky vytváří malou hodnotu
- ze stejného vstupu vytváří vždy stejný výstup
- každé výsledné hodnotě by mělo odpovídat více vstupních kombinací
- algoritmus by neměl být snadno odvoditelný či invertovatelný
- malá změna na vstupu má za následek velké změny ve výstupu

Aplikace: zabezpečení dokumentů (ftp), dig. podpis
Příklady: MD2, MD5, SHA, HAVAL, SNEFRU, RIPEMD160 ← nejsou bezpečné pro ukládání hesel
Krypthorafické funkce: bcrypt, argon2(d,di,id),**PBKDF2,SCrypt**

> ***Hammingova vzdálenost*** *je nejmenší počet pozic, na kterých se řetězce stejné délky daného kódu liší, neboli počet záměn, které je potřeba provést pro změnu jednoho z řetězců na druhý.*

Digitální podpis

Jedná se o výtah zprávy zašifrovaný privátním klíčem autora dokumentu.

- klíč distribuován spolu s dokumentem
- držitel příslušného veřejného klíče je schopen dešifrovat zakódovaný výtah zprávy a porovnat ho s výtahem, který vytvoří z obdrženého dokumentu.
- digitální podpis zajišťuje tři funkce:
  - integritu
  - autentizaci(kdo zprávu podepsal)
  - nepopiratelnost (autor nemůže v budoucnu zapřít, že zprávu podepsal)
- zpráva (soubor) bude čitelná (použitelná) i v případě, že nemáme příslušné nástroje pro ověření její pravosti

Spolehlivost datových přenosů

Typy chybových modelů

- kanály “bez paměti”:
  - AWGN kanál
  - BSC kanál
- kanály “s paměti”:
  - Gilbert-Elliott 1960

AWGN kanál (Additive white Gaussian noise)

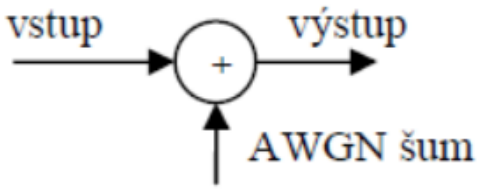
- (jediný) zdroj chyb v kanále je aditivní šum
- nezohledňuje řadu typů chyb (únik, vícecestné šíření, interference, ...)
- požití v modelech satelitních komunikací

BSC kanál (Binary Symmetric Channel)

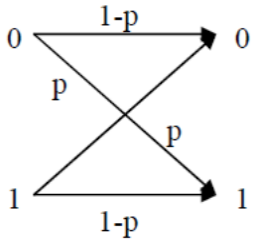
- p – pravděpodobnost chybného přenosu bitu

Gilbert-Elliott

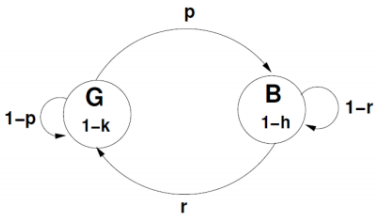
- pravděpodobnost chybného přenosu bitu závisí na výsledku přenosu bitu předchozího („závislé ztráty“)
- modelujeme pomocí dvoustavového Markovova řetězce
  - p ... pravděpodobnost přechodu ze stavu Good do stavu Bad
  - r ... pravděpodobnost přechodu ze stavu B do stavu G
  - 1-k ... pravděpodobnost chybného přenosu bitu v G stavu (obvykle = 0)
  - 1-h ... pravděpodobnost chybného přenosu bitu v B stavu (např. 0.5)



AWGN kanál



BSC kanál



Gilbert-Elliott

Zpracování chyb při přenosu

Automatic Repeat reQuest (ARQ)

- algoritmy detekce chyb
- v případě chyby je zdroj požádán o opakování přenosu

Stop and Wait ARQ

- po odeslání dat je očekáváno potvrzení (ACK)
- pokud není přijato do určité doby, přenos se opakuje
  - neefektivní pro kanály s vysokým zpožděním
  - potvrzení se může ztratit

        - tatáž data vyslána 2x
    - v případě dočasného zvýšení zpoždění může být potvrzení přiřazeno špatnému rámci
- lze řešit číslováním rámců a potvrzení

Go-Back-N ARQ

- datové rámce (pakety) obsahují pořadové číslo
- vysílající uzel smí odeslat až N rámců (paketů), aniž by obdržel potvrzení
    - N je velikost vysílacího okénka
- potvrzení obsahuje pořadové číslo posledního správně přijatého rámce (paketu)
    - chybné (či chybějící) a následující jsou ignorovány
- po odeslání N rámců (paketů) vysílač vyhodnotí pořadové číslo posledního přijatého potvrzení a pokračuje ve vysílání následujícího rámce (paketu)
    - rozsah pořadového čísla musí být > N
- všechny rámce (pakety) odeslané po chybě jsou opakovány 😢

Selective Repeat ARQ

- datové rámce (pakety) obsahují pořadové číslo
- vysílající uzel smí odeslat až N rámců (paketů), aniž by obdržel potvrzení
    - N je velikost vysílacího okénka
- přijímač přijímá všechny bezchybně doručené rámce i po výskytu chyby až do počtu M
    - M je velikost přijímacího okénka
- potvrzení obsahuje pořadové číslo prvního chybného rámce (paketu) nebo dalšího v pořadí (nenastala-li chyba)
- vysílač vyhodnotí pořadové číslo v posledním přijatém potvrzení a pokračuje vysíláním rámce (paketu) s tímto číslem a následujících (max. N, bezchybné se neopakují)
- rozsah pořadového čísla musí být ≥ N + M

Protokoly rodiny TCP/IP

IP[editovat | editovat zdroj]
Podrobnější informace naleznete v článku Internet Protocol.
Internet Protocol je základní protokol síťové vrstvy a celého Internetu. Provádí vysílání **datagramů** na základě síťových IP adres obsažených v jejich záhlaví. Poskytuje vyšším vrstvám **síťovou službu bez spojení**. 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). Tento protokol se dále stará o segmentaci a znovusestavení datagramů do a z rámců podle protokolu nižší vrstvy (např. ethernet).
V současné době je převážně používán protokol IP verze 4. Nová verze 6, která řeší nedostatek adres v IPv4, bezpečnostní problémy a vylepšuje další vlastnosti protokolu IP, je celosvětově používána jen několika procenty zařízení připojených k internetu, ale jejich počet rychle roste.

IPv4[editovat | editovat zdroj]

Podrobnější informace naleznete v článku IPv4.

Internet protokol verze 4

- 32 bitové adresy
- cca 4 miliardy různých IP adres, dnes nedostačující
- formát: xxx.xxx.xxx.xxx kde xxx je libovolné číslo od 0 do 255 (8 bitů)

IPv6[editovat | editovat zdroj]

Podrobnější informace naleznete v článku IPv6.

Internet protokol verze 6

- 128 bitové adresy
- podpora bezpečnosti
- podpora pro mobilní zařízení
- funkce pro zajištění úrovně služeb (QoS - Quality of Service)
- fragmentace paketů - rozdělování
- není zpětně kompatibilní s IPv4
- snadnější automatická konfigurace (NDP - Neighbor discovery protocol)

ARP[editovat | editovat zdroj]

Podrobnější informace naleznete v článku Address Resolution Protocol.

Address Resolution Protocol se používá k nalezení fyzické adresy MAC podle známé IP adresy. Protokol v případě potřeby vyšle datagram s informací o hledané IP adrese a adresuje ho všem stanicím v síti. Uzel s hledanou adresou reaguje odpovědí s vyplněnou svou MAC adresou. Pokud hledaný uzel není ve stejném segmentu, odpoví svou adresou příslušný směrovač.
Příbuzný protokol RARP (Reverse Address Resolution Protocol) má za úkol najít IP adresu na základě fyzické adresy.

ICMP[editovat | editovat zdroj]

Podrobnější informace naleznete v článku ICMP.

Internet Control Message Protocol slouží k přenosu **řídících hlášení**, které 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.

TCP[editovat | editovat zdroj]

Podrobnější informace naleznete v článku Transmission Control Protocol.

Transmission Control Protocol vytváří virtuální okruh mezi koncovými aplikacemi, tedy **spolehlivý přenos dat**. Vlastnosti protokolu:

- Spolehlivá transportní služba, doručí adresátovi všechna data bez ztráty a ve správném pořadí.
- Služba se spojením, má fáze navázání spojení, přenos dat a ukončení spojení.
- Transparentní přenos libovolných dat.
- Plně duplexní spojení, současný obousměrný přenos dat.
- Rozlišování aplikací pomocí portů.

UDP[editovat | editovat zdroj]

Podrobnější informace naleznete v článku User Datagram Protocol.

User Datagram Protocol poskytuje nespolehlivou transportní službu pro takové aplikace, které nepotřebují spolehlivost, jakou má protokol TCP. Nemá fázi navazování a ukončení spojení a už první segment UDP obsahuje aplikační data. UDP je používán aplikacemi jako je DHCP, TFTP, SNMP, DNS a BOOTP.
Protokol používá podobně jako TCP čísla portů pro identifikaci aplikačních protokolů.

SCTP[editovat | editovat zdroj]

Podrobnější informace naleznete v článku Stream Control Transmission Protocol.

Spolehlivý protokol pro přenos datagramů ve více proudech. Je využívaný zejména v telekomunikacích. Doplňuje některé vlastnosti, které TCP postrádá:

- Multihoming - komunikující uzel může mít několik IP adres.
- Členění datového toku na datagramy.
- Používání více proudů dat - omezuje blokování komunikace způsobené chybějícím blokem dat, ke kterému může dojít v TCP.
- Výběr a sledování cesty - Pokud má primární adresa problémy s dostupností lze používat alternativní.
- Ověřovací a potvrzovací mechanismy - SCTP komplikuje některé útoky směřující k nedostupnosti služeb (DoS). Zajišťuje ověření opakujících se a chybějících balíků.

Stejně jako TCP a UDP rozlišuje aplikační protokoly pomocí portů.