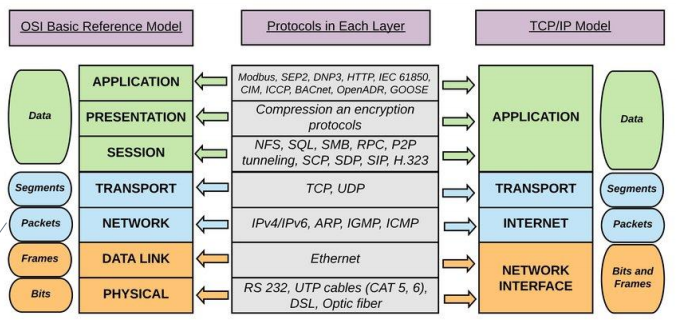
**BI-SPOL-23 ISO/OSI a TCP/IP model. Protokoly linkové vrstvy. Přepínání a směrování. Principy fungování propojovacích síťových prvků**

BI-PSI

### ISO/OSI a TCP/IP

**ISO/OSI** – teoretický model pro popis sítí a protokolů pro komunikaci. Nikdy nebyl plně implementován

**TCP/IP** – vznikl na základě reálných požadavků a praktických zkušeností v komunikaci (úspěšný díky internetu)



**Fyzická vrstva**

* umožňuje přenos bitů kanálem
* předepisuje vlastnosti média
* Určuje způsob sdílení média a definuje elektrické a mechanické rozhraní
* definuje způsob přenosu 0 a 1
* optika, kabely…

**Spojová (linková) vrstva**

* detekce spolehlivého připojení (detekce, příp. korekce chyb)
* formátování dat do rámců
* řízení přístupu a toku na lince
* jednoznačná adresa v rámci segmentu sítě (MAC adresa)
* obsahuje dvě podvrstvy:
  + MAC – řízení přístupu k médiu, sdílení média
  + LLC – poskytuje mechanismus multiplexování, které umožní, aby se v jedné síti mohlo používat více síťových protokolů (IPv4, IPv6)

**Síťová vrstva**

* adresace a směrování dat přes mezilehlé prvky (pomocí IP adres)
* jednoznačná adresa v rámci celé sítě (IP adresa)
  + vezme data od transportní vrstvy a přidá k nim hlavičku (např. IPv4, nebo IPv6) – nazýváme **paket**

**Transportní vrstva**

* rozklad dat na jednotlivé pakety a jejich uspořádání dle pořadí
* na transportní vrstvě se pracuje s portem
* multiplexuje a demultiplexuje data mezi transportními spoji (př. TCP, UDP
  + transportní vrstva dělá to, že vezme data (např. dlouhou zprávu) a rozdělí ji na menší díly
  + ke každému dílu přidá hlavičku (např. TCP, UDP, …) – nazýváme **segmenty**
* transportní adresy (adresa, port)

**Relační vrstva**

* vytváření logického rozhraní pro aplikace
* synchronizace spojení (transakce)
* přihlášení, udržení relace
* např. sdílení disků

**Prezentační vrstva**

* formátování a prezentace dat
* transformace dat (komprese/dekomprese)
* kódování a šifrování
* data – jejich struktura, nikoliv jejich význam
* např. ASCII, UNICODE

**Aplikační vrstva**

* způsob komunikace aplikací (protokoly)
* podpůrné funkce aplikacím
* představuje interface pro uživatele
* Na této vrstvě jsou data a zabýváme se jejich významem
* např. HTTP

### Protokoly linkové vrstvy

* hlavním účelem linkové vrstvy je přenášení dat v rámci lokální či metropolitní sítě
* základní přenášenou jednotkou je **rámec**, ten se skládá z:
  + Header (Záhlaví)
  + Data
  + Trailer (Zápatí)
* linkování se provádí na základě MAC adres (unikátní pro každý síťový prvek – router, karta atd, přiděleno výrobcem, 6x8 bitů – **00:1f:64:3f:4d:8d**)
* Linková vrstva se skládá z dvou podvrstev (MAC a LLC)

**MAC (Medium Access Control) podvrstva**

* Definuje linkové adresy (MAC adresy)
* Je hardwarově závislá
* Zajišťuje:
  + Předávání dat k fyzické vrstvě
  + Přístup k médiu (může být nesdílený nebo sdílený)
  + Sdílený přístup k médiu se řeší pomocí multiplexu, popř. přístupových metod
  + Fyzickou adresaci prostřednictvím MAC adres
  + Filtrování MAC adres (kvůli bezpečnosti)
  + Přepínání a doručování rámců (mezi zařízeními)
  + Ukládání rámců do front a plánování jejich odesílání
  + Kvalitu služeb (Quality of Service)
  + Vytváření virtuálních sítí (VLAN)
* Její popis je vždy svázán s konkrétní technologií, která je použita na úrovni fyzické vrstvy (metalika, optika, bezdrát)

**LLC (Logical Link Control) podvrstva**

* Zajišťuje řízení toku a kontrolu chyb (k tomu využívá detekční a samoopravné kódy)
  + Obecně se používá pro zajištění doručování dat
  + Nesmí odesílat více rámců, než je cíl schopný přijmout
  + Potvrzovací schémata
    - Jednotlivé potvrzování (Stop&Wait)
    - Kontinuální potvrzování se selektivním opakování
    - Kontinuální potvrzování s návratem
    - Klouzavé okénko
* Kódování
* Definuje potvrzovací schéma pro spolehlivé doručování rámců
* Je hardwarově nezávislá

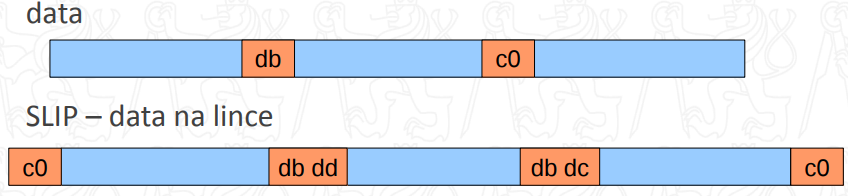
#### HDLC (High Level Data-Link Control)

* Protokol HDLC provádí detekci chyb i řízení toku dat
* Existují různé implementace s omezenou kompatibilitou
* bitově orientovaná protokol
* využit v sériových linkách a ovlivnil řadu protokolů
* synchronní i asynchronní přenos
* rozeznává tzv. módy (asi se neučit)
  + ABM – určen pro propojení dvou stanic plně duplexním spojem (tj. obě stanice mohou vysílat současně, aniž by si vzájemně na lince překáželi)
  + NRM – situace, kdy je propojeno více stanic na poloduplexním spoji (tzv přepínaný duplex mezi vysíláním a příjmem). Pro vysílání i příjem slouží společné přenosové médium, tj. v jednom okamžiku lze buď přijímat, nebo vysílat
  + ARM – málo běžný
* Flag uvozuje a ukončuje datový rámec – skládá se z 8 bitů
* Adresa – 8 bitů – adresa stanice, které je paket určen
* Řídící pole – určuje typ HDLC rámce
  + - Informační – pro data
    - U-rámce – pro data a řídící funkce (např. řízení linky)
    - S-rámce – pro řízení toku dat (požadavek na vysílání, potvrzování informačních atd.)
* Kontrolní součet – 32 nebo 16 bitů
  + Umí zjišťovat chybné rámce



#### SLIP (Serial Line IP)

* definuje pouze zapouzdření paketů na sériové lince
* nedefinuje adresaci, typ paketů, detekci chyb, kompresi…
* SLIP vkládá IP pakety přímo do sériové linky
* Pro řízení toku jsou mezi data vkládány ESC-sekvence
* Každý rámec je ukončen značkou END (většina implementací ho umisťuje i na začátek)
* Jestliže se vyskytne znak 0xc0 v přenášených datech, je nahrazen tzv. ESC-sekvencí: dvojicí 0xdb, 0xdc. Znak 0xdb je nahrazován dvojicí 0xdb 0xdd
* speciální znaky END a ESC
* rámec ohraničen znaky END



#### PPP (Point to Point Protocol)

* podmnožina HDLC
* může používat jak asynchronní, tak bitově čii znakově synchronní přenos dat
* bitové linky – bit stuffing
* umožňuje souběh více protokolů
* zpravidla využívá 16 nebo 32 bitů pro kontrolní součet (zjistí případně poškození rámce)
* cílem PPP je umožnit po jedné lince přenos více síťových protokolů současně

Obsah obrázku stůl

Popis byl vytvořen automaticky

#### Ethernet

* nejběžnější technologie linkové vrstvy
* MAC adresy – 48 bitů (6B)
  + První 3B jsou přiděleny výrobci
  + Další 3B je sériové číslo daného zařízení

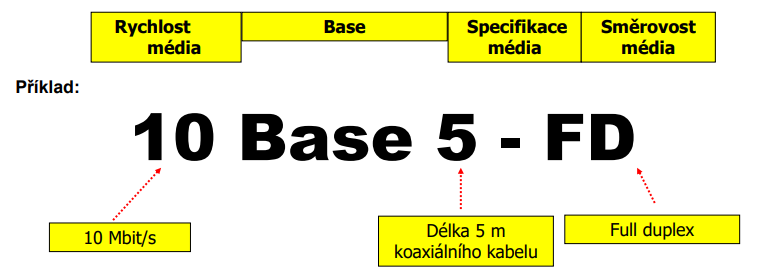
**Obecný formát rámce Ethernetu:**

Obsah obrázku text

Popis byl vytvořen automaticky

* + Preambule = úvodní hlavička pro identifikaci rámce – liší se dle verze použitého Ethernetu
  + SOF = start of frame
  + Ethertyp = 16 bitů, které specifikují detailněji typ použitého rámce (různé využití)
  + FCS (Frame Check Sequence) = sekvence 32 bitů obsahující kontrolní součet rámce umožňující detekci poškozeného rámce
  + Běžná velikost přenášených dat v Ethernetu je rovna maximální (1500 bytů). Tato velikost je nastavena v konfiguraci konkrétního síťového rozhraní a není uvedena v rámci!

**Označování druhů Ethernetu**



**Mechanismus řízení toku v Ethernetu**

* Ethernet umí ovlivnit příchozí tok a zabránit zahlcení sítě
* Pokud určitý přepínač výkonově „nestíhá“, může požádat své sousedy o zpomalení vysílání – příkazem PAUSE
* Ethernet neobsahuje žádný mechanismus, který by garantoval spolehlivé doručení dat

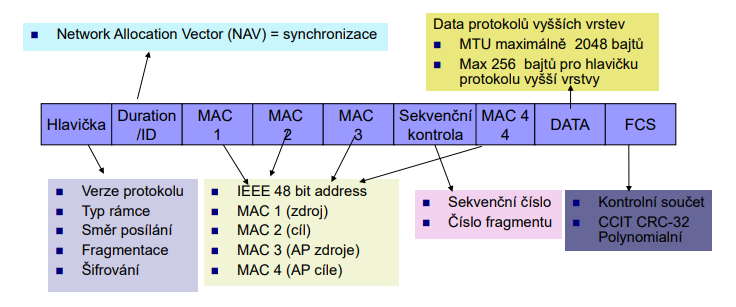
Autonegociace = metoda, kterou si vysílač a přijímač dohodnou parametry přenosu

#### Bezdrátové sítě

* Použité přenosové médium je vzduch či vakuum
* Pro přenos dat využívají antény
* Komunikační kanály (KK) = komunikační linky bezdrátových sítí
* KK jsou méně odolné vůči rušení

**IEEE 802.11 WiFi**

* Dělíme dle topologie na
  + Infrastrukturální – centrální jednotka (AP = Access Point) řídí komunikaci mezi stanicemi
  + Adhoc – stanice spolu komunikují napřímo, bez AP
* Rámec:



* Druhy rámců:
  + **Kontrolní** – slouží pro připojení stanice do sítě (přihlášení k AP)
  + **Řídící** – řídí přístup k médiu, tj. určuje, která stanice může vysílat
  + **Datové** – přenášejí data mezi AP a stanicí, popř mezi stanicemi

**Bluetooth** – propojení zařízení místo kabelu

### Přepínání a směrování

#### Přepínání (switching)

* propojuje dvě sítě na základě MAC adresy a přenáší rámce – **v lokální sítí**
* Každý **přepínač (switch)** obsahuje vstupně/výstupní porty, pomocí kterých jsou připojeny komunikační linky.
* Porty obsahují vyrovnávací paměť (buffers)
* Funkcí těchto zařízení je **přepínání (switching)** rámců. Přepínání rámců znamená přijetí rámce, zpracování tohoto rámce a odeslání výstupním portem ven směrem k příjemci.
* **Přepínací tabulka** (Switching table) slouží k výběru výstupního portu dle MAC adresy příjemce – tabulka dvojic – port a MAC adresa
* **Záznamy** ve směrovací tabulce **se** **aktualizují** u přepínačů **učením** (poprvé se rámec s neznámou adresou příjemce pošle na všechny porty s výjimkou odkud přišel a zaznamená se, z kterého portu přišla odpověď)
* Rámce lze zadržovat ve výstupních frontách (bufferech) a tím lze regulovat odchozí tok.
* na linkové vrstvě

**Režimy práce přepínačů**

* *store-and-forward*:
  + přepínač přijme celý rámec
  + uloží jej do až pak odešle dále
* *cut-through*:
  + přepínač přijme hlavičku rámce
  + překontroluje jen zdrojovou a cílovou adresu a odesílá data okamžitě

#### Směrování (routing)

* **cesta** = posloupnost směrovačů, kterou musí projít paket mezi zdrojovou a cílovou adresou
* **směrování** = hledání cesty mezi konkrétní zdrojovou a cílovou stanicí
  + slouží pro doručení dat adresátovi, pokud možno nejefektivnější cestou
* informace, na základě kterých probíhá směrování, jsou uloženy ve **směrovacích tabulkách**
* **statické směrování** – směrovací tabulky na směrovačích se nastavují manuálně
* **dynamické směrování** – směrovací tabulky se konfigurují autonomně během provozu

**Druhy směrování**

* *Optimální* (z hlediska nějakého kritéria = metriky)
  + **Nejkratší** (shortest path), typicky z hlediska počtu mezilehlých směrovačů
  + Nejlevnější (různé trasy stojí finančně různě, jsou-li na trase zpoplatněné celky)
* *Redundantní* (tzv. multi-path)
  + Provoz může být veden různými cestami. Hodí se pro zálohování cest popř. rozdělování zátěže
* *Dle symetrie*
  + **Symetrické** (dopředná i zpětná cesta jsou stejné)
  + **Asymetrické** (dopředná i zpětná cesta nejsou stejné, např. satelitní systémy)
* *Dle způsobu hledání cesty*
  + **záplavové** (nejjednodušší, používá je řada technologií a protokolů).
  + **proaktivní** (cesta je předpočítána a uložena ve směrovacích tabulkách).
  + **reaktivní** (cesta se vytváří až když je potřeba) - adhoc či mobilní sítě (proměnlivá topologie).
* *Dle oblasti výpočtu trasy*
  + **Vnitřní** (v rámci jedné sítě)
  + **Vnější** (mezi různými sítěmi)

**Redundantní směrování**

* směrování k cílové stanici je možné prostřednictvím více navzájem disjunktních cest. Směrovač si zvolí jednu cestu na základě určitého kritéria (např. nejkratší vzdálenost).
* Redundantní směrování se běžně používá v případě, že je nutné garantovat vysokou dostupnost cílové stanice (resp. sítě, ve které se stanice nachází).

**Záplavové směrování**

* Používá BFS
* Požadavek je opatřen identifikátorem (ID) – podle toho směrovač zjistí, jestli ho už neobdržel dříve – pokud ano, tak zahodí, pokud ne odešle ho všem sousedům

**Směrovací tabulka**

* Funguje jako rozcestník, který umožnuje zjistit směr, kterým má směrovač přijaté pakety dále odesílat
* Typické položky záznamu ve směrovací tabulce:
  + Destinace: IP (analogie cílové město).
  + Brána: IP (analogie další rozcestník na trase).
  + Maska: dle prefixu (analogie oblast).
  + Metrika: cena (analogie vzdálenost v km).
  + Rozhraní: název (analogie vstup do silnice)

**Proaktivní směrování**

* Směrování, které používá směrovací tabulky
* Proaktivní směrování se využívá v sítích, které se nemění anebo mění vždy za delší dobu.
* U proaktivního směrování před posíláním dat je cesta od zdrojové stanice k cílové již známá.
* Tento způsob je nejběžnější v počítačových sítích

**Reaktivní směrování**

* Směrování, které **nepoužívá** směrovací tabulky
* Směrovací tabulky se nepoužívají, jelikož jejich sestavování může být pro dané typy sítí příliš pracné či neúčelné.
* Využití toho druhu směrování má smysl např. v mobilních sítích popř. v ad-hoc sítích, protože se jejich uspořádání (topologie) často mění.
* Směrování má obvykle 2. fáze

1. nalezení cesty mezi zdrojovou a cílovou stanicí (většinou záplavovým směrováním).
2. posílání dat přes nalezenou cestu a následné zapomenutí cesty.

-Cesta může být uložena v hlavičce anebo dočasně v paměti mezilehlých směrovačů

**Statická konfigurace směrovacích tabulek**

* Typické operace ve směrovací tabulce
  + Přidání záznamu.
  + Odebrání záznamu.
  + Změna záznamu (např. sousední směrovač nebo metrika).
* Vlastnosti statického směrování
* Funguje okamžitě po zapnutí směrovačů.
* Není schopné reagovat na změny.
* Vyžaduje manuální konfiguraci všech prvků → pracné pro velké sítě.
* Hodí se hlavně pro menší sítě s jednoduchou strukturou.

**Dynamická konfigurace směrovacích tabulek**

* Tabulky si nastavují směrovače automaticky s využitím **směrovacích protokolů**
* Směrovače si informace týkající se směrování periodicky předávají zasíláním zpráv.
* Díky periodickému zasílání zpráv směrovače dokáží rychle reagovat na změny v síti (např. připojení nových směrovačů, výpadky linek atd.).
* Dynamické konfigurace směrovacích tabulek ve velkých sítích mohou trvat i desítky vteřin.
* Rozlišujeme 3 druhy dynamických směrovacích algoritmů (A), které jsou pojmenovány dle obsahu zasílaných zpráv a sice:
  + Distance Vector Algoritmy (DVA)
  + Link State Algoritmy (LSA)
  + Path Vector Algoritmy (PVA)

*DVA*

* Směrovače si vyměňují navzájem záznamy svých směrovacích tabulek prostřednictvím zpráv (M). Metrika je vzdálenost. – záznam – (cíl, brána, metrika)
* Používá se protokol **RIP**
  + Směrovače si vyměňují navzájem záznamy
  + Když se připojí nový router dá o sobě vědět – odešle zprávy – nejbližší router mu zase odešle celou svojí tabulku
  + Umí detekovat výpadek linky a nastaví metriku příslušného řádku na ∞

LSA

* Linka = propojení dvou sousedních směrovačů s určitým ohodnocením
* Směrovače si vyměňují stavy linek a z těchto si každý směrovač vytvoří vnitřní reprezentaci ohodnoceného grafu sítě
  + Směrovač si poté nad grafem najde kostru, která obsahuje nejkratší cestu (používá Dijkstru)
  + Z nalezené kostry se procházením do hloubky (DFS) sestaví postupně směrovací tabulka

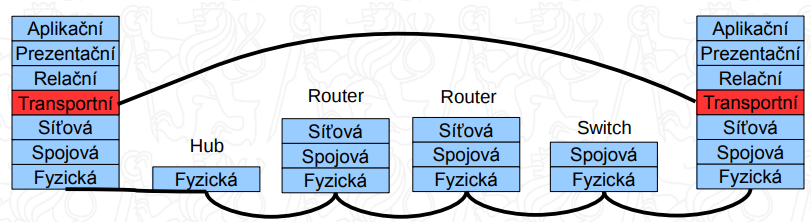
Obsah obrázku stůl

Popis byl vytvořen automaticky

PVA

* Směrovače si vyměňují kompletní cesty ke konkrétním cílům

### Principy fungování propojovacích síťových prvků



**Hub**

* Přeposílá obdržený paket na všechny své porty bez podmínek
* Na fyzické vrstvě

**Repeater**

* Zesilovač signálu pro předávání na velké vzdálenosti
* Na fyzické vrstvě

**Switch**

* Jako hub přeposílá pakety, ale už si pamatuje přiřazení MAC-adresa zařízení se svými porty, proto odesílá pouze na potřebné porty
* Na linkové vrstvě

**Bridge**

* Propojují, resp. oddělují provoz více částí sítě (obsahují taktéž přepínací tabulky). Mohou být realizované jako HW či SW zařízení. Softwarové mosty jsou implementovány jako součástí operačního systému
* Jednodušší než switch – málo portů (2 až 4)
* Propojují většinou jen dvě sítě
* Na linkové vrstvě

**Router (směrovač)**

* Propojuje sítě na síťové vrstvě, směřuje pakety na základě IP adres
* Na síťové vrstvě