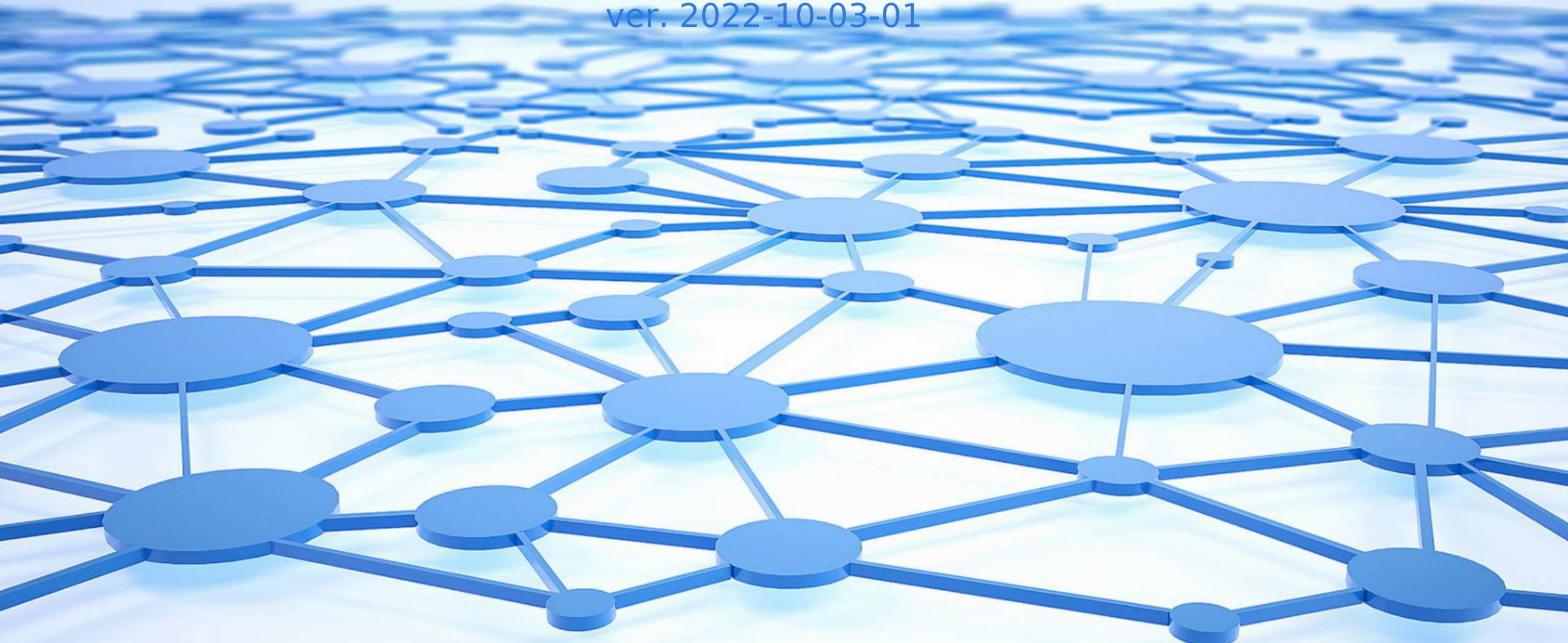


Úvod do počítačových sítí

Přednáška 3

(2023/2024)

ver. 2022-10-03-01



ISO/OSI model

- Referenční model popisující fungování počítačové sítě
 - Neřeší jednotlivé technologie, jen zavádí vrstvy a co by se na nich mělo řešit
- Má sedm vrstev
- Vznikl bez vazby na konkrétní implementaci
- Snaží se být maximalistický a řešit vše
 - Což se neukázalo jako ideální řešení
- Vychází spíše z telekomunikace
- Některé vrstvy jako L5 a L6 se nepoužívá
 - Respektive v praxi nejsou realizované jako samostatná vrstva
- Některé vrstvy se musí dále dělit L2
 - LCC A MAC

ISO/OSI model

7	Data	Aplikační vrstva	Komunikace s procesem
6	Data	Prezentační vrstva	Prezentace dat a šifrování
5	Data	Relační vrstva	Koordinace komunikace
4	Segment	Transportní vrstva	Spojení
3	Paket	Síťová vrstva	Určení cesty a logická adresace
2	Rámec	Linková vrstva	MAC a LLC – fyzická adresace
1	Bity	Fyzická vrstva	Média, signál, binární přenos

ISO/OSI model – popis vrstev

1. Fyzická

- Spojení dvou uzlů, přenos signálů jako je například světlo, řeší přenos jednotlivých bitů, řeší kódování, synchronizaci, časování, modulaci, parametry vodičů a signálu, konektory atd.

2. Linková

- dělí se vnitřně na dvě vrstvy
 - MAC - Media Access Control - fyzické adresování, přístup k mediu
 - LLC - Logical Link Control - zabezpečení proti chybám, multiplex, řízení toku dat

3. Síťová

- Přenos paketů - forwarding, adresace síťovou adresou, např. IPv4 nebo IPv6, směrování dat v síti - routing

4. Transportní

- „Přizpůsobovací vrstva“, přenos segmentů, adresace pomocí protokolu (TCP, UDP, ICMP, ..) a portu, spojuje aplikace

5. Relační

- Tvorba a ukončování spojení, tvoří relaci mezi komunikujícími uzly, řeší výpadek spojení, šifrování

6. Prezentační

- Převádí přenesená data do formátu podporovaného aplikací a zpětně data od aplikaci do formátu vhodného pro přenos, měla by řešit rozdíly v platformách - například pořadí bitů v číslech, kódování atd.

7. Aplikační

- Vrstva určená pro běh samotných aplikací

TCP/IP model

- ISO/OSI je příliš teoreticky, v praxi se používá zjednodušený TCP/IP model, který vznikl na základě použitých technologií
- TCP/IP model má jen 4 nebo 5 vrstvy:
 1. Vrstva síťového rozhraní – síťového přístupu
 - fyzická+přístupová (proto 4 nebo 5)
 - závislá na mediu, využívá existující řešení např Ethernet
 2. Síťová - internetová
 - nezávislá na mediu
 - řeší adresování a směrování, IP, ..
 3. Transportní
 - komunikace mezi procesy
 - adresa protokolem a portem, TCP, UDP, ..
 4. Aplikační
 - komunikace mezi aplikacemi
 - zde přenášíme ta data co opravdu chceme SMTP, HTTP, FTP, ...

RM ISO/OSI



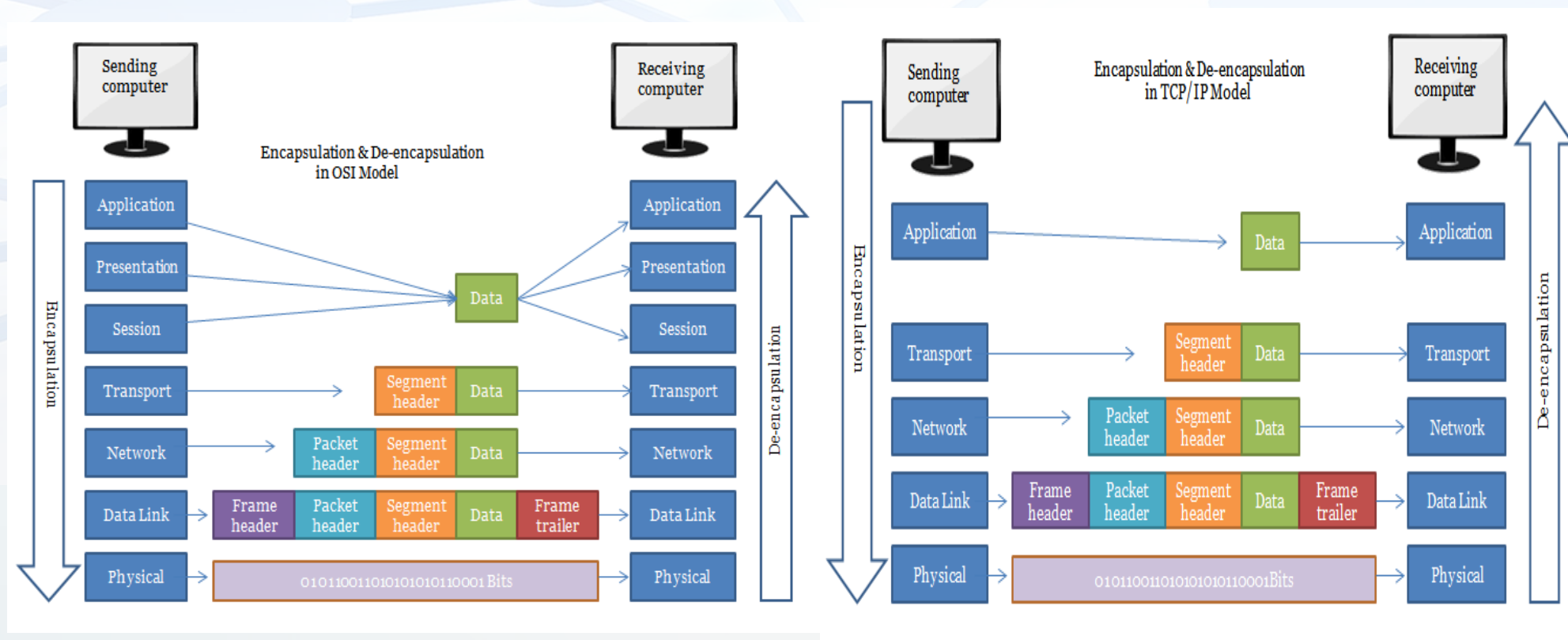
TCP/IP



Zapouzdření a rozbalování

- Nepřenášíme jen požadovaná data, ale i další řídicí data, jako je například adresa odesílatele, adresa příjemce
- Na každé jednotlivé vrstvě ISO/OSI se adresuje jinak každá vrstva potřebuje přidat své řídicí informace
- Zapouzdření
 - Přijmu data od nadřazené vrstvy (například L2 přijme paket od L3)
 - Přijatá data doplním další informace v podobě hlavičky a patičky
 - **L2 hlavička** + „L3 data – paket“ + **L2 patička**
 - Nově vzniklý datový blok předám nižší vrstvě ke zpracování
- Rozbalení
 - Přijmu data od nižší vrstvy (například z L1 přijme blok dat, na L2 detekuje v nich rámeček)
 - Odstraní z přijatých svou hlavičku a patičku
 - ~~L2 hlavička~~ + „**L3 data – paket**“ + ~~L2 patička~~
 - Předám data L3 vrstvě ke zpracování

Zapouzdření a rozbalení - příklad



zdroj: <https://www.stackoverflow.com>

zdroj: <https://www.computernetworkingnotes.org>

ISO/OSI – L1: Fyzická vrstva

- Základní funkce a význam fyzické vrstvy
- Zařízení fyzické vrstvy
- Topologie a typy spojů a typy přenosů
- Analogový a digitální signál
- Fourierova analýza
- Modulační rychlost
- Nyquistovo a Shannonovo kritérium
- Rušení a útlum
- Synchronizace
- Kódování
- Přenos v základním a přeneseném pásmu
- Modulace
- Multiplex

L1: Základní funkce

- Jedná se nejnižší vrstvu, která je implementována hardwarově
- Hlavním cílem je přenos dat mezi dvěma koncovými body
- Chceme přenést data od nadřazené vrstvy (frame/rámec) rozložený na jedničky a nuly
- K přenosu využíváme:
 - Zařízení: síťová karta, modem, opakovač, hub
 - Přenosové medium: metalický vodič, optické vlákno, vzduch/prostor, ...
 - Přenášený signál: fyzikálně měřitelný signál
 - napětí, proud, světlo, radiové vlny, zvuk,
- **Základní problém: Jak co nejpřesněji a nejrychleji přenést co největší množství binárních dat (0,1) pomocí výše uvedených signálů / vodičů / zařízení.**
 - Chceme přenášet co nejrychleji
 - Chceme přenášet co největší množství dat za jednotku času
 - V jednom stavu signálu nemusím přenést jen 0/1
 - Chceme přenášet s co nejmenším počtem chyb
 - Chyby nepoznáme a musí je za nás řešit vyšší vrstva, což „drahé“



L1: Zařízení fyzické vrstvy

- Síťové karty (vysílač – přijímač)
 - Metalické, optické, bezdrátové, ...
 - Zajišťují v zařízení (jako je PC, switch, router, mobil, ...) převod vstupního signálu na 0 a 1 a opačně
- Zesilovače
 - Přijímají signál a zesilují jej na výstupu
- Huby – Více-portové opakovače
 - Přijímají signál na jednom portu a kopírují/opakují jej na všechny ostatní porty
 - Zesilují signál a zároveň umožňují propojit více zařízení
- Modem – „modulátor - demodulátor“
 - Zajišťuje přenos dat počítačové sítě (0 a 1) jinou technologií – například prostřednictvím telefonní linky nebo kabelové televize
 - Musí umožnit jak přenos dat původního významu (telefonní hovor) tak nových – počítačové data
 - Opakem **modemu** je **kodek**, který zajišťuje přenos například hlasu pomocí počítačové sítě

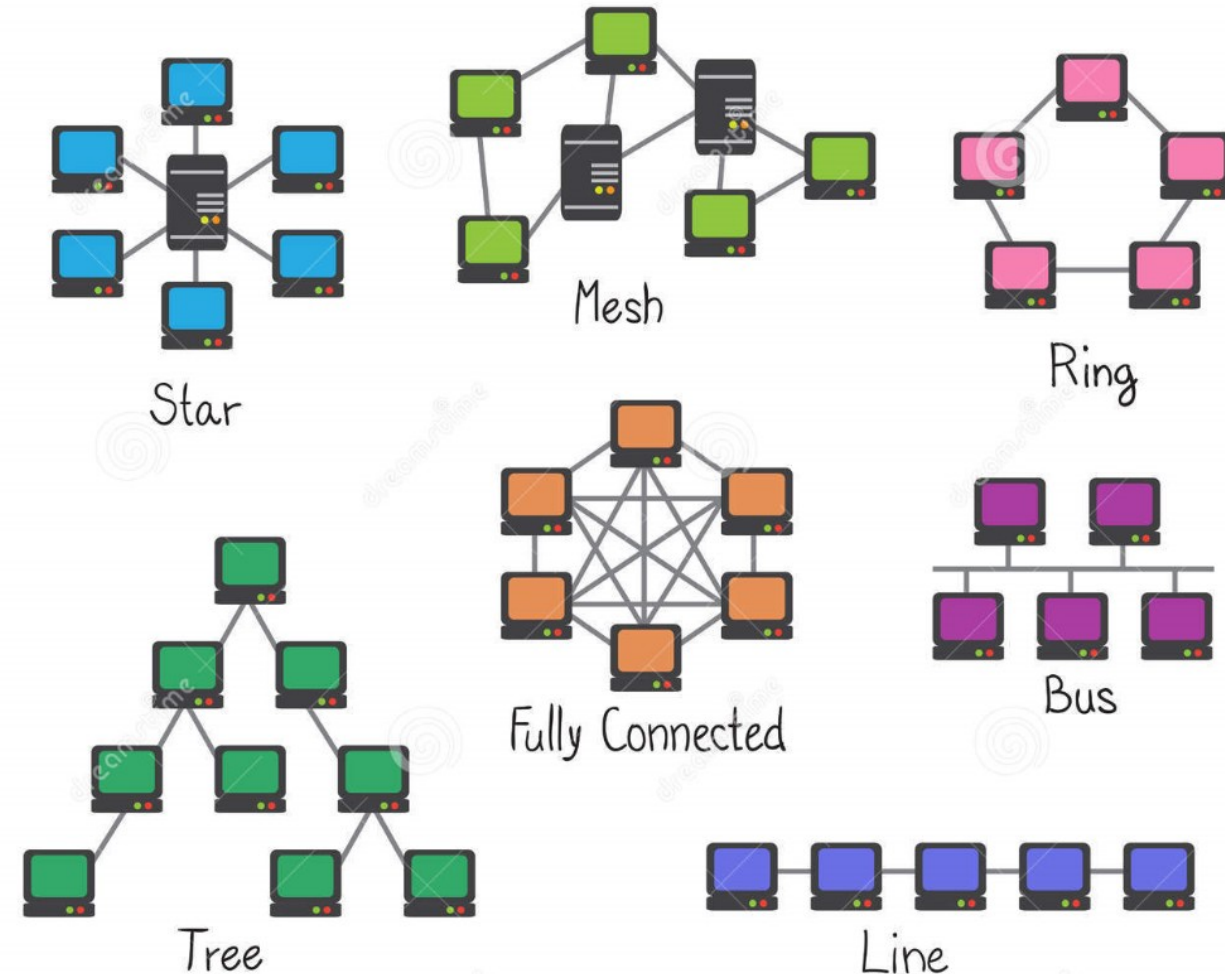


L1: Typy propojení uzlů

- Propojení uzlů
 - Dvoubodové – jeden vysílač a jeden přijímač
 - Jednoduché, ale nákladné
 - Použití například v WAN
 - Mnohabodové – více vysílačů a více přijímačů
 - Nutné řešit sdílení media – tedy kdo kdy může vysílat aby nedocházelo ke kolizím
 - Dva typy z pohledu řízení přístupu
 - Pasivní – sběrnice, například Ethernet
 - Nikdo přístup k mediu aktivně neřídí, každé zařízení je zde samo za sebe
 - Aktivní – kruhové sítě, Token Ring
 - V síti existují zařízení či sdílené principy, které aktivně určují kdo kdy může síť využívat

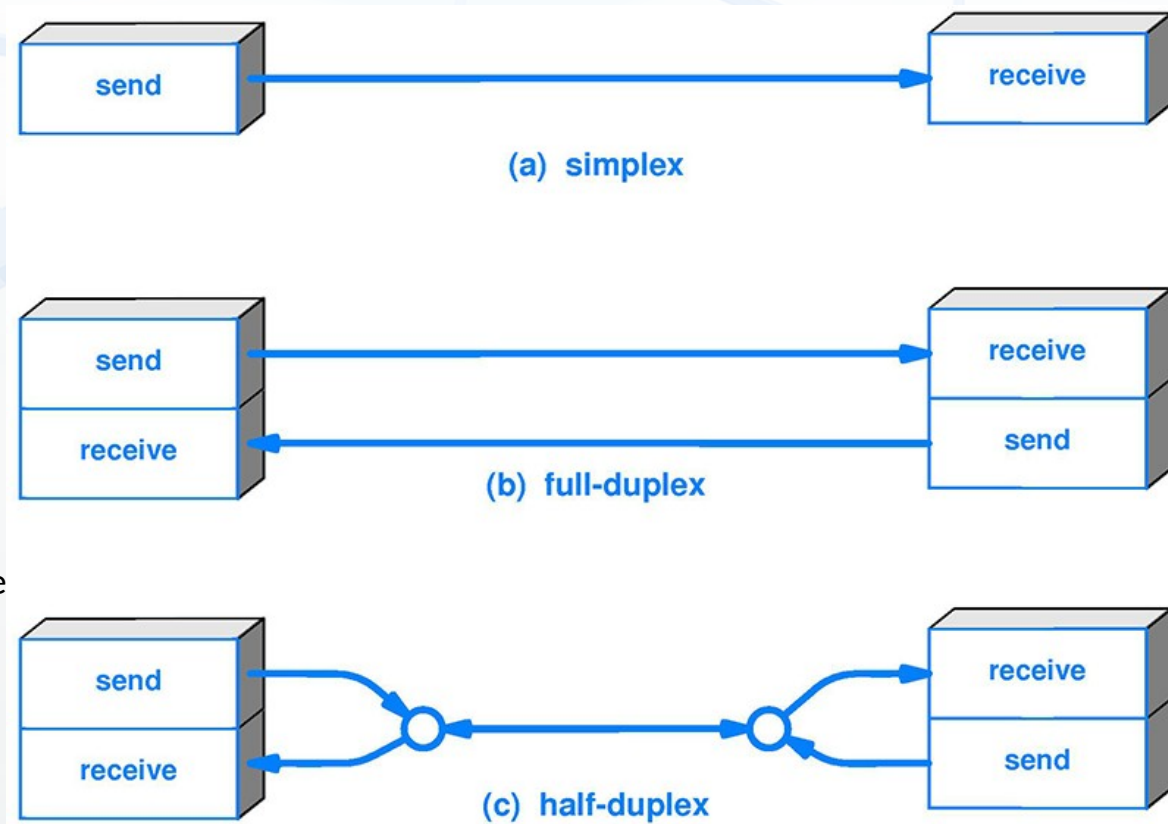
L1: Topologie

- Topologie propojení
 - Úplná polygonální síť (Full Connected)
 - propojení každého uzlu s každým
 - Neúplná polygonální síť (Mesh)
 - propojení každého uzlu s jedním nebo více dalšími
 - Hvězdová síť (Star)
 - jeden hlavní uzel, ke kterému jsou připojeni ostatní
 - Hierarchická síť (Tree)
 - stromové propojení
 - Lineární síť (Line)
 - zapojení k sousedním uzlům vždy jedním spojem
 - Kruhová síť (ring)
 - „lineární síť“ s propojenými konci
 - Může být jak fyzický nebo logický například nad Bus
 - Sběrníkové zapojení (Bus)
 - všichni připojeni ke společné sběrnici



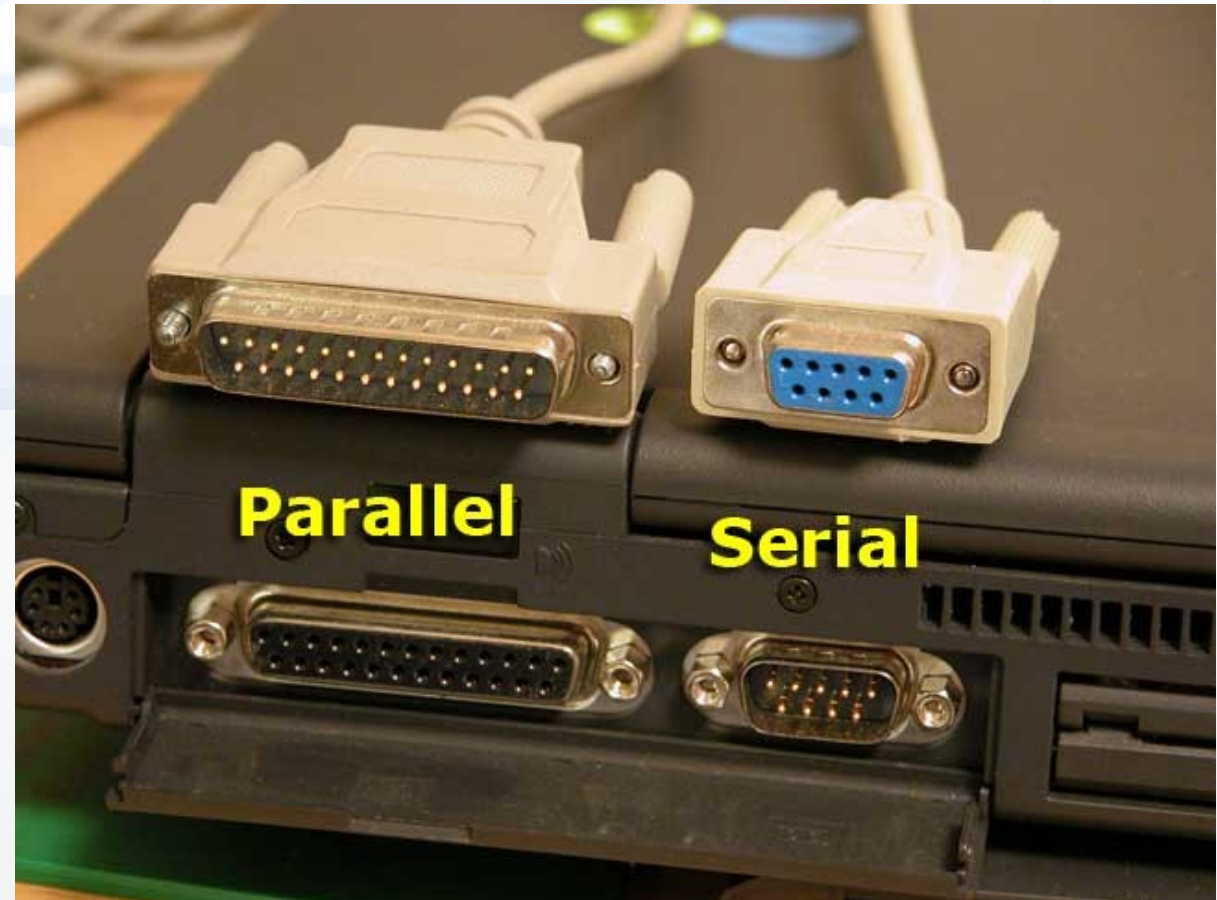
L1: Druhy linek dle možnosti přenosu

- Simplexní
 - přenos jen v jednom směru
 - Například TV
- Duplexní
 - současný přenos v obou směrech
 - Například Ethernet od 100Base-TX
 - extra vodič pro vysílání a příjem, které jsou navzájem křížené
- Poloduplexní
 - obousměrný přenos, ale NE současně
 - Například vysílačka, je nutné řízení
 - semafor – pomocí kterého se pozná kdo může vysílat, hrozí kolize



L1: Způsoby přenosu dat

- **Paralelní**
 - Přenášíme naráz více bitů prostřednictvím více vodičů
 - Typicky na kratší vzdálenosti
 - Například paralelní kabel k tiskárně
 - V počítačových sítích se příliš nepoužívá
- **Sériový přenos**
 - Přenášíme data bit po bitu jednou datovou cestou za sebou
 - Používám pouze jednu datovou cestu
 - Respektive mohou být dvě, pro vysílání a příjem
 - Typické pro počítačové sítě
 - Může dále dělit na
 - Synchronní
 - Asynchronní
 - Arytmický
 - **V dalších slidech budeme vždy uvažovat sériový přenos**



L1: Analogový a digitální přenos

- Analogový přenos
 - Analogový signál, který je fyzikálně měřitelný či rozpoznatelný
 - Hodnota napětí, proudu, světlo atd.
 - Je vždy specificky řešen pro dané prostředí a daný signál
 - **Může nabývat libovolných hodnot**
- Digitální přenos
 - Neřešíme konkrétní hodnoty veličin, ale stavy
 - **Může nabývat jen konkrétních stavů**
 - Tedy z vybraných hodnot analogového signálu odvodíme stav digitálního signálu
 - Přenášíme 0 a 1, ale můžeme je přenášet i více bitů naráz pomocí více stavů
 - Více stavů umožňuje přenést ne jen 0 a 1, ale např 00, 11, 01, 10, ...
 - Přenášenou jednotkou je 1 bit
 - Rychlost přenosu je b/s – počet bitů za vteřinu

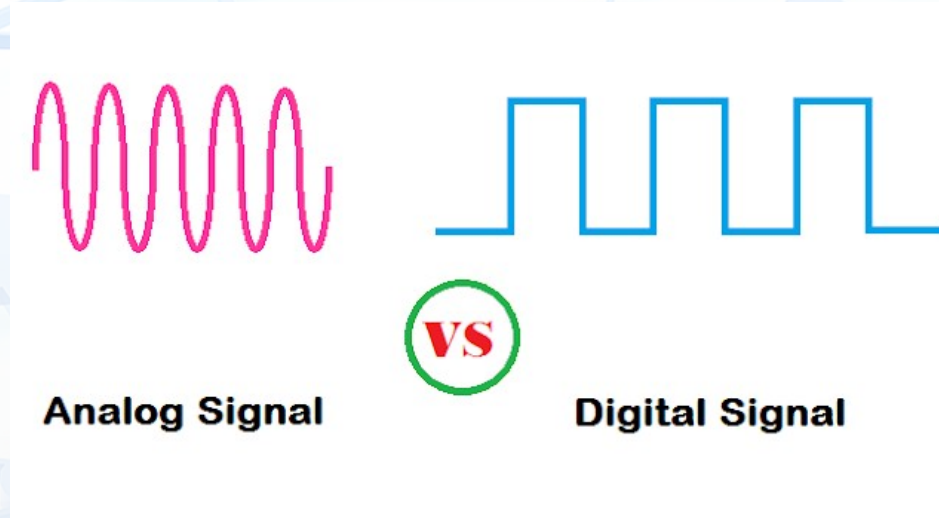
L1: Reálný přenos

- Co přenášíme
 - Analogový signál
 - Harmonický
 - Libovolný
 - Problém s popisem
 - Rádi bychom obdélníkový signál
 - Defakto jedničky a nuly – dobře by se četl
 - Fourierův rozklad / transformace / analýza

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos \omega_0 t + b_n \sin \omega_0 t)$$

$$a_0 = \frac{1}{T} \int_0^T f(t) dt$$

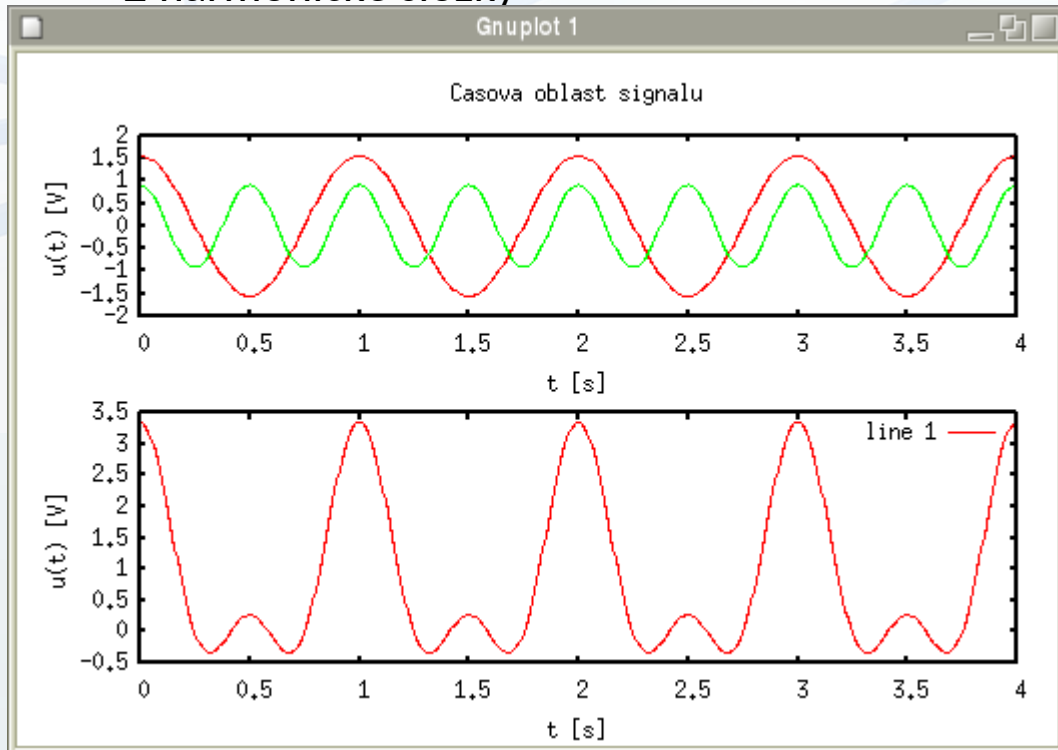
$$a_n = \frac{2}{T} \int_0^T f(t) \cos(n\omega_0 t) dt \quad b_n = \frac{2}{T} \int_0^T f(t) \sin(n\omega_0 t) dt$$



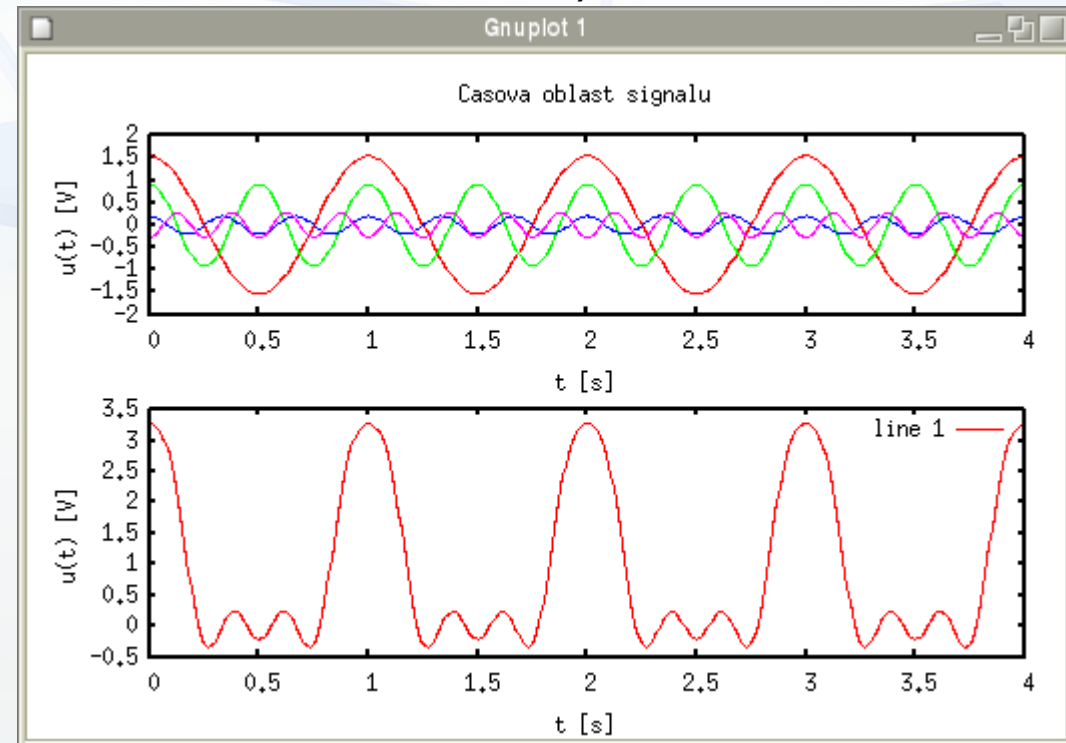
L1: Fourierův rozklad

- Umožňuje libovolný signál transformovat na součet harmonických signálů, které jsou celým násobkem základního signálu

2 harmonické složky

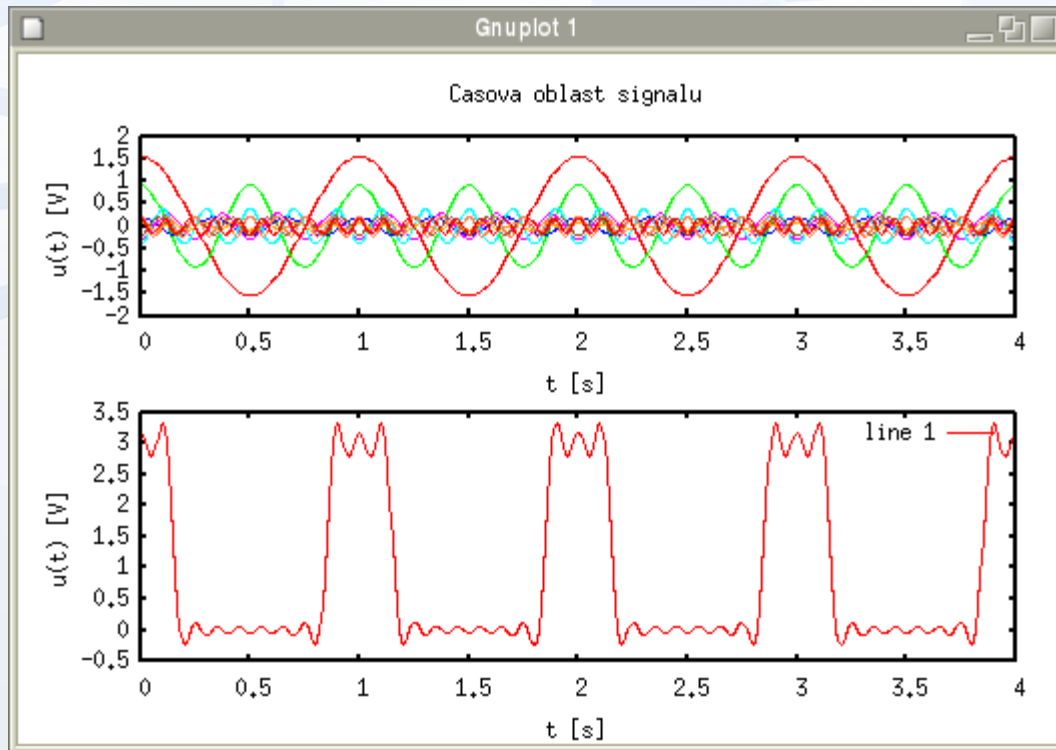


4 harmonické složky

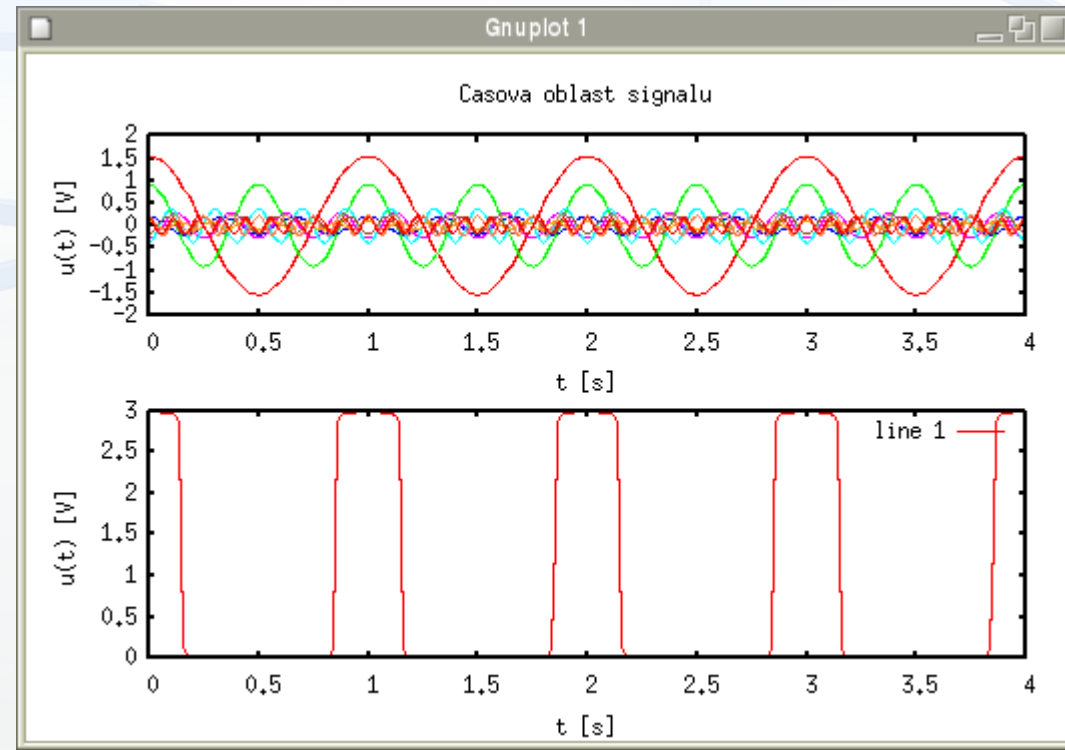


L1: Fourierův rozklad - pokračování

10 harmonických složek



100 harmonických složek



L1: Limitace šířkou pásma

- Šířka pásma
 - Rozmezí frekvencí, které je vodič schopen přenést
 - Signály s frekvencí mimo definovanou šířku pásma se nepřenesou vůbec
 - Tedy například v Fourierově transformaci musí přenášet méně přesný signál než bychom chtěli
 - Čím větší šířka pásma tím více harmonických signálů můžu přenášet
 - Čím na vyšší frekvence první harmonické, tím vyšší přenosová rychlost, ale nižší počet dalších harmonických

Bps	T (msec)	First harmonic (Hz)	# Harmonics sent
300	26.67	37.5	80
600	13.33	75	40
1200	6.67	150	20
2400	3.33	300	10
4800	1.67	600	5
9600	0.83	1200	2
19200	0.42	2400	1
38400	0.21	4800	0

L1: Modulační – baudová rychlost (Nyquistovo kritérium)

- Udává jak rychle jsme schopni měnit stavy/parametry signálu
- Je logicky závislá na šířce pásma
- Její hodnotu určuje Nyquistovo kritérium, které říká:
 - $v_m = 2 \cdot W$
 - v_m – modulační / baudová rychlost, jednotka jeden Baud [Bd]
 - W – šířka pásma, tedy rozdíl mezi nejvyšší a nejnižší přenositelnou rychlostí

L1: Modulační – baudová rychlost - příklad

- Máme přenosový kanál, který umožňuje přenášet frekvence od 100kHz do 1MHz, jaká je modulační rychlost ?
 - $W = 1.000.000 - 100.000 = 900.000 \text{ Hz}$
 - $v_m = 2 * W = 2 * 900.000 = 1.800.000 \text{ Bd}$
- Máme přenosový kanál, který umožňuje přenášet frekvence od 900kHz do 1MHz, jaká je modulační rychlost ?
 - $W = 1.000.000 - 900.000 = 100.000 \text{ Hz}$
 - $v_m = 2 * W = 2 * 100.000 = 200.000 \text{ Bd}$
- Čím větší je rozsah frekvencí (šířka pásma), kterou je přenosový kanál schopen přenést, tím větší je modulační rychlost.
- **!!! POZOR – změna modulační rychlosti neříká sama přímo nic o změně rychlosti přenosové !!!**

L1: Přenosová rychlost / kapacita přenosového kanálu

- Uvažujme **Přenosový Kanál Bez Šumu**
 - Pro zjednodušení – v realitě neexistuje
- Vyjdeme z Nyquistova kritéria – čím vyšší modulační rychlost, tím vyšší přenosová rychlost
 - Logicky čím rychleji můžu měnit stavy signálu, tím více dat za jednotku času přenesu
- Zároveň je ale třeba brát v potaz, že signál nemusí být binární, ale může přenášet více stavů, které můžeme reprezentovat jako 0,1,11,00,01,10 a tím přenosovou rychlost opět zvýšit
- Obecně platí $v_p = 2 * W * \log_2(V)$ [b/s], $v_p = v_m * \log_2(V)$
 - W – šířka pásma [Hz]
 - V – počet stavů
 - Čím větší je modulační rychlost nebo čím vyšší je počet stavů, tím vyšší je přenosová rychlost
 - Modulační rychlost je daná šířkou pásma a ta typicky nejde snadno měnit
 - Počet stavů můžeme přidávat snáze, ale ne neomezeně, protože musíme být stále schopni dva stavy od sebe bezpečně rozeznat

L1: Přenosová rychlost / kapacita přenosového kanálu - příklad

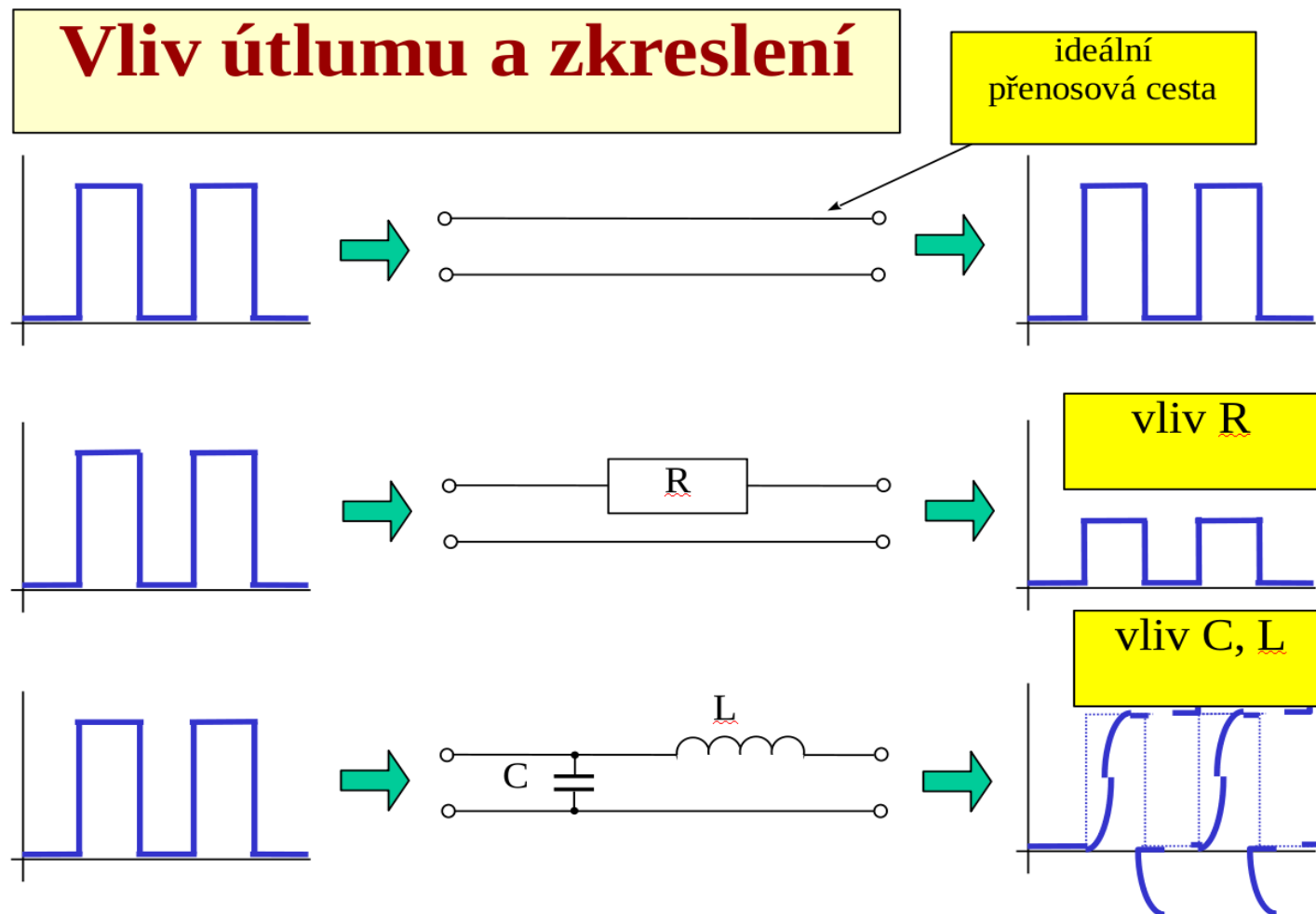
- Máme přenosový kanál bez šumu, který dokáže přenášet frekvence od 100Hz 1000Hz a ve kterém je možné identifikovat 4 úrovně signálu. Jaká je kapacita tohoto přenosového kanálu ?
 - $v_p = 2 * W * \log_2(V)$ [b/s]
 - $W = 1000 - 100 = 900$ Hz
 - $V = 4$
 - $v_p = 2 * W * \log_2(V) = 2 * 900 * \log_2(4) = 2 * 900 * 2 = 3.600$ b/s = 3.6 kb/s = 3.6kbps
- Jak se změní kapacita přenosového kanálu z předchozího příkladu, pokud se počet stavů zvýší o 4 ?
 - $v_p = 2 * W * \log_2(V)$ [b/s]
 - $W = 1000 - 100 = 900$ Hz
 - $V = 4 + 4 = 8$
 - $v_p = 2 * W * \log_2(V) = 2 * 900 * \log_2(8) = 2 * 900 * 3 = 5.400$ b/s = 5.4 kb/s = 5.4kbps
 - $Změna = 5400 / 3600 = 1.5$, přenosová rychlost se zvýší 1.5x

L1: Útlum, zkreslení a rušení

- V ideálním světě je přenos ovlivňován
 - Útlum
 - Vlivem vzdálenost – délky přenosového kanálu dochází ke snížení intenzity/kvality signálu
 - Vlivem prvků v zařízeních, kterými signál prochází dochází také k útlumu – například na odporu
 - Zkreslení
 - Vlivem prvků v zařízeních, kterými signál prochází dochází také k útlumu – například cívka či kondenzátor
 - Rušení
 - Vnější – může jít například o silové vedení v blízkosti nestíněného vodiče
 - Nemusí se projevovat konstantně, ale může se v čase měnit podle spotřeby
 - Vnitřní – i jednotlivé vodiče například v twistu se mohou ovlivňovat
 - Defakto každý vodič vedoucí elektřinu se zároveň chová jako anténa i vysílač
 - Proto se páry v twisty kříží aby se rušení eliminovalo
 - Další možností je použití stíněných vodičů (uvnitř i vně vodiče – stínění mohou mít jednotlivé vodiče, páry i celý svazek)
- Čím delší je datové cesta – vzdálenost, kterou musejí data urazit - tím se tyto negativní vlivy projevují více

L1: Rušení a útlum - příklady

Vliv útlumu a zkreslení



L1: Shannonovo kritérium

- V ideálním světě je přenos ovlivňován
 - Rušení, útlum, zkreslení ...
- Kapacita přenosového kanálu bude ovlivněna kvalitou daného kanálu
 - Tedy jak věrně/bezpečně dokáže data přenést a tím i jak lze rozlišit jednotlivé stavy
- Pro reálný přenosový kanál platí Shannonovo kritérium
 - $v_p = W \cdot \log_2(1 + S/N)$ [b/s]
 - W – šířka pásma [Hz]
 - S – úroveň signálu
 - N – úroveň šumu
- V reálu je tedy zásadnější kvalita signálu a přenosového media a tedy nízké hodnota šumu / rušení
 - Signál je třeba zesilovat
 - Vodič je třeba odstínit od okolí, ale i od ostatních žil v rámci daného vodiče – například v twistu

L1: Shannonovo kritérium - příklad

- Mějme přenosový kanál, ve kterém je poměr signálu a šumu 1000:1 a šířka pásma, kterou kanál dokáže přenášet je 3.1KHz (telefonní linka v optimálním případě). Jaká je kapacita takového kanálu?
 - $v_p = W \cdot \log_2(1 + S/N)$ [b/s]
 - $W = 3100$ Hz
 - $S = 1000$, $N = 1 \Rightarrow S/N = 1000/1 = 1000$
 - $v_p = 3100 \cdot \log_2(1 + 1000) = 3100 \cdot 9,97 = 30.907$ b/s = 30,907 kbps
- Jak se změní kapacita kanálu, pokud se úroveň šumu 1x zvýší ?
 - $v_p = W \cdot \log_2(1 + S/N)$ [b/s]
 - $W = 3100$ Hz
 - $S = 1000$, $N = 2 \Rightarrow S/N = 1000/2 = 500$
 - $v_p = 3100 \cdot \log_2(1 + 500) = 3100 \cdot 8,97 = 27.807$ b/s = 27,807 kbps
 - **Změna = $27.807/31.907 = 0,89$, přenosová rychlost klesne o 11%**

L1: Zesílení a decibely

- Decibel je logaritmická jednotka určující poměr dvou veličin
 - Základem není Bel ale DeciBel – ve vzorci je třeba násobit výsledek deseti
 - Může sloužit k popisu změny – například signálu, kde můžeme měřit zesílení nebo zeslabení např výkonu, proudu nebo napětí – označujeme jako A_x – kde x specifikuje zdroj poměru
 - **Výkonový:** $A_p = 10 \cdot \log_{10}(P_{out}/P_{in})$ [dB]
 - (10x právě proto, že výsledek je v deciBelech)
 - **Proudový:** $A_I = 20 \cdot \log_{10}(I_{out}/I_{in})$ [dB]
 - (20x proto, výkon je úměrný kvadrátu intenzity pole - případně napětí či proudu)
 - **Napěťový:** $A_U = 20 \cdot \log_{10}(U_{out}/U_{in})$ [dB]
 - (20x proto, výkon je úměrný kvadrátu intenzity pole - případně napětí či proudu)

L1: Zesílení a decibely

příklad

- Jaké je bude zesílení/zeslabení pokud vstupní výkon bude 10mW a výstupní bude 5mW ?
 - $A_p = 10 \cdot \log_{10}(P_{out}/P_{in})$
 - $P_{in} = 10 \text{ mW}$
 - $P_{out} = 5 \text{ mW}$
 - $A_p = 10 \cdot \log_{10}(5/10) = 10 \cdot -0,30 = -3\text{dB} \Rightarrow$ dojde k útlumu o 3dB (~ na polovinu)
- Jaké je bude zesílení/zeslabení pokud vstupní výkon bude 5mW a výstupní bude 15mW ?
 - $A_p = 10 \cdot \log_{10}(P_{out}/P_{in})$
 - $P_{in} = 5 \text{ mW}$
 - $P_{out} = 15 \text{ mW}$
 - $A_p = 10 \cdot \log_{10}(15/5) = 10 \cdot 0,48 = 4,8\text{dB} \Rightarrow$ dojde k zesílení signálu o 4,8dB

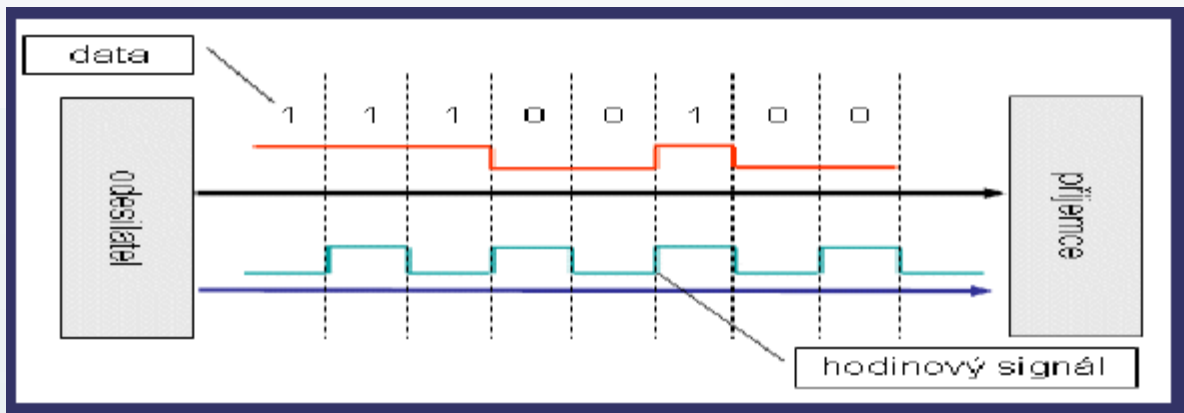
L1: Shannonovo kritérium a zeslabení signálu - příklad

- Jaké je bude kapacita kanálu, pokud šířka pásma bude 3.1kHz a zesílení/zeslabení signálu bude 30dB ?
 - $A_p = 10 \cdot \log_{10}(P_{out}/P_{in})$, $v_p = W \cdot \log_2(1+S/N)$ [b/s]
 - Poměr signálu a šumu se rovná poměru vstupního a výstupního výkonu signálu
 - $S/N = P_{out}/P_{in} = 10^{(A_p/10)} = 10^3 = 1000$
 - Protože $y = \log_a(x) \iff a^y = x$
 - $v_p = W \cdot \log_2(1+S/N) = 3100 \cdot \log_2(1+1000) = 3100 \cdot 9,97 = 30,907 \text{ kbps (kb / s)}$
 - Poměr signálu a šumu tedy můžeme udávat i ve formě dB

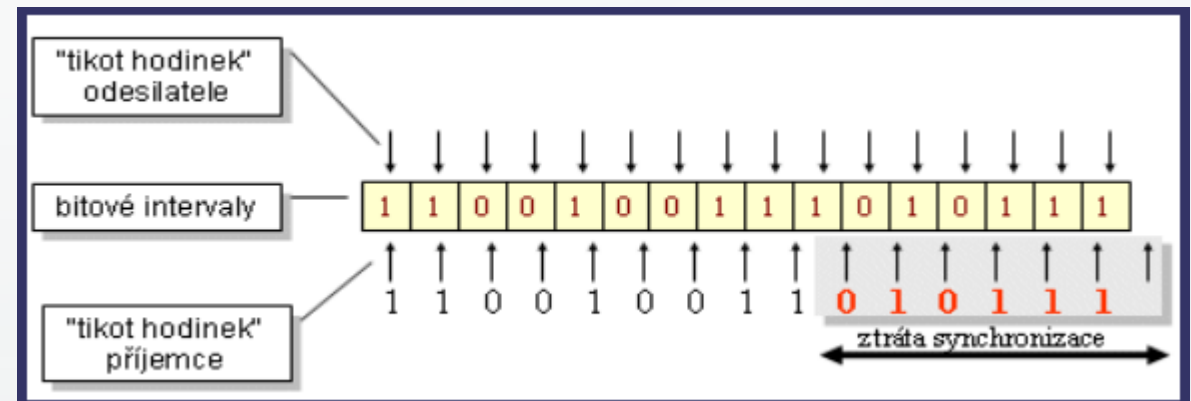
L1: Synchronizace

- Signál může mít jeden či více stavů které od sebe potřebuje přesně rozeznat
- Data (0,1) jsou posílána signálem v „pulzech“ - každý pulz přenáší jeden stav
- Pulz – tik hodin – musí mít konstantní délku a musí jí znát vysílač i přijímač
 - Logicky, jinak bychom více stavů mohli interpretovat jako jeden či naopak jeden stav jako více stavů

Ideální přenos – odesílatel a příjemce jsou synchronní

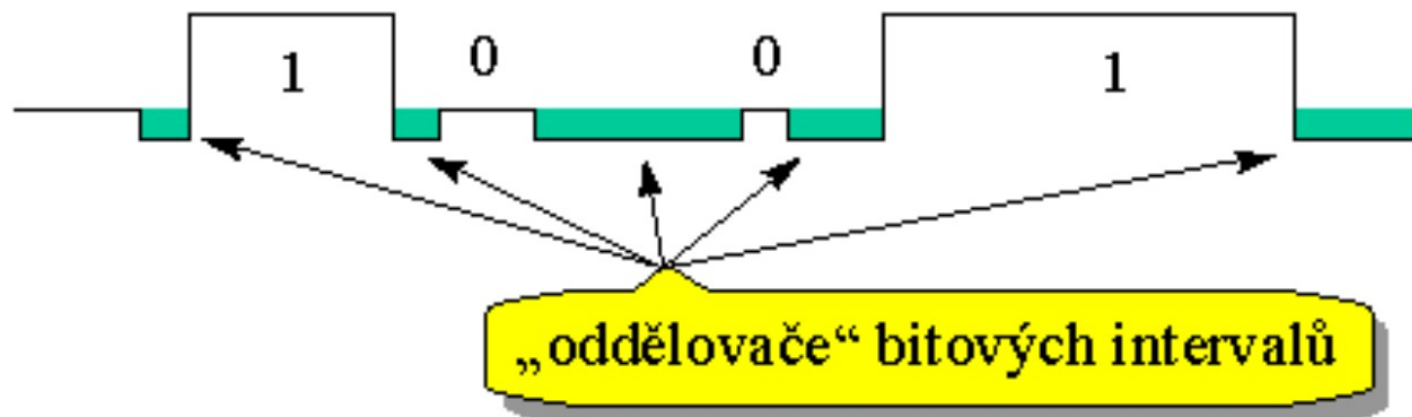


Problém v desátém pulzu



L1: Synchronizace: Asynchronní přenos

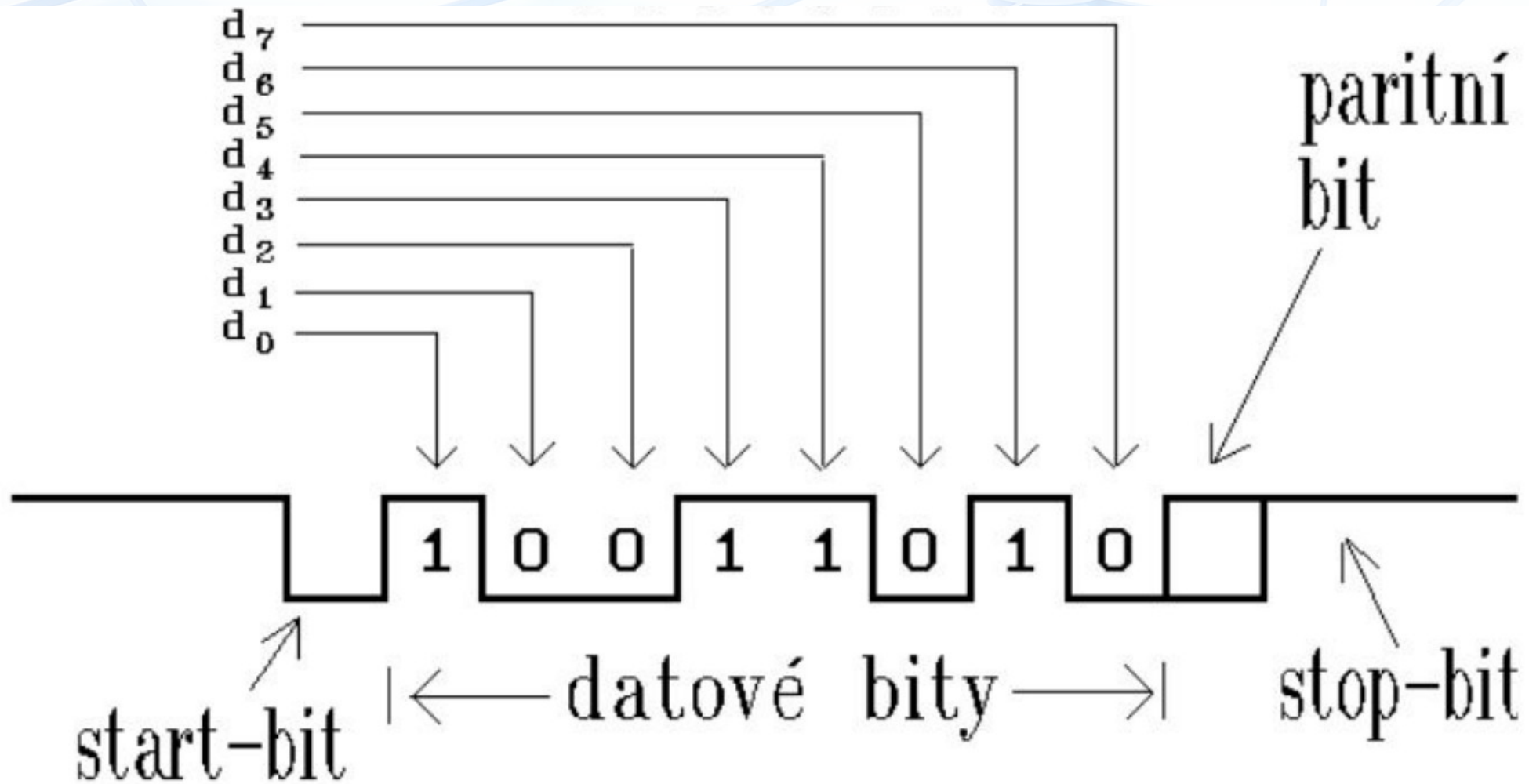
- Zde jednoduše žádná trvalá synchronizace není
- **Vysílač může vysílat v různé časy - neřídí se hodinami ...**
- **Délka pulzu se může pro jednotlivé pulzy lišit**
- Může to vůbec fungovat ?!
- Může, ale je nákladné, protože to vyžaduje **tří stavovou logiku** - min tři stavy
 - Dva stavy na přenos požadovaných dat 0 a 1
 - Třetí stav pro informaci, zda se začíná/končí přenos dat
- Výhodou je, že nepožaduje žádnou synch
- V praxi se příliš nepoužívá



L1: Synchronizace: Arytmický přenos

- „Speciální“ varianta asynchronního přenosu
- **Vysílač může vysílat v různé časy – neřídí se hodinami ...**
- **Délka pulzu (jednoho bitu) je pevně daná**
- Vycházíme z předpokladu, že trvalou synchronizaci se udržet nepovede, ale na omezenou dobu ano
 - Jinak řečeno na dlouhé přenosy se čas rozjede, ale na krátké jej dokážeme udržet „synchronní“
 - Možná ne dokonale, ale tak aby se nepřekročila hranice bitů a to stačí
- Potřebujeme nějak zajistit prvotní synchronizaci
 - To by zas šlo extra vodičem, ale to jsme u synchronní varianty
- Doplníme signál o „start“ a „stop“ značky/bity
 - Pozor nejedná se o jeden bit – jen se to tak nazývá !!!
- Princip fungování
 - Posloucháme komunikaci a neděláme nic, dokud nepřijde „start bit“
 - Od přijetí start bitu začnu s dohodnutou frekvencí vzorkovat data – rozpoznávat jednotlivé bity
 - Úvodní synchronizaci nám zajistil start bit a předpokládáme, že během krátkého přenosu se hodiny udrží synchronní
 - Pokud nastane i přesto chyba, na L1 to nepoznáme, ale zjistí to L2 – řekneme si později
 - Počet dat může být pevně stanoven a nebo může být dohodnuta ukončovací značka – stop bit

L1: Synchronizace: Arytmický přenos příklad



L1: Synchronizace: Arytmický přenos II.

Paritní bit

- Paritní bit slouží k základnímu ověření správnosti přenosu
- Paritní bit doplňuje bitovou posloupnost o jeden bit, který doplňuje počet jedniček v posloupnosti na lichý nebo sudý počet
- Parity tedy mohou v základ být dvě
 - Sudá parita => 1101|**1** (datové jedničky jsou tři, aby jich byl sudý počet musím jednu doplnit)
 - Lichá parita => 1101|**0** (datové jedničky jsou tři, což je liché, tedy v paritním bitu nedoplňuji nic)
- Jednoduchá implementace, protože se jedná jen o jednoduchý čítač počtu
- Nemusí odhalit násobnou chybu
 - Přenáším se sudou paritou: 1101|**1** (doplnil jsem 1 na sudý počet)
 - Dorazilo mi: 1000|1 – výsledná parita parita je 1 a tedy souhlasí, ALE v datech jsou dvě chyby
 - Stejně tak nepozná, zda chyba byla jen jedna nebo více
 - Důkladnější zabezpečení se realizuje až na L2 – řekneme si následně

L1: Synchronizace: Arytmický přenos IV.

Využití kapacity přenosového kanálu

- U arytmičkého přenosu jsme se poprvé potkali s tím, že kromě dat, která chceme přenášet, přenášíme navíc i data „řídící“
- Tyto data „navíc“ my vlastně přenášet nechceme, ale musíme aby se přenos realizoval
- Jedná se o
 - Start bit
 - Stop bit
 - Paritní bit
- Tato data nám „snižující“ kapacitu přenosového kanálu – opticky nám snižují přenosovou rychlost, protože „zabírají“ místo informačním datům
- Využití kapacity přenosové kanálu může spočítat jako :
 - $n = N/M * 100$ [%]
 - N – počet datových bitů – ty co opravdu chceme přenášet
 - M – počet všech bitů, které musíme přenášet

L1: Synchronizace: Arytmický přenos IV.

Využití kapacity přenosového kanálu

příklad

- Příklad:

Mějme arytmičtý přenos s délkou informačních dat 8 bitů, s 1 start bitem, jedním paritním bitem a dvěma stop bity. Jaké je využití kapacity přenosového kanálu ?

$$n = N/M$$

$N = 8$ bitů (ty chceme přenášet)

$M = 8 + 1 + 1 + 2 = 12$ (datový bity + start bit + paritní bit + dva stop bity)

$$n = 8/12 = 0,6667 = 66,67\%$$

Tedy náš arytmičtý přenos spotřebuje 1/3 kapacity přenosového kanálu.

L1: Synchronizace: Arytmický přenos V.

Využití kapacity přenosového kanálu versus délka přenosu

- Z příkladu využití kapacity přenosového kanálu je jasné, že bychom rádi co nejdelší přenos, aby využití kanálu bylo co nejlepší
 - Pokud budeme mít 1 start bit, 1 paritní bit a jeden stop bit a délku dat **8bit**
 - Využití kapacity kanálu bude $n = N/M$, $N = 8 \text{ b}$, $M = 8 + 1 + 1 + 1 = 11 \text{ b}$, $N = 8/11 = \mathbf{72,72\%}$
 - Pokud budeme mít 1 start bit, 1 paritní bit a jeden stop bit a délku dat **16bit**
 - Využití kapacity kanálu bude $n = N/M$, $N = 16 \text{ b}$, $M = 16 + 1 + 1 + 1 = 19 \text{ b}$, $N = 16/19 = \mathbf{84,42\%}$
 - Čím delší budou přenášená data, tím bude vyšší využití kapacity přenosového kanálů
 - ALE zároveň čím delší budou přenášená data tím vyšší bude pravděpodobnost chyby/ztráty synchronizace v rámci arytmičského přenosu
 - Chceme tedy co nejdelší datovou část, ale NE delší než by došlo ke zvýšení pravděpodobnosti chyby nad námi stanovenou mez
 - Pravděpodobnost úspěšného přenosu N bitů: $P_N = p_1^N$
 - p_1 – pravděpodobnost úspěšného přenosu 1 bitu
 - q_1 – pravděpodobnost chyby v přenosu 1 bitu, $q_1 = 1 - p_1$

L1: Synchronizace: Arytmický přenos V.

Využití kapacity přenosového kanálu versus délka přenosu příklad

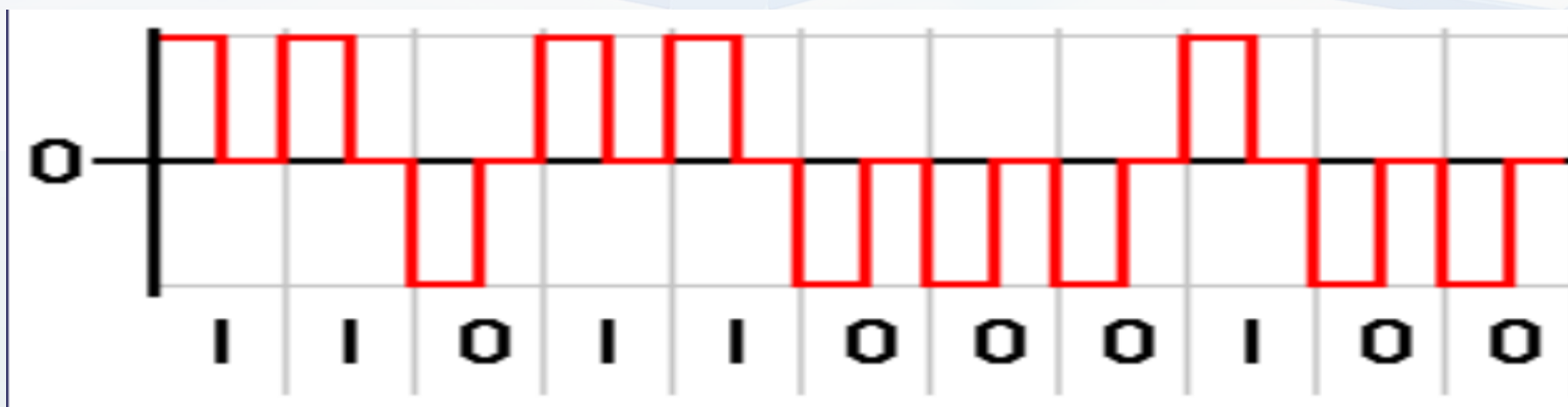
- Mějme Symetrický Binární Přenosový Kanál Bez Paměti, kolik můžeme přenést bitů v rámci jediného přenosu, aby pravděpodobnost bezchybného přenosu neklesla pod 99%, pokud pravděpodobnost bezchybného přenosu 1 bitu je 99.99%
 - $P_N = p_1^N$
 - $P_N = 99\% = 0,99$, $p_1 = 99,99\% = 0,9999$, $N = ?$
 - $N = \log_{p_1}(P_N) = \log_{0,9999}(0,99) = 100,4983 = \mathbf{100b}$ (nemůžeme jít nad 100b)
- Pokud budeme chtít mít pravděpodobnost bezchybného přenosu neklesla pod 99,9, kolik bitů můžeme přenést ?
 - $P_N = p_1^N$
 - $P_N = 99,9\% = 0,999$, $p_1 = 99,99\% = 0,9999$, $N = ?$
 - $N = \log_{p_1}(P_N) = \log_{0,999}(0,99) = 10,0045 = \mathbf{10b}$ (nemůžeme jít nad 10b)

L1: Synchronizace: Synchronní přenos

- Ideální stav
- Odesílatel a příjemce mají „nějak“ zajištěnou trvalou synchronizaci hodin
 - POZOR nejedná se reálný čas NTP – jde o délku a start pulzu
- Pro zajištění trvale synchronizace může být použit extra vodič s hodinovým pulzem
 - Bude těžko / drahé realizovatelné na velké vzdálenosti
- Může modifikovat signál tak, aby v sobě kromě požadovaného rozlišení bitů s hodnotou 0 a 1 přenášel i informaci o hodinách
 - Může se jednat o místo kde je možné se sesynchronizovat (NRZ)
 - Může se jednat přímo o signál, který v sobě nese hodiny (RZ, Manchester)
- Nedochozí k žádné změně ani doplnění dat – jde „pouze“ o změnu toho, jakým, signálem jsou data přenášena
- Tomuto způsobu zajištění synchronizace se říká „kódování“

L1: Kódování signálu: RZ

- RZ - Return to Zero
- Patrně nejjednodušší kódování
- 0 a 1 jsou reprezentovány kladnými a zápornými pulzy
- Třetí stav je „nulový“ nebo „neutrální“ a k němu se signál po každém pulzu vrací
- „Nese“ v sobě hodiny, neboť ke změně dochází v každém pulzu bez ohledu na data



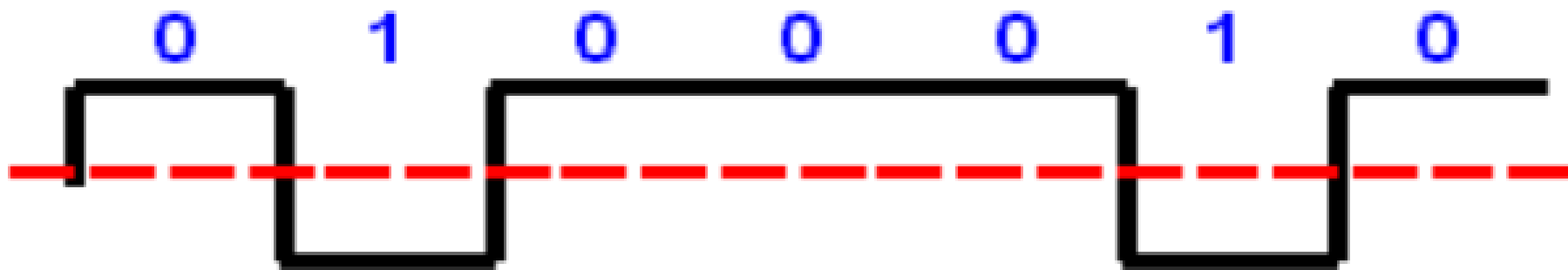
L1: Kódování signálu: RZI

- RZI – Return To Zero Inverted
- Používá už jen dvě úrovně
 - 0 – kratší změna signálu než je délka hodin (ale je tam nějaký pulz)
 - 1 – bez změny signálu
- Hodiny u v sobě přímo nemá – například pro posloupnost jedniček nedochází ke změně signálu
- Pokud v signálu je 0 je možné hodiny synchronizovat na konci pulzu
- Odolný vůči změně polarity



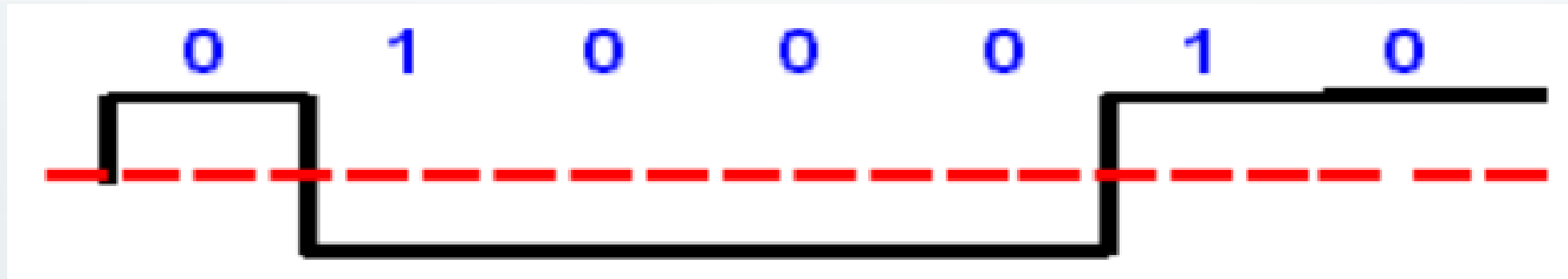
L1: Kódování signálu: NRZ

- NRZ – Non Return To Zero
- Používá jen dva stavy
 - 0 - například záporný pulz
 - 1 - například kladný pulz
- Mezi jednotlivými stejnými bity (více nul nebo více jedniček) nedochází ke změně
- **Není možné použít k synchronizaci hodin** a je třeba řešit synchronizaci jinak - externě



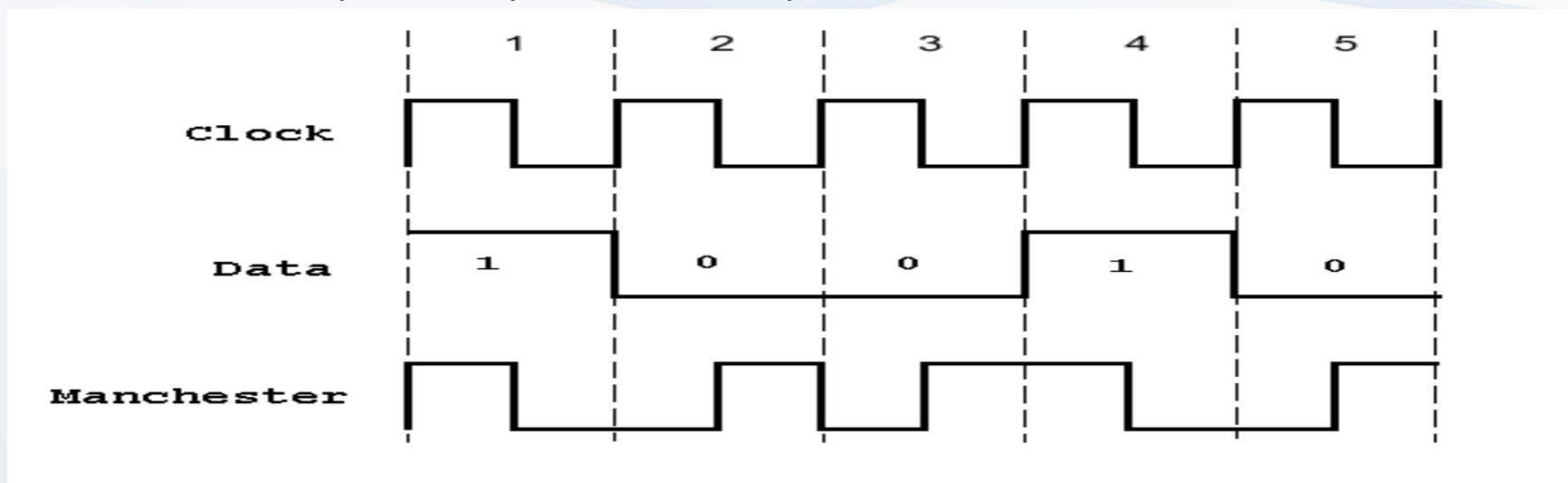
L1: Kódování signálu: NRZI

- NZRI – Non Return To Zero Inverted
- Má jen dvě úrovně signálu, ale ty přímo nesignalizují 0 a 1
- Bity 0 a 1 jsou rozlišeny:
 - 0 – nenastává změna v signálu (je jedno zda je právě v kladné či záporné hodnotě signálu)
 - 1 – nastává změna stavu na začátku pulzu
- **Může sloužit k synchronizaci hodin** – s každou jedničkou vím kde jsou hodiny
- **Je odolný vůči změně polarity** – hodiny nejsou definované konkrétním stavem, ale změnou či zachováním stavu



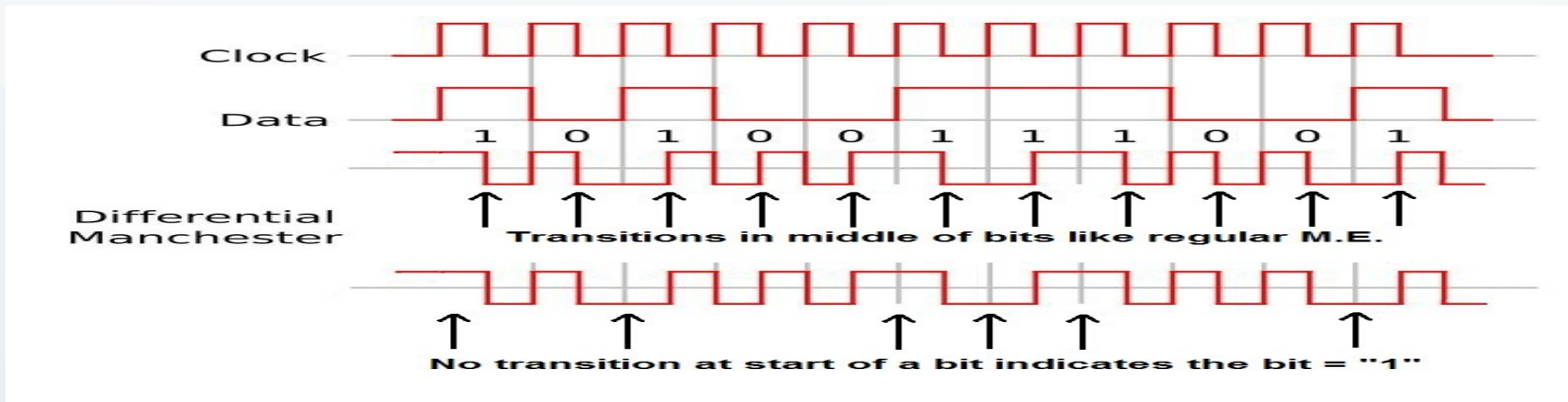
L1: Kódování signálu: Manchester

- Manchester má opět jen dva stavy
- Rozdělení nuly a jedničky se odečítá uprostřed pulzu, kde VŽDY musí nastat změna
 - **Můžeme dobře synchronizovat hodiny**
- Hodnoty (například, může být i otočená varianta)
 - 0 – změna signálu z kladného na záporný
 - 1 – změna pulzu ze záporného na kladný



L1: Kódování signálu: Diferenciální Manchester

- Diferenciální Manchester opravu nedostatky klasické verze
- Opět má jen dva stavy a vždy uprostřed pulzu dojde ke změně stavu (nezávisle na datech) - **přímo v sobě nese hodiny.**
- Data jsou odečítána na vstupní hraně pulzu a to následovně:
 - 0 – na vstupní hraně pulzu nastala změna (nezáleží na směru)
 - 1 – na vstupní hraně pulzu nenastala změna
- **Je odolný vůči změně polarity**



L1: Kódování signálu: Klasický a Diferenciální Manchester

