UPS 2016/2017

- Jindřich Skupa
- http://home.zcu.cz/~skupaj/
- skupaj@kiv.zcu.cz
    - UN305
    - St: 8:25-9:10
    - Pa:8:25-9:10
- Provozní řád laboratoře

Zápočet

- 80% účast na cvičení
- Semestrální práce
  - Server v C, klient aplikace v Javě
  - Max 30b, Min 15b, Bonus 10b
  - Bodová penalizace za pozdní odevzdání
    - -1 za každý den
- Test cca v 2/3 semestru
  - Max 20b, Min 10b
  - Jeden opravný termín

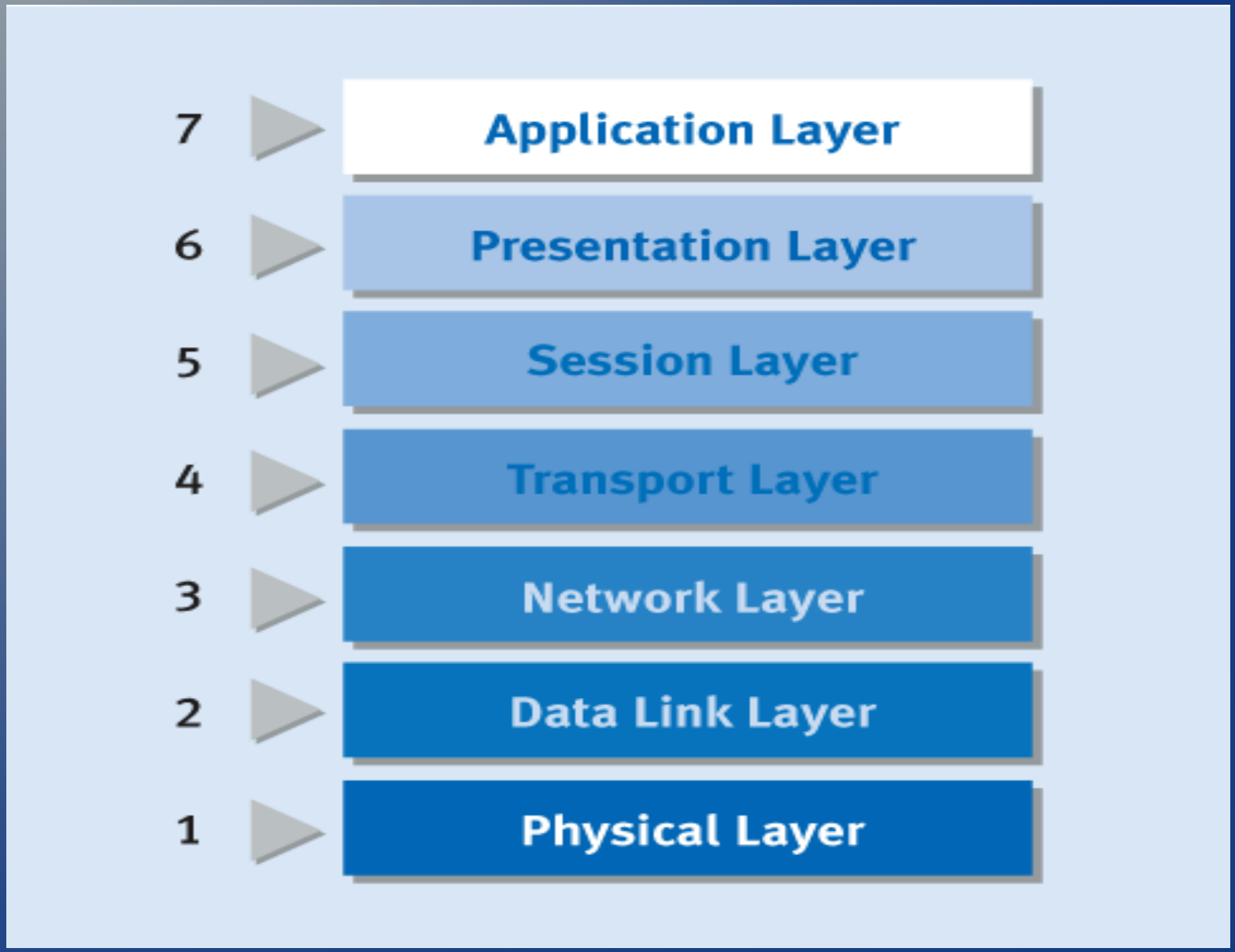
První cvičení

- Co jsou sítě
- ISO/OSI model
- Adresy MAC, IP
- Linux/Unix
  - Přihlášení, základní orientace
  - Nastavení sítě
  - Diagnostika sítě

Počítačové sítě

- Množina komunikujících zařízení
- Heterogenní prostředí
  - HW architektura, OS, Software
  - Přenosové médium
  - NIC
- Protokoly a standardy
  - IEEE
  - ISO/OSI
  - RFC

ISO/OSI I.


7
Application Layer
6
Presentation Layer
5
Session Layer
4
Transport Layer
3
Network Layer
2
Data Link Layer
1
Physical Layer

ISO/OSI II.

Aplikační (7)

obecné a speciální služby pro aplikace, např. přenos souborů, terminál, …

Prezentační (6)

Převod aplikačních dat na data vhodná pro přenos (heterogenita, komprese, šifrování)

Relační (5)

Řešení problému chyb nad přenosovými protokoly (výpadek spojení)

Transportní (4)

Přizpůsobení různorodých síťových služeb potřebám aplikace (řešení chyb)

Síťová (3)

Přenos dat mezi koncovými uzly sítě (směrování, adresování, řízení toku dat)

Linková (2)

Přenos dat mezi sousedními uzly sítě (zabezpečení proti chybám)

Fyzická (1)

Definice signálů, konektorů, vedení, rychlostí, …

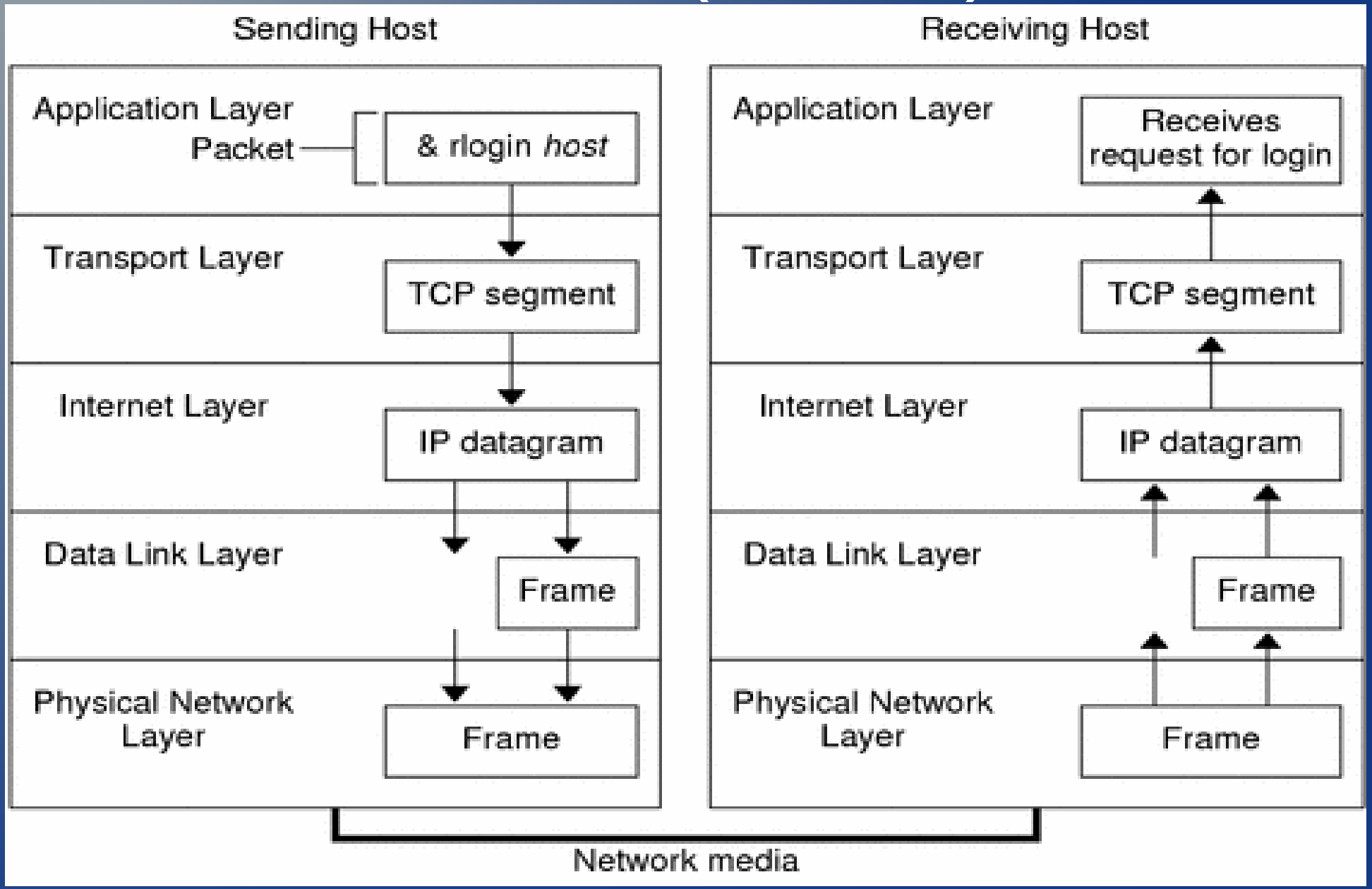
ISO/OSI III.

- Fyzická vrstva
  - Přenáší se bity
  - Zařízení: přenosové médium, konektory
- Linková vrstva
  - Přenáší se rámce
  - Zařízení: přepínač (switch, bridge)
- Síťová vrstva
  - Přenáší se packety
  - Zařízení: router, L3switch
- Transportní
  - Přenáší se: segmenty (TCP), datagramy (UDP)
  - Zařízení: jádro operačního systému

ISO/OSI (TCP I.)

TCP/IP	Model ISO/OSI
Aplikační vrstva	Aplikační vrstva
	Prezentační vrstva
	Relační vrstva
Transportní vrstva	Transportní vrstva
Síťová (IP) vrstva	Síťová vrstva
Vrstva síťového rozhraní	Linková vrstva
	Fyzická vrstva

ISO/OSI (TCP II.)



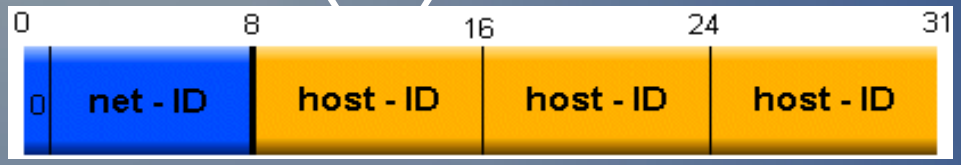
Sítě

- Podle velikosti
  - PAN, LAN, MAN, WAN
- Podle topologie
  - Sběrnice, kruh, hvězda, kombinace
- Podle způsobu komunikace
  - Přepínání kanálů
  - Přepínání zpráv

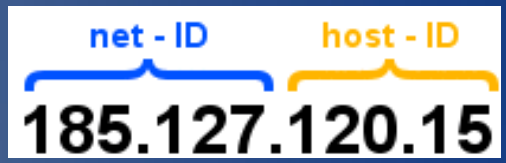
Adresy v síti

- ipv4 192.168.0.1/24
  - Veřejné, privátní, lokální, 32b
- ipv6 2A01:0430:003E::2/64
  - Globální, linková, lokální, 128b
- MAC f0:de:f1:42:40:da
  - Světově unikátní, 48b

Třídy IP adres

- Třída A (/8)
- 

- 0.0.0.0 až 127.255.255.255
- Třída B (/16)
- 



- 128.0.0.0 až do 191.255.255.255.
- Třída C (/32)
- 

- 192.0.0.0. až 223.255.255.255

CIDR

- Classless Inter Domain Routing
- Volná délka masky (/28, /20)
- Nejmenší smysluplná maska (/30)
- Adresa sítě (sudá, nejnižší)
- Všesměrová adresa (lichá, nevyšší)
- Adresy uzlů
- Bit +1 půlí síť
- Bit -1 sdružuje dvě sítě

Privátní adresní rozsahy

- IPv4 (tools.ietf.org/html/rfc1918)
  - 127.0.0.1/8 – localhost
  - 192.168.0.0/16
  - 172.16.0.0/12
  - 10.0.0.0/8
  - 169.254.0.0/16 – link-local
- IPv6 (tools.ietf.org/html/rfc4291)
  - ::1 – localhost
  - Fe80::/64 – link-local

Linux

- ifconfig (do 2.4.x), ip (od 2.6.x)
- route
- iptables
- ping, traceroute
- dig, host, whois
- netstat
- lsof
- nc

UPS 2015/2016

Cvičení 2

http://home.zcu.cz/~skupaj

Opakování / Co se nestihlo

- Několik otázek z/k předcházejícímu cvičení
- Doplnění z minula
- Dokončení Linuxových příkazů

Obsah

- TCP/IP model/zásobník
- Typy serverů
- Porty
- BSD sockety
- Paralelní procesy, select

TCP/IP

TCP/IP	Model ISO/OSI
Aplikační vrstva	Aplikační vrstva
	Prezentační vrstva
	Relační vrstva
Transportní vrstva	Transportní vrstva
Síťová (IP) vrstva	Síťová vrstva
Vrstva síťového rozhraní	Linková vrstva
	Fyzická vrstva

Server / Client

- Server
  - Způsob odbavení požadavku
    - Interaktivní – požadavky ve frontě a postupně odbavuji
    - Paralelní – při přijetí požadavku spouštím proces/vlákno
- Client
  - Program připojující se k serveru

TCP/IP

- Síťové rozhraní
  - Ethernet, PPP, SLIP
- Síťová
  - IP
- Transportní
  - TCP, UDP, ICMP,....
- Aplikační
  - Telnet, FTP, HTTP, DNS, ...

Server / Client

- Server
  - Program běžící na serveru, poslouchá na portu (v UNIX systémech démon)
  - Typ spuštění – stavové / bezstavové
    - Stavové servery
      - SSH, APACHE
    - Bezstavové - Internet Daemon
      - FTP, NTP
    - Pozor – neplést se službami
  - Udržování spojení
    - S udržovaným spojením TCP - SSH
    - Bez udržovaného spojení UDP - DNS

Porty

- Porty
  - Definují aplikaci v rámci konkrétního stroje
  - Celé číslo v rozmezí 0 – 65535
  - /etc/services, netstat -ln
- Tři skupiny:
  - Dobře známé / privilegované (jen root)
    - 0-1024 – běžné služby, SSH/22, FTP/21, HTTP/80
  - Registrované
    - 1024-49151 Registrované u ICANN, MySQL/3306
  - Dynamické a soukromé
    - 49152-65535 Nejsou určena jejich použití

BSD Sockety

- BSD sockety
  - Dostupné na většině OS: Linux, *BSD, Windows(winsock)
  - Komunikační mechanismus jako soubor
- Atributy socketu
  - Domain
    - AF_UNIX, AF_INET, AF_ISO, ...
  - Type
    - SOCK_STREAM, SOCK_DGRAM
  - Protocol
    - Většinou se nevybíra, default 0

BSD Sockety ukazka

- AF_UNIX
  - Pojmenované sockety, pouze v rámci jednoho stroje, např. /tmp/mysql.sock
- AF_INET
  - TCP
    - socket(), bind(), listen(), accept
  - UDP
    - socket(), bind(), recvfrom(), sendto()
- Překlad adresy i portů
  - inet_addr
  - htonl, htons, ntohl, ntohs

Paralelní procesy

- Detailně v ZOS
- Pro paralelní obsloužení více klientů
- Vlákna - knihovna pthread
  - pthread_create()
- Procesy - fork()
  - Mění se jen server
  - fork()
    - fork() == 0 potomek
    - fork() != 0   rodič

Select

- Systémové volání
- V případě, že potřebujeme pasivně čekat
  - vstup / výstup / chybu
  - aktivita socketu
- int select(int n, fd_set *readfds, fd_set *writefds, fd_set *exceptfds, struct timeval *timeout);

Otázky

- Uveďte rozdělení počítačových sítí podle rozlehlosti. Uveďte i jejich další vlastnosti.
- Rozdíl mezi dvoubodovými a mnohabodovými spoji, výhody, nevýhody, použití.
- Nakreslete sběrnicovou a kruhovou topologii počítačové sítě, vysvětlete princip přenosu dat a řízení přenosu (sdílení komunikačního média)
- Sdílení komunikačního média, sítě s přepínáním kanálů, zpráv/paketů.
- Znázorněte rozdíl při přenosu dat přes mezilehlý uzel.
- Co je to úrovňová architektura, jaké má výhody a nevýhody, kde se obecně používá.
- Vysvětlete, co v referenčním modelu ISO znamenají pojmy úroveň nebo vrstva, n-tita, služba, protokol, datová jednotka n-té vrstvy a přístupový bod.

Otázky

- V sedmiúrovňovém modelu ISO/OSI vyjmenujte jednotlivé vrstvy od nejnižší po nejvyšší a vyjmenujte jejich funkci při přenosu dat.
- Která vrstva zajišťuje směrování v síti
- Která vrstva zajišťuje převod logického signálu na napětí
- Která vrstva zajistí, aby byla data přenesena bezchybně mezi sousedními uzly
- Zakreslete schematicky model TCP/IP, vysvětlete význam jednotlivých vrstev a uveďte příklady protokolů.
- Porovnejte referenční model ISO/OSI s modelem TCP/IP. Které vrstvy v modelu TCP/IP chybí a jak jsou nahrazovány.
- Uveďte základní aplikační protokoly TCP/IP.
- Co znamená zkratka TCP a co IP. Kde se TCP/IP používá.

Otázky

- Co jsou to spojované a nespojované služby. Kterým protokoly jsou v zásobníku TCP/IP realizovány
- Uveďte výhody a nevýhody spojovaných služeb. Kdy (v jakých typických aplikacích) se zejména používají
- Uveďte výhody a nevýhody nespojovaných služeb. Kdy (v jakých typických aplikacích) se zejména používají.

UPS 2015/2016

Cvičení 3

Opakování / Co se nestihlo

- Několik otázek z/k předcházejícímu cvičení
- Doplnění z minula

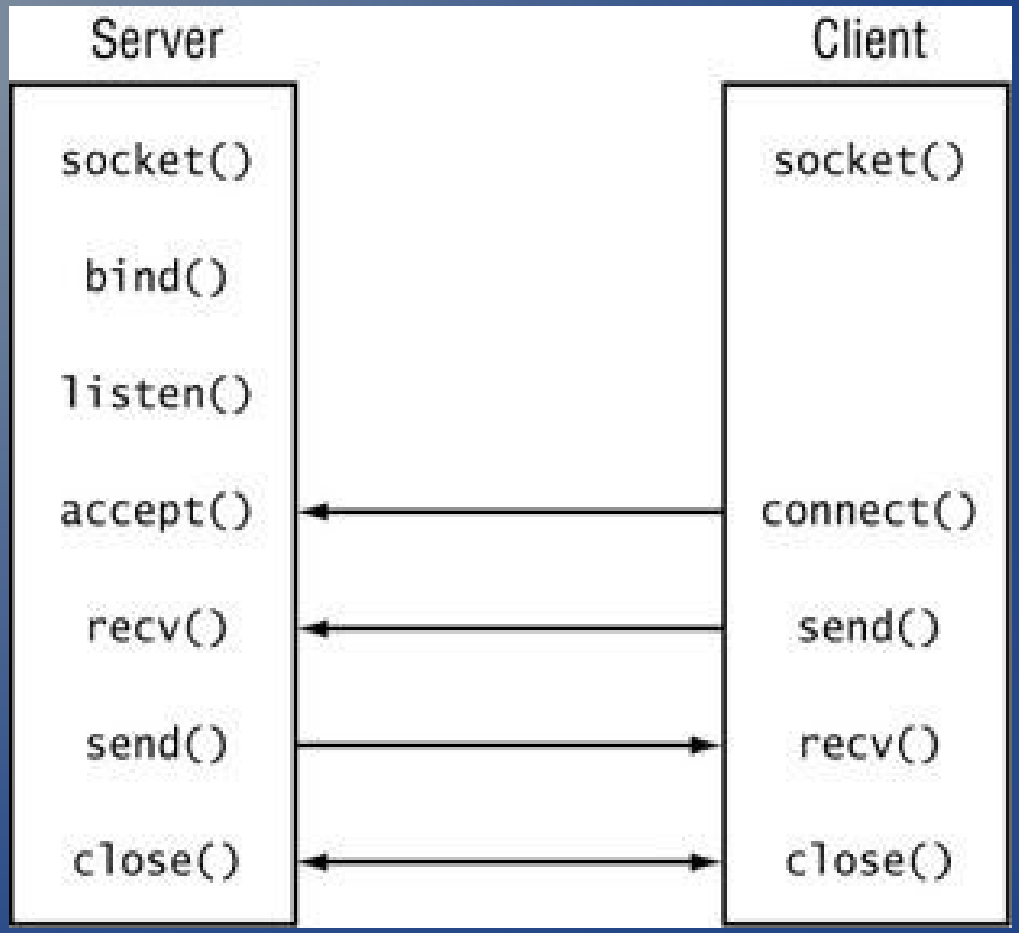
Obsah

- Server/klient v C pod GNU/Linuxem
- C
- Funkce
- Makefile

C

- Překladač: gcc
- Editor: vim, joe, nano
- Automatický překlad: Makefile / make
- Manuál: man
- #include <sys/types.h>
- #include <sys/socket.h>

Funkce - TCP/IP



Funkce UDP/IP

- Socket()
- Bind()
- Recvfrom()
- Sendto()

Makefile

```
CC=gcc
all:    clean client-unix server-unix
client-unix:
        ${CC} -o client-unix client-unix.c
server-unix:
        ${CC} -o server-unix server-unix.c
clean:
        rm -f client-unix
        rm -f server-unix
        rm -f server_socket
```

UPS 2015/2016

Cvičení 4

Opakování / Co se nestihlo

- Několik otázek z/k předcházejícímu cvičení
- Doplnění z minula

Obsah

- Server/klient v Javě
- Java
- Funkce
- Build.xml

Java

- Překladač: javac (Sun/Oracle, IBM, OpenJDK)
- Editor: Eclipse, NetBeans, vim, joe, nano
- Automatický překlad: build.xml / ant
- Manuál:
  http://docs.oracle.com/javase/tutorial/networking/sockets/index.html
  http://docs.oracle.com/javase/tutorial/networking/overview/networking.html
- Import java.net.*;
- Import java.io.*;

Funkce TCP/IP

- Třída ServerSocket
  - ServerSocket(10001, 10, InetAddress.getByName("localhost"));
  - serverSocket.accept();
- Třída Socket
  - Socket("127.0.0.1", 10001);
- Streamy
  - socket.getInputStream()
  - socket.getOutputStream()

Funkce UDP/IP

- Třída DatagramSocket
  - DatagramSocket(10000);
- Třída DatagramPacket
  - DatagramPacket(buffer, buffer.length);
- Vstup/Výstup
  - DatagramSocket.send
  - DatagramSocket.receive

build.xml

```
<project name="mydemo" default="build" basedir=".">
<target name="build" depends="server,client">
    <echo message="${basedir}"/>
</target>
<target name="server" depends="">
    <echo message="Prekladam server"/>
    <exec executable="javac" dir="${basedir}">
        <arg value="serverTCPSingle.java"/>
    </exec>
</target>
<target name="client" depends="">
    <echo message="Prekladam clienta"/>
    <exec executable="javac" dir="${basedir}">
            <arg value="clientTCP.java"/>
    </exec>
</target>
<target name="clean" depends="">
    <echo message="Mazu binarky..."/>
    <exec executable="rm" dir="${basedir}">
            <arg value="-f"/>
        <arg value="serverTCPSingle.class"/>
            <arg value="clientTCP.class"/>
    </exec>
</target>
</project>
```

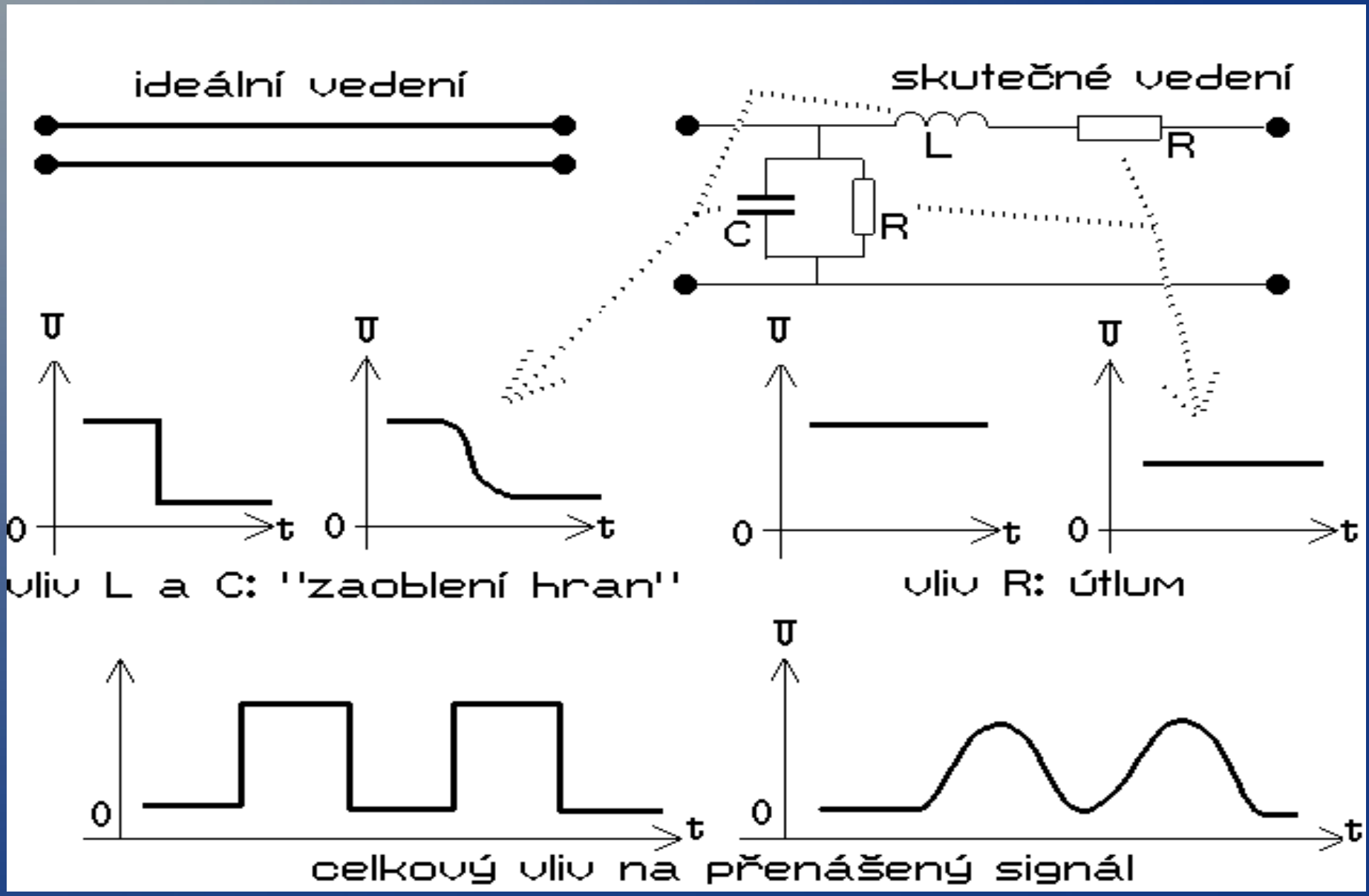
UPS 2015/2016

Cvičení 5

Obsah cviceni

- kapacita přenosového kanálu
- šířka pásma
- počet úrovní, bity, Baudy
- model kanálu se šumem
- modulace
- arytmický přenos, arytmická značka

Přenosový kanál



Kapacita kanálu

- W – šířka pásma [Hz]
  - Telefon 300-3400Hz = 3100Hz
- C – kapacita kanálu b/s
- V – počet úrovní signálu
- C = W log2 (1 + signál[w]/šum[w]) – Shannon
- S/N = 10 log(S/N) [dB]
- C = 2 W log2 (V) – Niquist
- Vp = Vm log2 (V)

Přenos

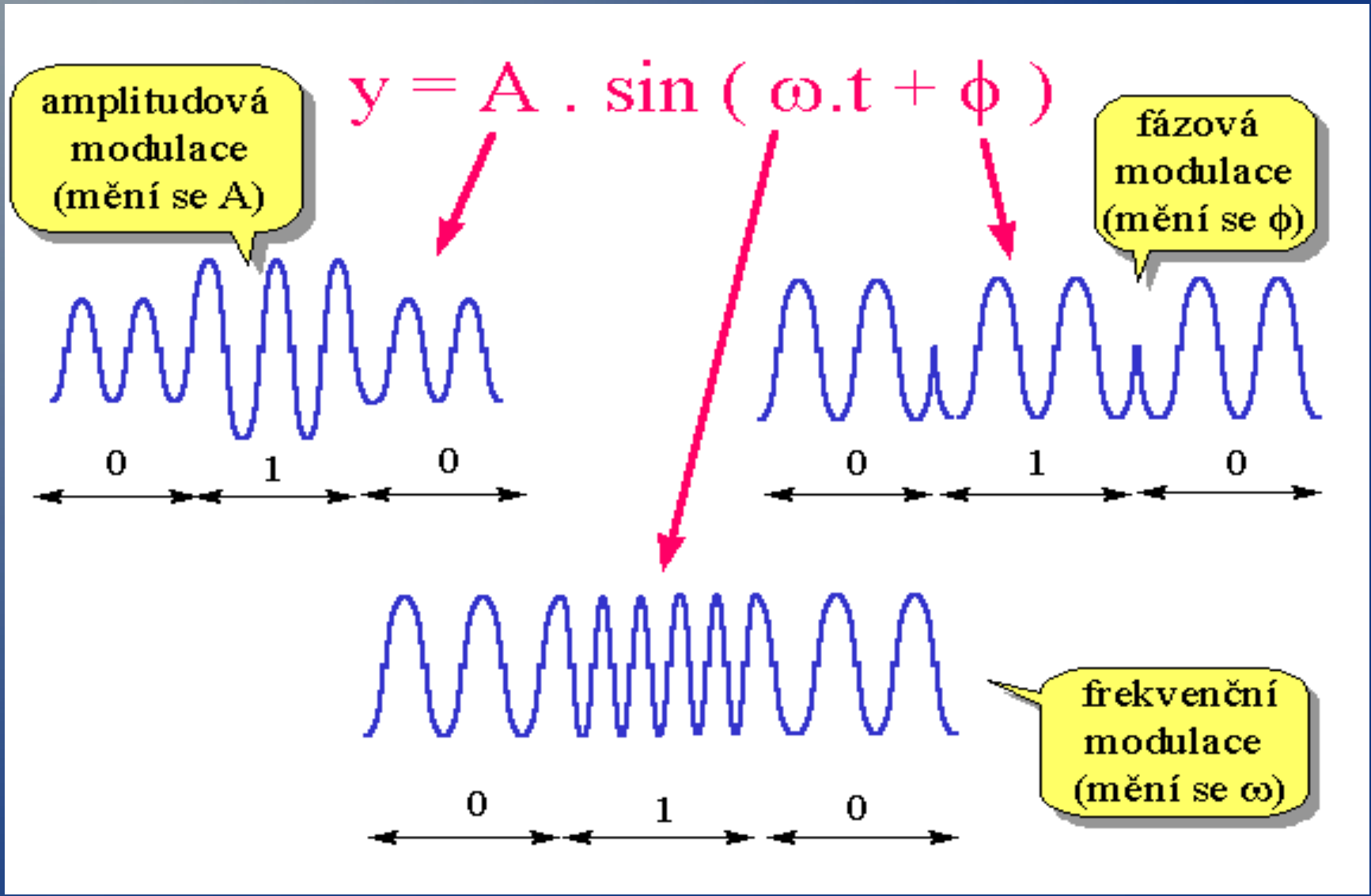
- Přenos v základním pásmu
  - 10BASE-T
  - Přenáší se pulzy (digitální technika)
  - Kratší vzdálenosti (menší vliv rušení, zkreslení)
- Přenos v přeloženém pásmu
  - Hlas, analogový modem
  - Signál je modulovaný (analogový přenos)
  - Delší vzdálenosti

Modulace

$y = A * \sin(\omega t + \phi)$

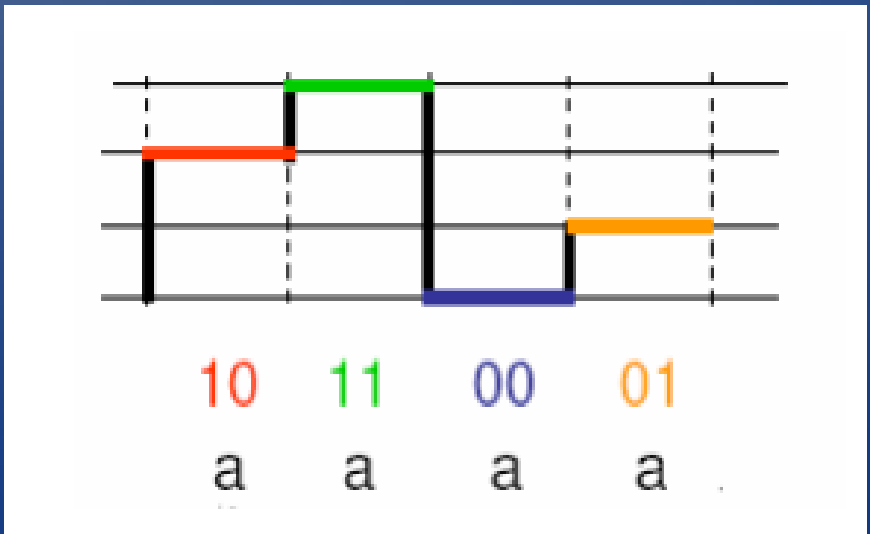
- Frekvenční
  - Mění se parametr ω
- Fázová
  - Mění se parametr ϕ
- Amplitudová
  - Mění se parametr A

Modulace


y = A . sin (ω.t + ϕ)
amplitudová modulace (mění se A)
fázová modulace (mění se ϕ)
frekvenční modulace (mění se ω)
0
1
0
0
1
0
0
1
0

Bit vs. Baud

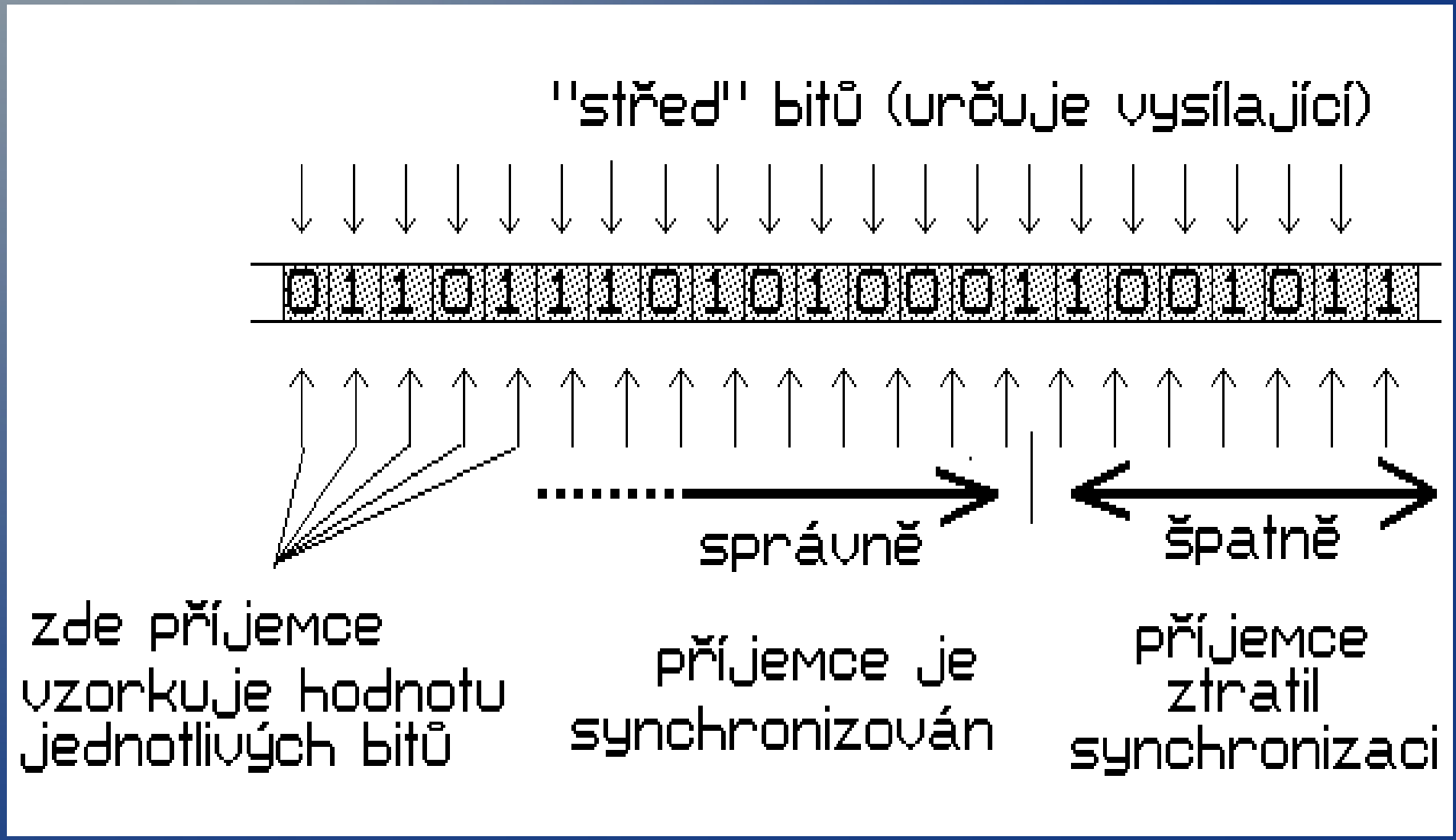
- Bit – jednotka informace (1 nebo 0)
- Baud – jednotka modulace (počet stavů/s)
  - Modulační rychlost (neboli rychlost, s jakou dochází k přechodům analogového signálu mezi stavy, reprezentujícími jednotlivé diskrétní hodnoty), může být maximálně rovna dvojnásobku šířky přenosového pásma.
- Obecně: Bit/sec nerovná se Baud



Přenosová rychlost

přenosová rychlost [bitů/s]	modulační rychlost [Bd]	počet rozlišovaných stavů	bitů/ změnu	standard
2400	600	16	4	V.22bis
9600	2400	16	4	V.32
14400	2400	64	6	V.32bis
28800, 33600	2400-3200	512	9	V.34
56000	8000	128	7	V.90,V.92

Přenos

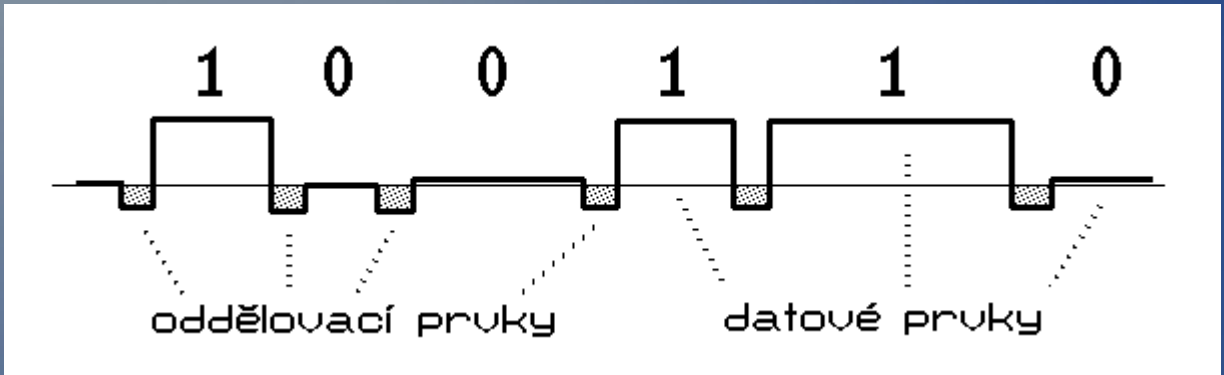

''střed'' bitů (určuje vysílající)
0 1 1 0 1 1 1 0 1 0 1 0 0 0 1 1 0 0 1 0 1 1
správně
špatně
zde příjemce vzorkuje hodnotu jednotlivých bitů
příjemce je synchronizován
příjemce ztratil synchronizaci

Typy přenosů

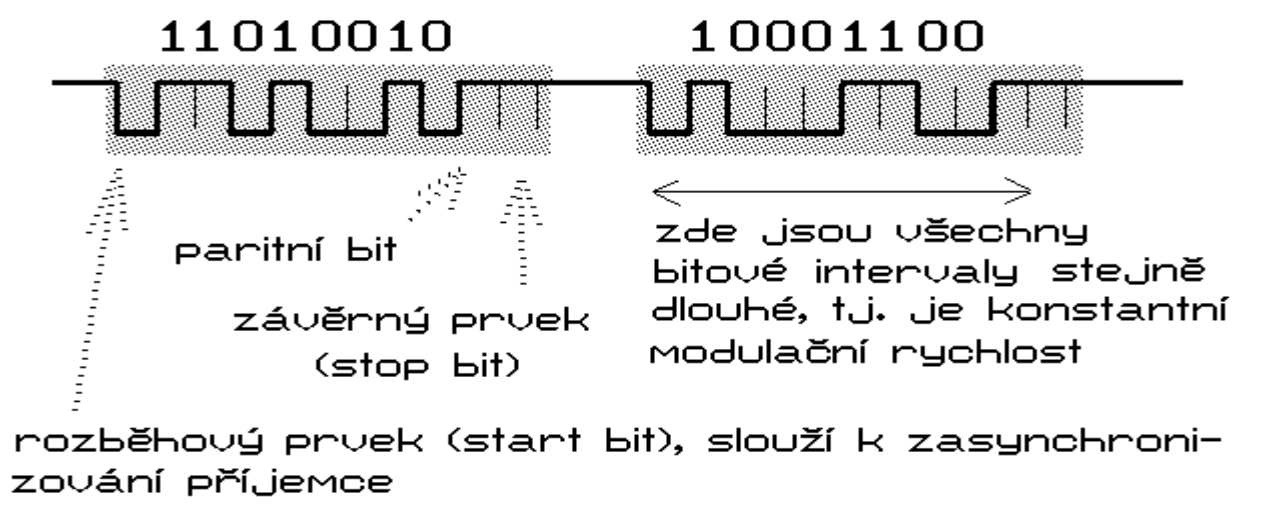
- Asynchronní – mezi příjemcem a vysílajícím neexistuje žádná synchronizace, speciální značky, přenos jednoho bitu může trvat, libovolně dlouhou dobu.
- Arytmický – mezi příjemcem a vysílajícím existuje synchronizace, na začátku a na konci přenosu bloku bitů, START/STOP bity, délka, přenosu znaku je pevná, délka přenosu bloku proměnlivá.
- Synchronní – mezi vysílajícím a přijímajícím existuje synchronizace, po celou dobu, hodiny jsou zakódovány do přenášených dat; NRZ, diferenciální manchester, …

Přenos II.

- Asynchronní oddělovací prvky

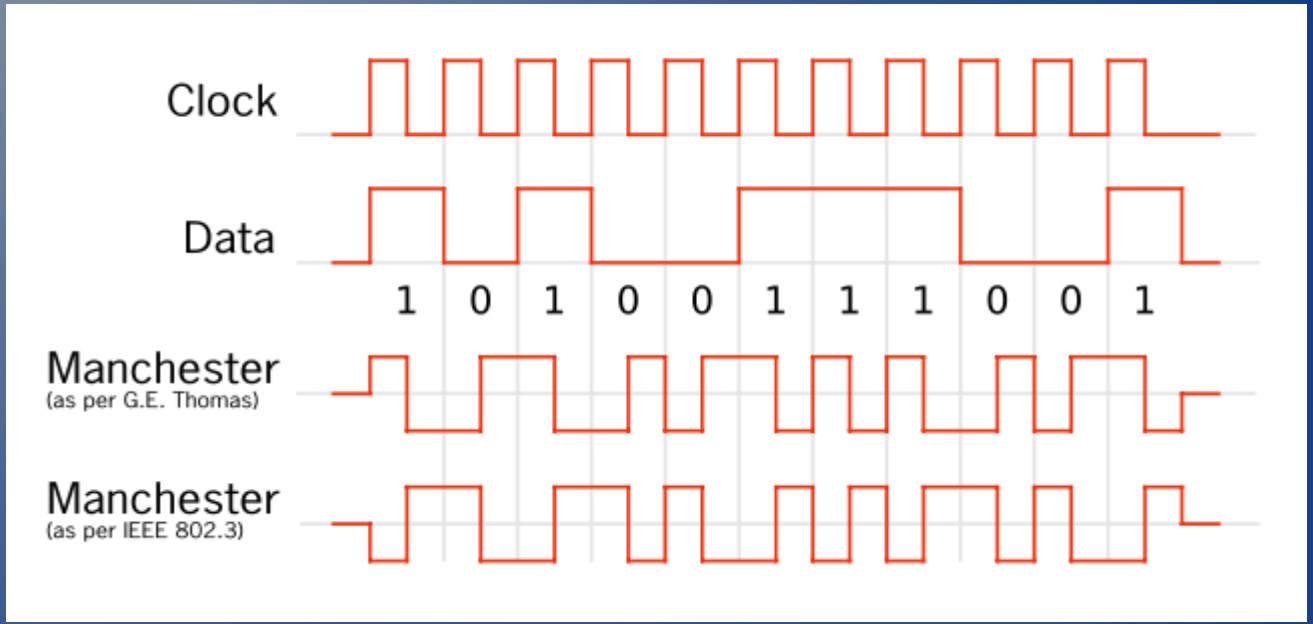


- Arytmický Start/stop bity označující hranice



Přenos III.

- Synchronní



UPS 2014/2015

Cvičení 6

Obsah

- Synchronní přenos, rámce, transparentnost přenosu, tvary rámců (s délkou, vkládání slabik, vkládání bitů), hranice rámců.
- Problém synchronizace (synchronní a asynchronní systémy).
- Kódování signálu, NRZ, NRZI, Manchester, RZ.
- Multiplexování, časový a frekvenční multiplex, synchronní a asynchronní multiplex.
- Sítě s přepínáním kanálů, zpráv a paketů.

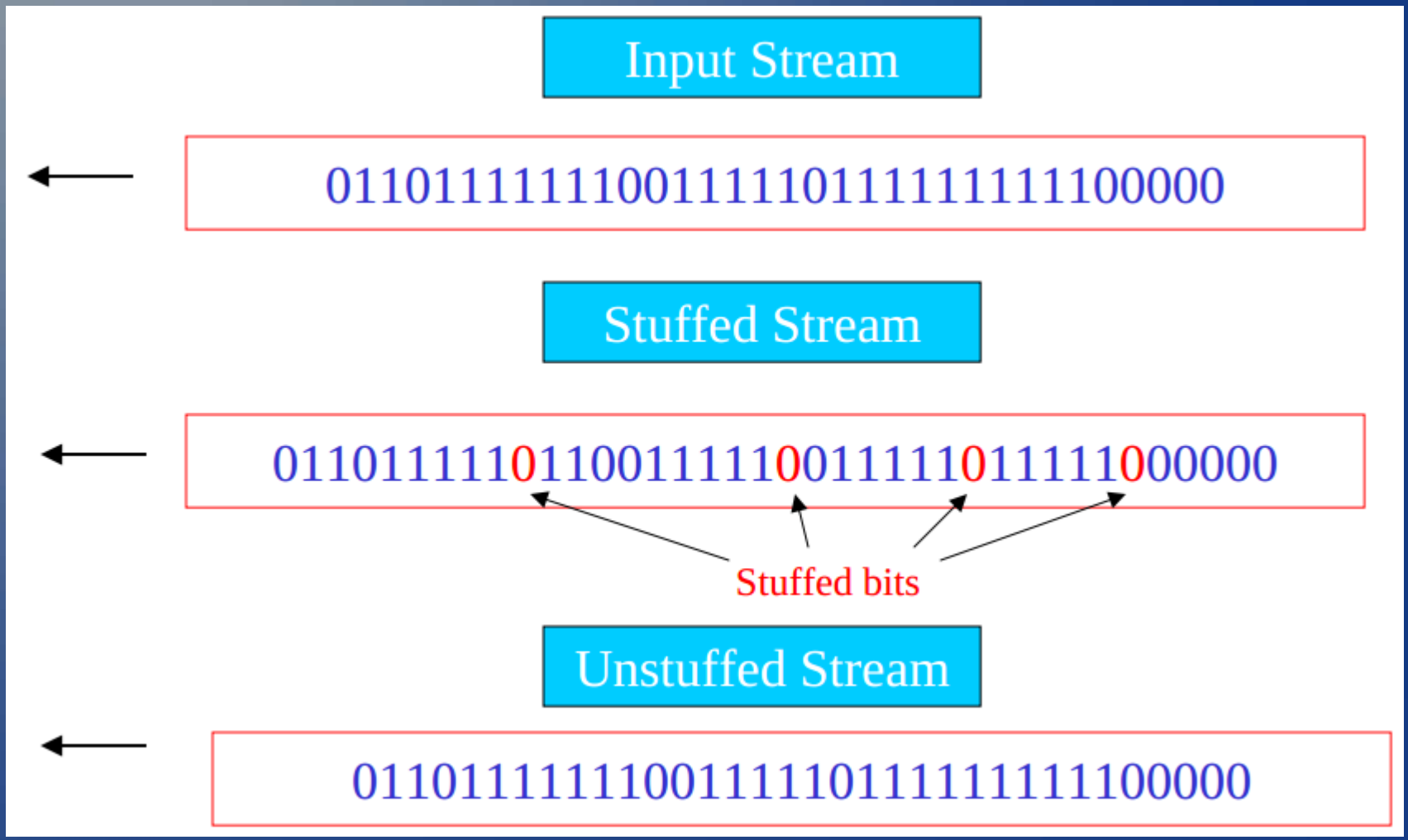
Rámce

- Datová jednotka linkové vrstvy
- 3 části, hlavička, data, patička
- Transparentnost přenosu
  - Vkládání bitů – po 5 jedničkách se vkládá nula
  - Vkládání speciálních znaků, např Escape sekvence
- Hranice rámce
    - STX - Start of TeXt
    - ETX - End of TeXt
    - DLE - Data Link Escape

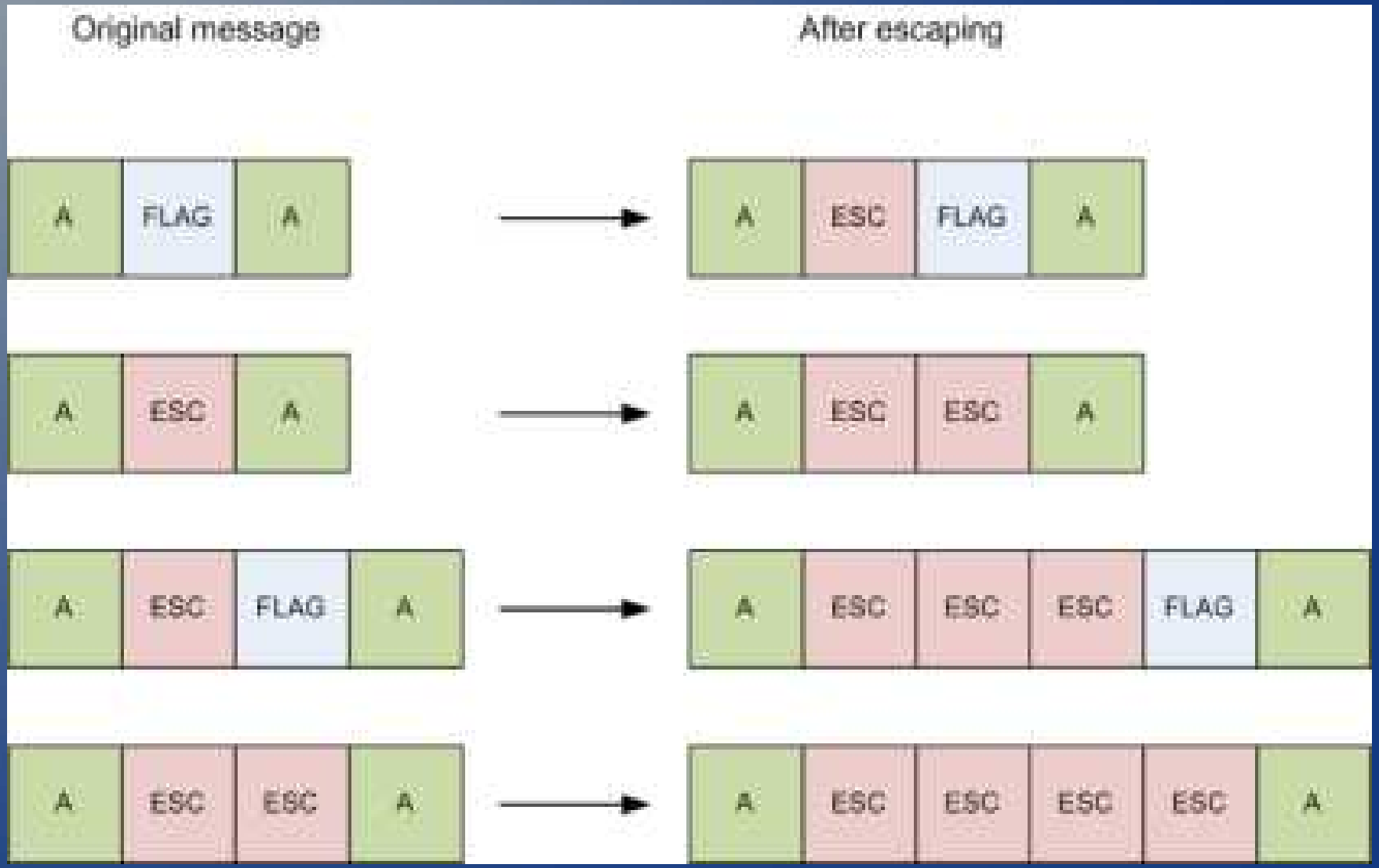
Synchronizace

- Bitová
  - Start/stop bit (hooodně režie)
- Bytová (znaková)
  - Start/stop bity
  - 8N1, 8E2
- Rámcová/délkově
  - Start/Stop znaky (STX,ETX)

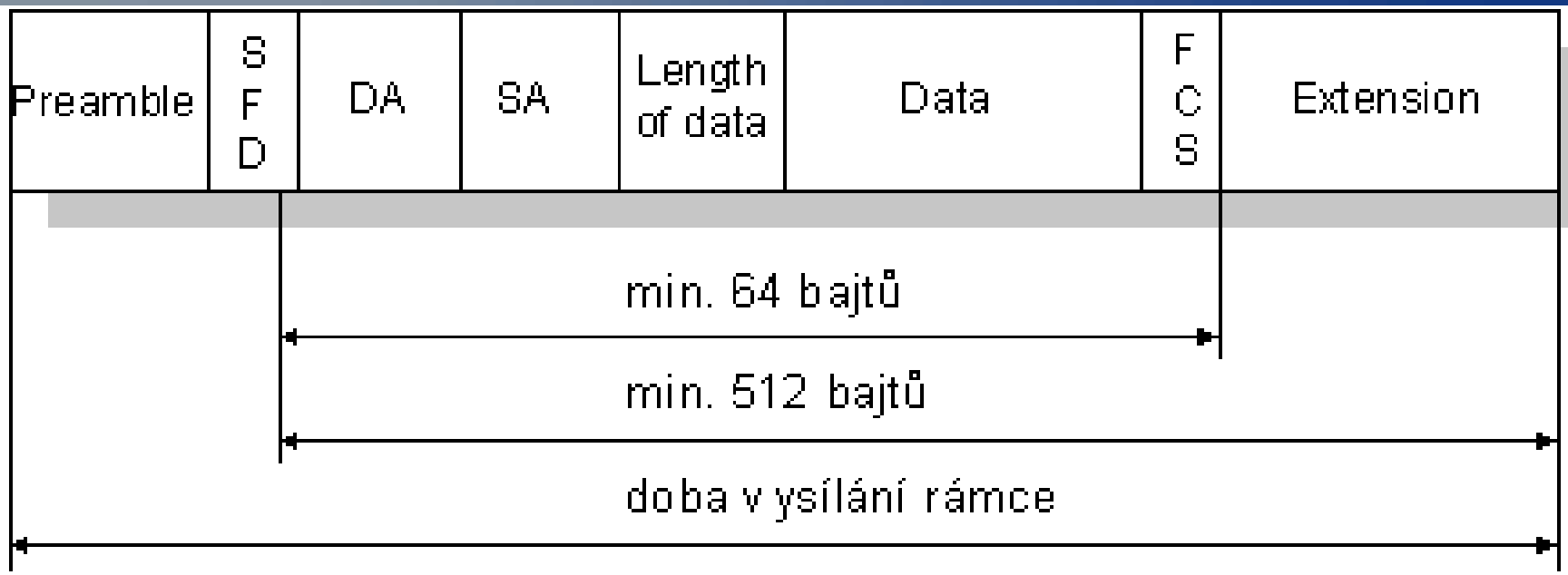
Vkládání bitů



Vkládání bajtů



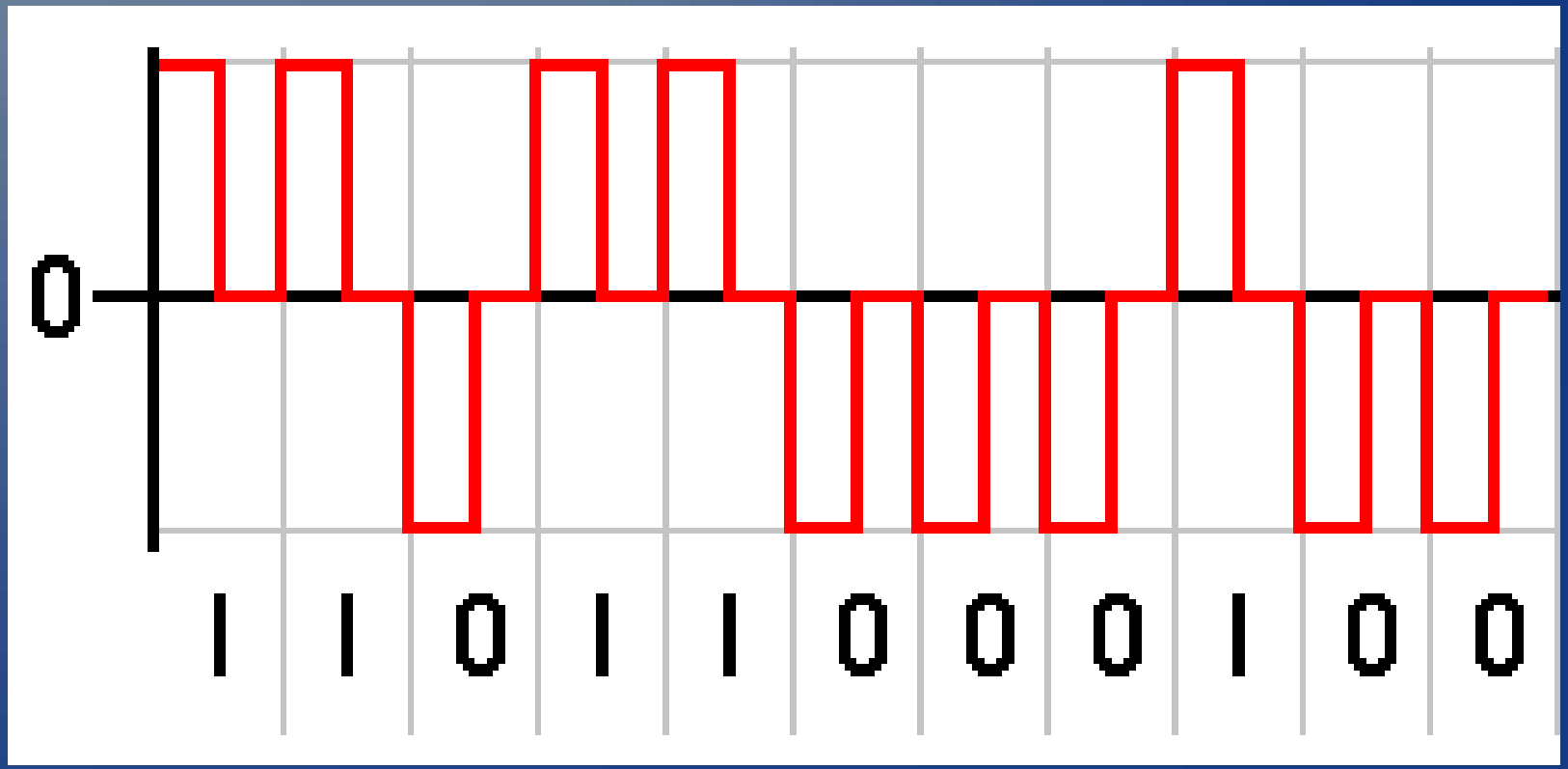
Rámce



SFD ... Start of Frame Delimiter
DA ..... Destination Address
SA ..... Source Address
FCS ... Frame Check Sequence

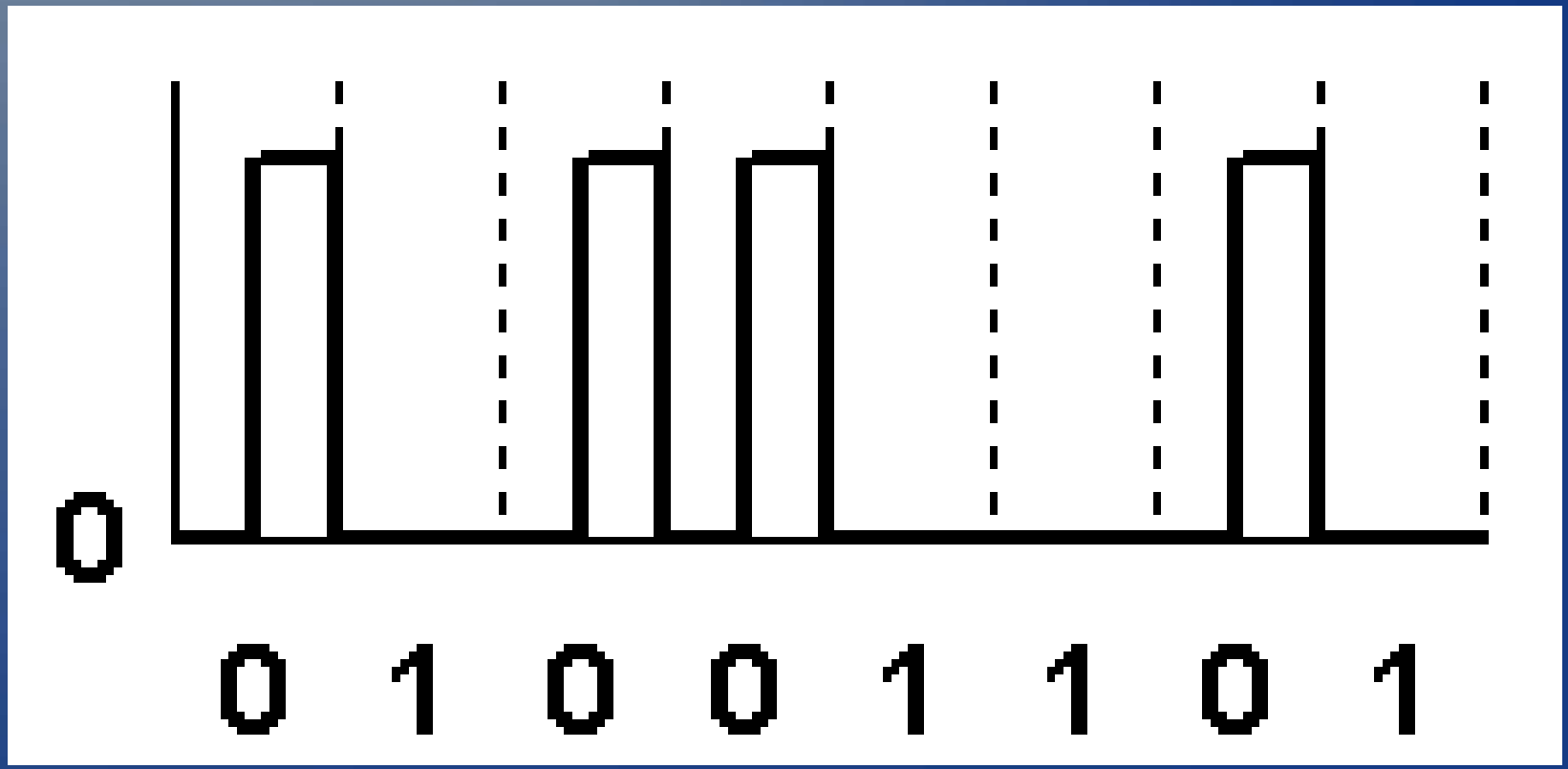
Kódování signálu

- RZ – Return To Zero
  - Kladné a záporné pulsy a vrací se k nule



RZI

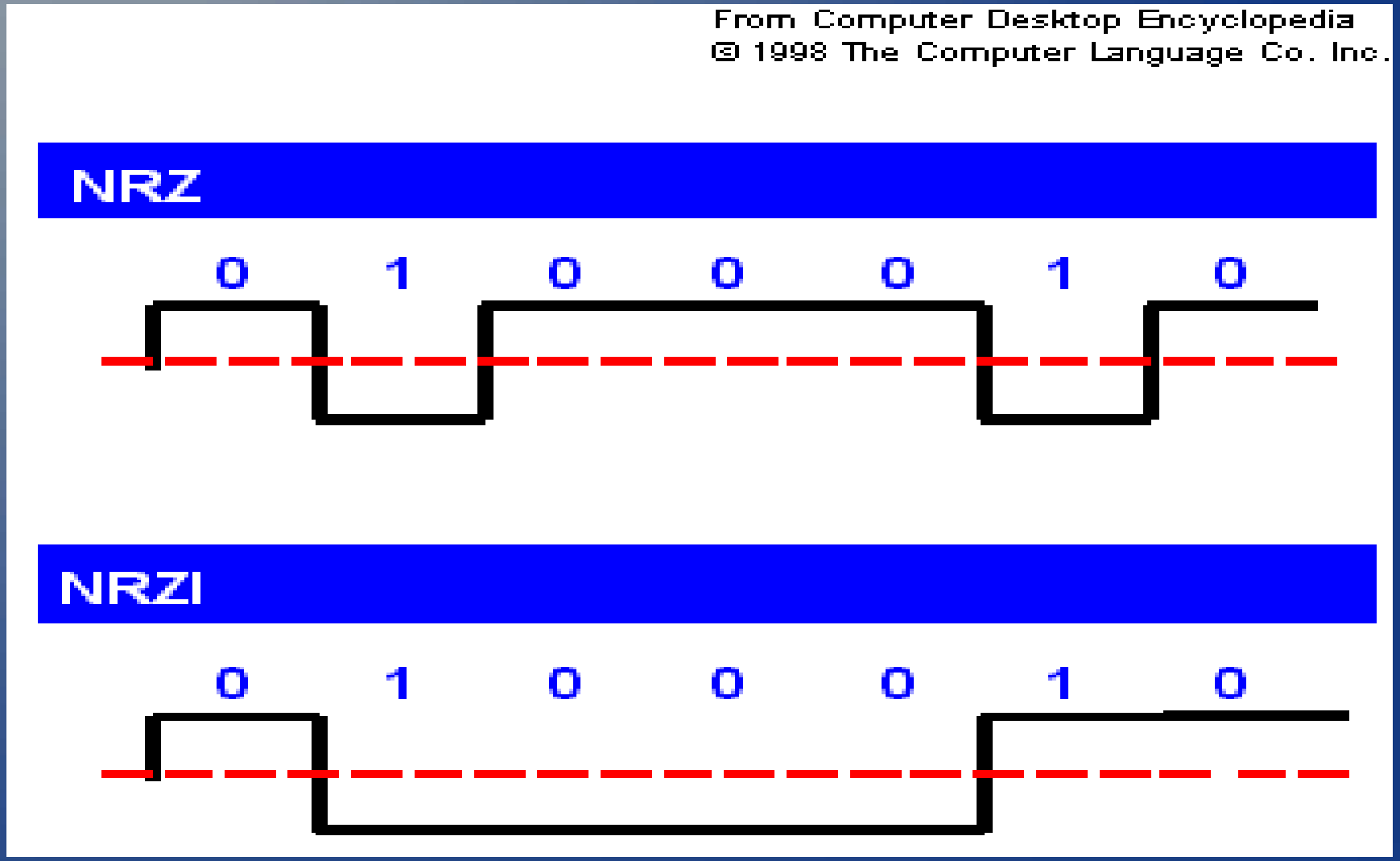
- RZI – Return To Zero Inverted
  - 0 – kratší signál než hodiny, 1 delší



NRZ, NRZI

- NRZ – Non Return To Zero
  - Pouze dvě úrovně nedochází k návratu k nule
- NRZI – Non Return To Zero Inverted
  - 1 – změna, 0 – pokud změna nenastala
  - Změna na vzestupné hraně hodinového signálu

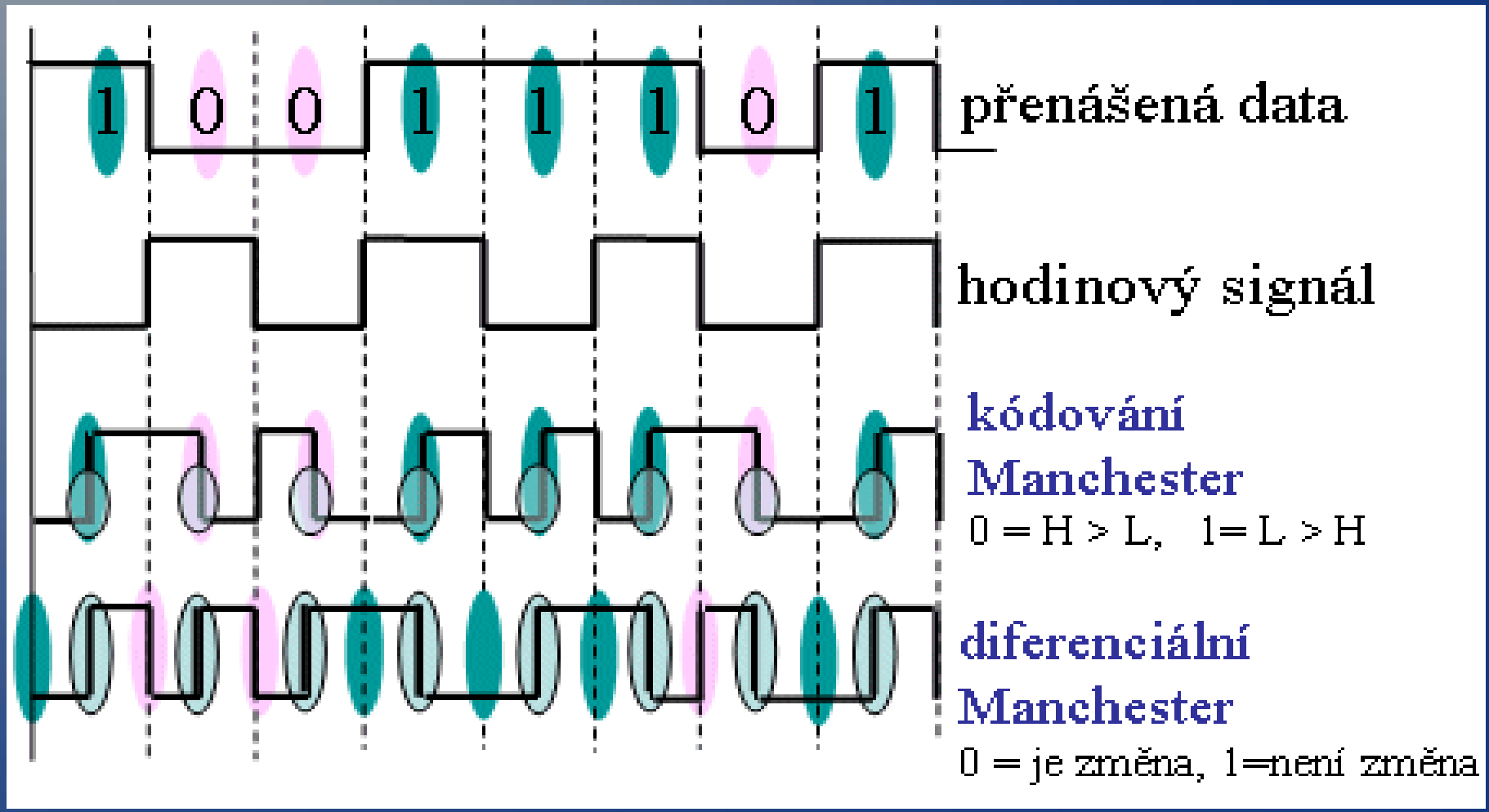
Kodování


From Computer Desktop Encyclopedia
© 1998 The Computer Language Co. Inc.
NRZ
0 1 0 0 0 1 0
NRZI
0 1 0 0 0 1 0

Kodování

- Manchester
  - 0/1 podle směru uprostřed pulzu
  - Hrana je vždy uprostřed, může dobře sloužit k synchronizaci
- Diferenciální Manchester
  - Hodiny jsou přímo součást dat
  - Signály se určují na základě přechodu
  - Lepší pro zašuměný kanál
  - Důležitý je přechod, ne směr, nevadí změna polarity

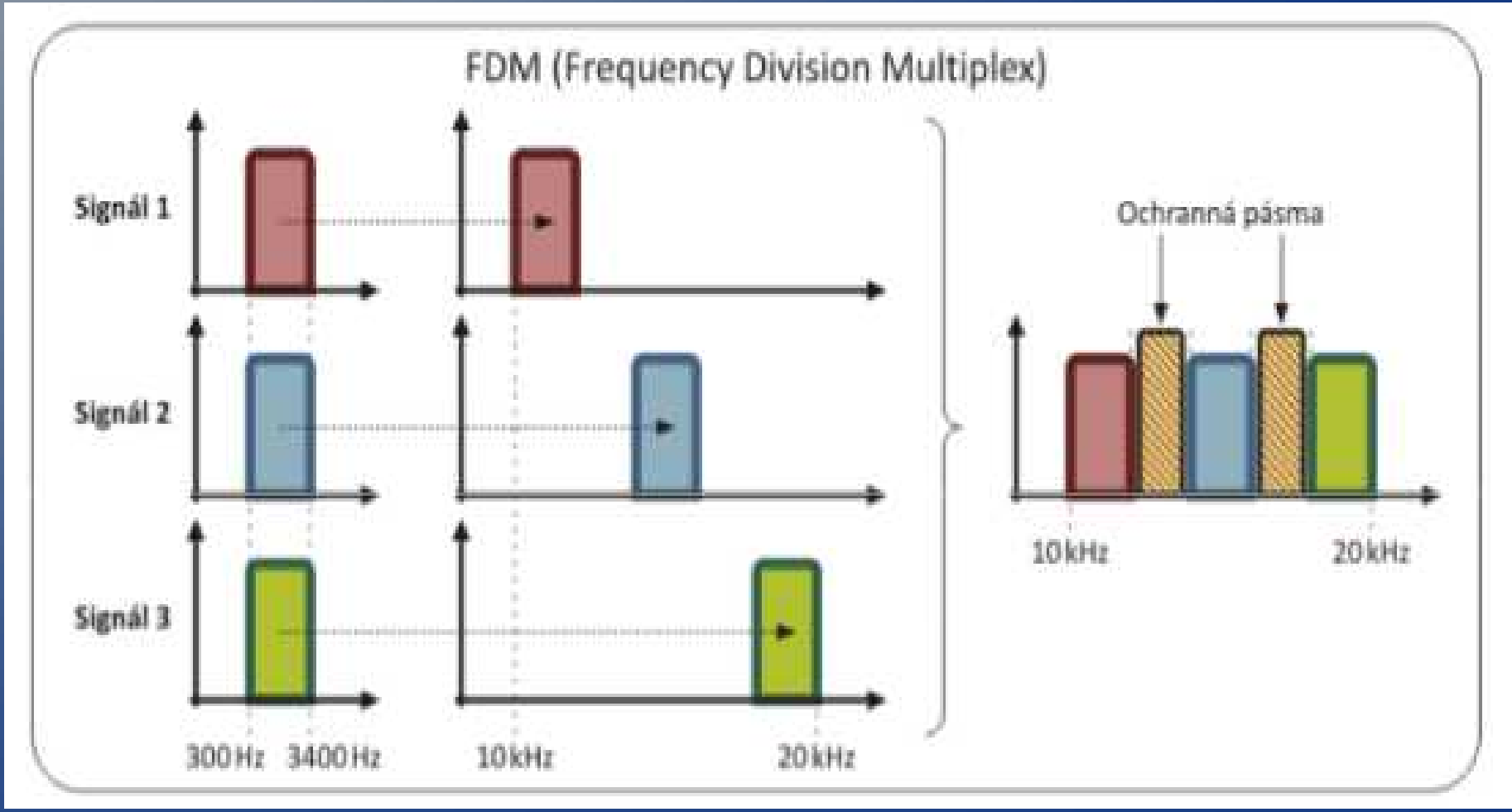
Kodování


1 0 0 1 1 1 0 1
přenášená data
hodinový signál
kódování
Manchester
0 = H > L, 1 = L > H
diferenciální
Manchester
0 = je změna, 1=není změna

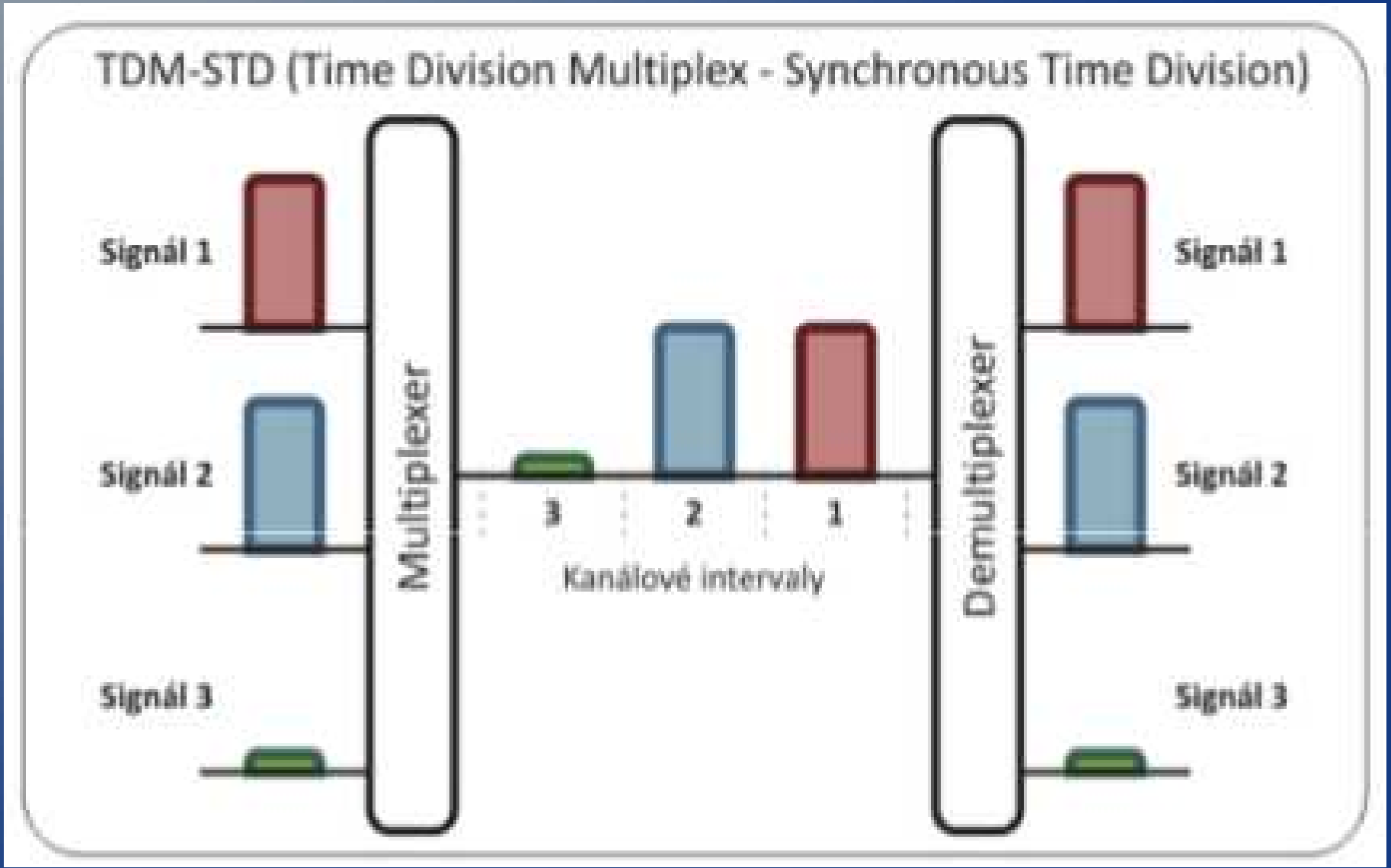
Multiplex

- Frekvenční – FDMA (analog)
  - Více vysílání na různých frekvencích
- Časový – TDMA (ISDN, GSM)
  - Časové sloty/rámce
- Vlnový – WDMA (DWDM, optické sítě)
  - Defacto frekvenční, do optického vlákna se dává více zdrojů světla o různých vlnových délkách
  - Tvoří samostatné kanály
- Kodový - CDMA (CDMA)
  - Zakódovaná data pro všechny a každý si vezme jen co je jeho

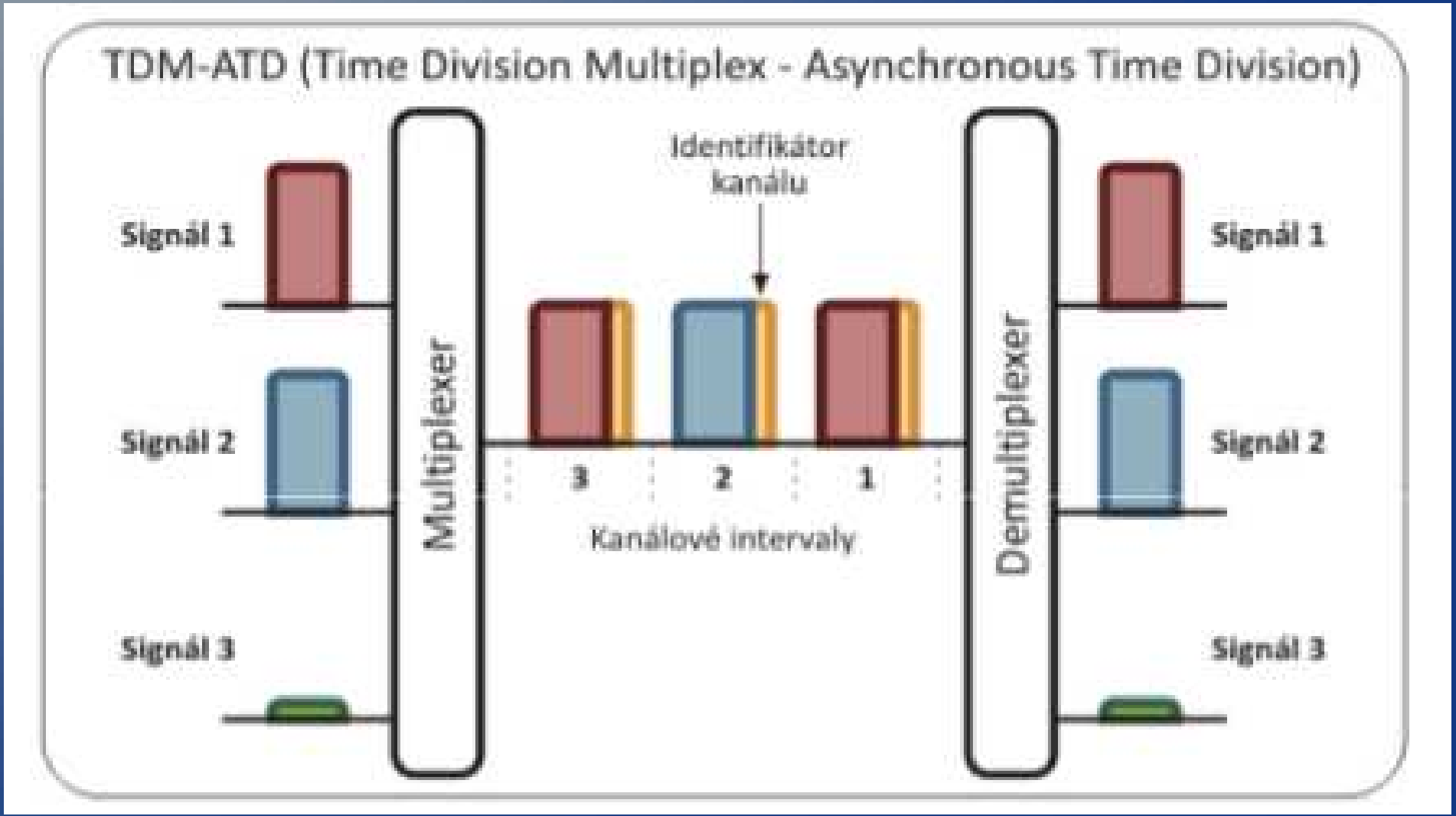
FDM


FDM (Frequency Division Multiplex)
Signál 1
Signál 2
Signál 3
300 Hz
3400 Hz
10 kHz
20 kHz
Ochranná pásma
10 kHz
20 kHz

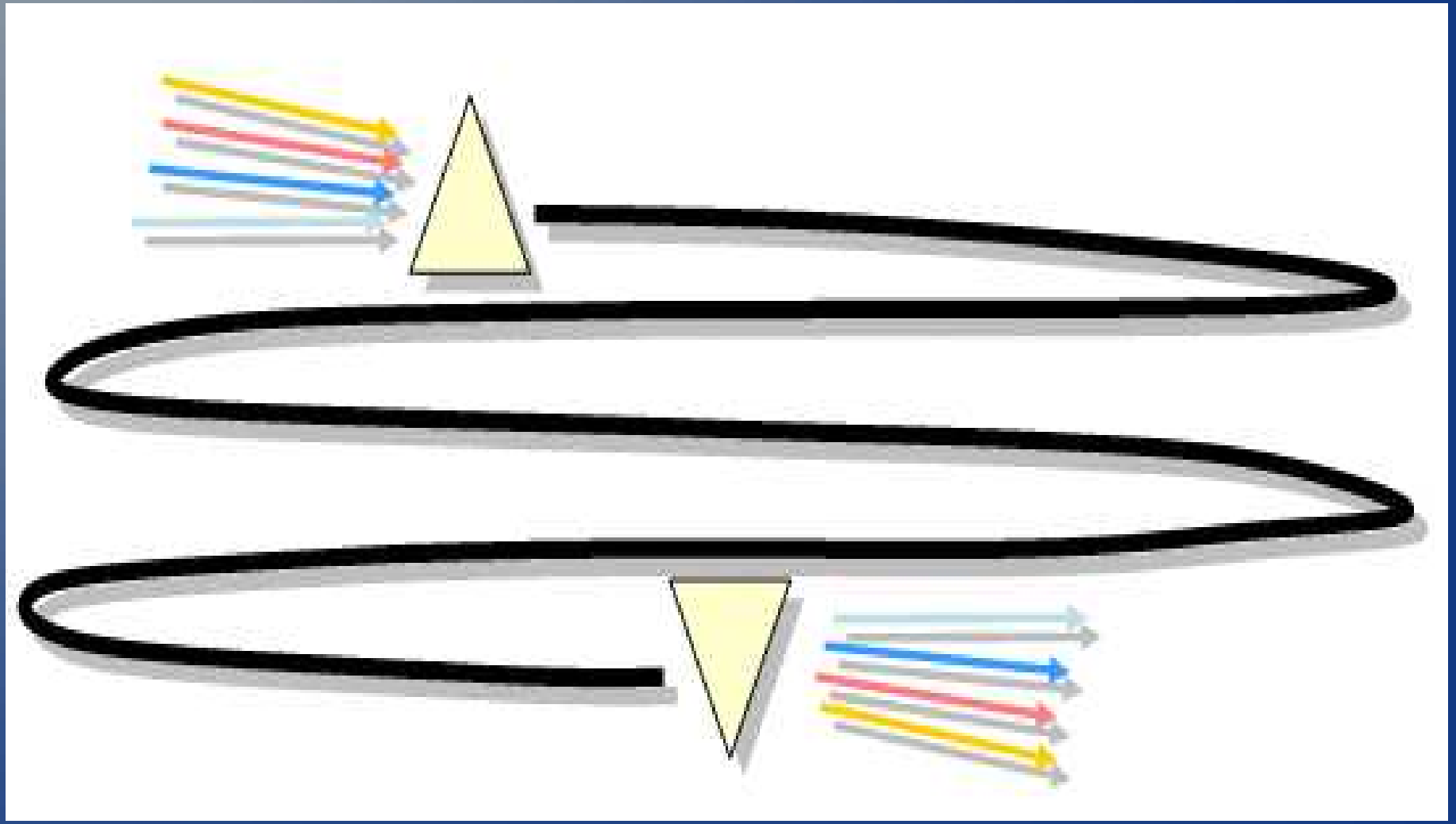
TDM I.


TDM-STD (Time Division Multiplex - Synchronous Time Division)
Signál 1
Signál 2
Signál 3
Multiplexer
3
2
1
Kanálové intervaly
Demultiplexer
Signál 1
Signál 2
Signál 3

TDM II.


TDM-ATD (Time Division Multiplex - Asynchronous Time Division)
Identifikátor kanálu
Signál 1
Signál 2
Signál 3
Multiplexer
3
2
1
Kanálové intervaly
Demultiplexer
Signál 1
Signál 2
Signál 3

WDM



Sítě s přepínáním

- Kanálů (telefon, ATM, FrameRelay)
  - virtuální kanál kterým tečou veškerá data
  - Daným kanálem tečou veškerá data
  - Tvoří se před navázáním spojení
- Packetů (TCP/IP)
  - Žádná pevná cesta
  - O každém packetu se rozhoduje zvlášť na
  - Linkové vrstvě – přepínání rámců
  - Síťové vrstvě – přepínání packetů
- Zpráv (analogie email)
  - Speciální případ předchozího
  - Přepnutí mezi dvěma body naráz
  - Store-and-forward

UPS

Cvičení 7

http://siroky.cz/vyuka/ups/

Opakování

- ISO/OSI
- TCP/UDP
- základní/přeložené pásmo
- modulace
- $\log_2(L)$
- bit/baud
- asynchronní, arytmický, synchronní
- Manchaster

Dobrovolné odevzdání

- příští cvičení
- protokol
- prototyp serveru
  - rychle předvést pár dotazů a odpovědí (např. nc jako klient)

Chyba přenosu

- dojde ke ztrátě či záměně dat
  - zkreslení signálu, rušení, šum
- bezpečnostní kódy
  - detekce chyb x oprava chyb
- Uvažuje binární symetrický přenosový kanál bez paměti:
  - binární: přenáší se 0/1
  - symetrický: 0/1 se přenáší se stejnou pravděpodobností
  - bez paměti: nezáleží co se přeneslo v předchozím kroku

Chyba při přenosu

- pravděpodobnost správného přenosu 1 bitu
  $P_1 = p_1$
  - Kontrolní otázka: jaká je nejmenší použitelná pravděpodobnost?

- pravděpodobnost správného přenosu N bitů
  $P_N = p_1^N$

Příklad

- P_1=0.9999
- P_n=0.9
- N=?

Příklad

- $P_1 = 0.9999$
- $P_n = 0.9$
- $N = ?$
- $P_N = P_1^N$: $0.9 = 0.9999^N$

Příklad

- $P_1 = 0.9999$
- $P_n = 0.9$
- $N = ?$
- $P_N = P_1^N$: $0.9 = 0.9999^N$
- $\log(x^y) = y \log(x)$: $\log(0.9) = N \log(0.9999)$

Příklad

- $P_1 = 0.9999$
- $P_n = 0.9$
- $N = ?$
- $P_N = P_1^N$: $0.9 = 0.9999^N$
- $\log(0.9) = N \log(0.9999)$
- $N = \log(0.9) / \log(0.9999)$

Příklad

- $P_1 = 0.9999$
- $P_n = 0.9$
- $N = ?$
- $P_N = P_1^N$: $0.9 = 0.9999^N$
- $\log(0.9) = N \log(0.9999)$
- $N = \log(0.9) / \log(0.9999)$
- $N = 1053$

Bezpečnostní kódy

- přidáme nějaké bity navíc nebo pozměníme data
- čím více bitů navíc tím účinnější metoda
- detekční – kontrola zda jsou data správně
- samoopravné – chybu rozpoznají a opraví

Parita

- přidáváme jeden paritní bit
- sudá 0 = sudý počet 1, 1 = lichý počet 1
  - vždy sudý počet 1 ve zprávě
  - umí jen detekovat, nevíme co je špatně
- lichá parita je analogie k sudé

Parita

- př.: doplňte lichý paritní bit do zpráv
  - 01001101
  - 111101

Parita

- př.: odesílatel: 10101101
  - Jaká je parita?
  - Které přijaté zprávy jsou "správné"?
    - 10101001
    - 10101000
    - 11111110

Checksum

- kontrolní součet – pro celý blok dat
- jednotlivé znaky chápeme jako čísla bez znaménka
- provádíme sčítání modulo 2^8 nebo 2^{16}
  - Kontrolní otázka: proč 2?
  - Kontrolní otázka: proč 8 nebo 16?
- výsledek je číslo o délce 1 nebo 2 bytů
- výpočet probíhá postupně

Checksum

- př.: spočítejte checksum modulo 2^8 zprávy

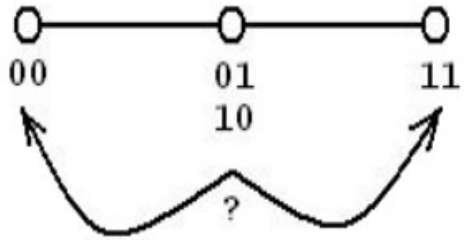
  0x3a 0x10 0x00 0xab 0x9f

Hammingova vzdálenost

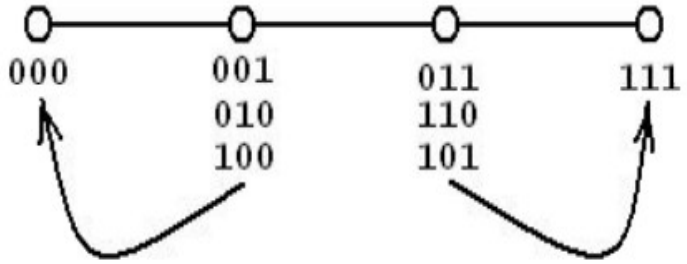
- počet míst v němž se dvě kódová slova liší
  - např.: 000 a 001 mají vzdálenost 1, 010 a 101 mají vzdálenost 3
- minimální Hammingova vzdálenost d_{min} = minimální vzdálenost mezi všemi možnými páry vektorů
  - př. {0000, 1011, 1111}, d_{min}=?
  - př. {00000, 10110, 11100}, d_{min}=?

Hammingova vzdálenost

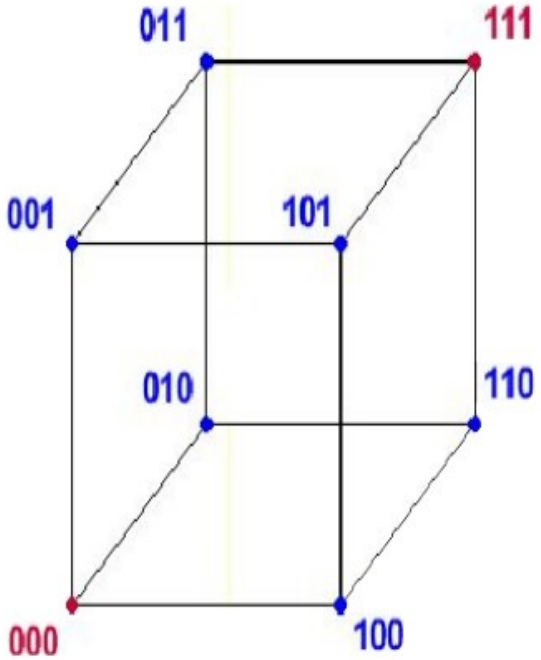
- 0 a 1 budu kódovat více bity, např. 00 nebo 111



Minimální Hammingova vzdálenost kódu je 2.
Jednobitová chyba jde detekovat, ale nelze opravit.



Minimální Hammingova vzdálenost kódu je 3.
Jedno a dvoubitová chyba jdou detekovat.
Opravit lze pouze jednobitovou chybu.



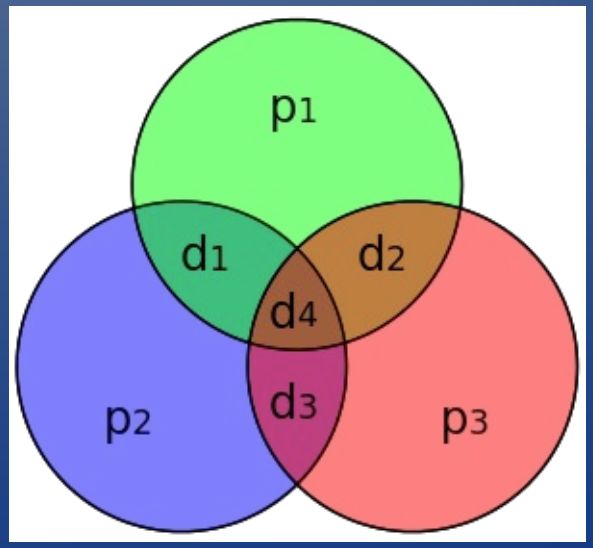
Hammingova vzdálenost

- Pro detekci n bitových chyb platí
  - $d_{min} > n$
- Pro detekci a korekci n bitových chyb platí
  - $d_{min} > 2n$

- Př.: 0:0000, 1:1111 - co dokážu říct o přijaté zprávě 0101?
- Př.: 0:00000, 1:11111 - co dokážu říct o přijaté zprávě 01010?

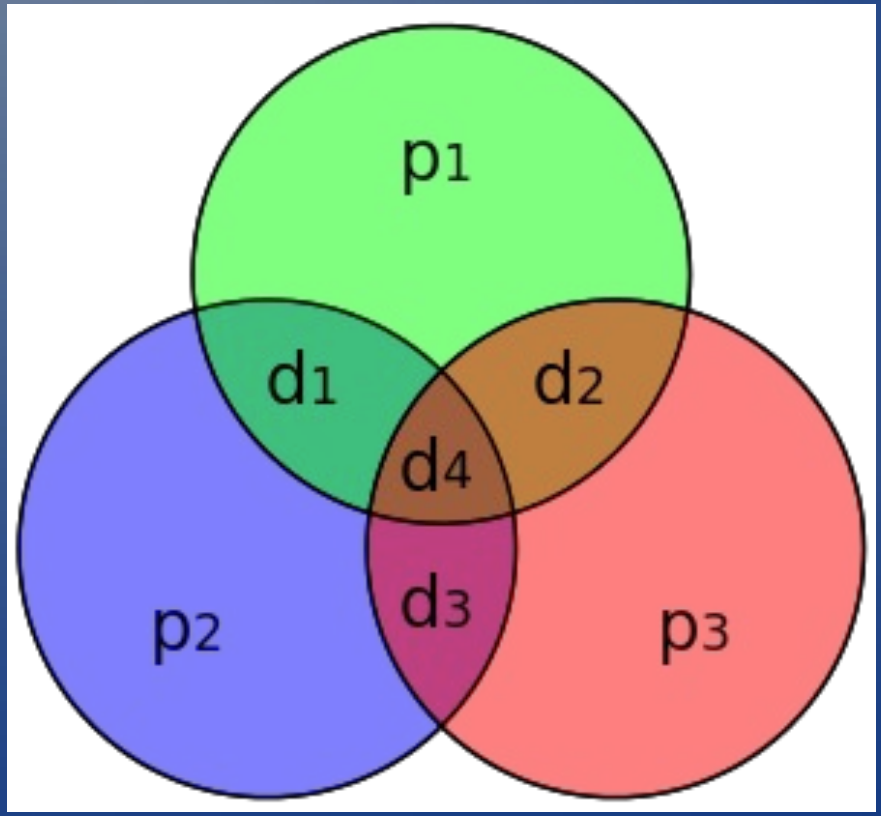
Hammingův kód (7,4)

- dovoluje detekovat dvojitou a opravit jednoduchou chybu
  - Kontrolní otázka: jaká je d_{min}?
- 4 datové bity, 3 sudé paritní
  - p1, p2, d1, p3, d2, d3, d4



Hammingův kód (7,4)

- p1, p2, d1, p3, d2, d3, d4
- př.: zakódujte zprávy:
  - 0000, 1111, 1011, 0010



Hammingův kód (7,4)

$$\mathbf{G} := \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{H} := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

- t = Gm
- c = Ht
- př.: zakódujte a ověřte zprávu:
  - 0110

CRC

- cyklický redundantní součet
- jednotlivé datové bity tvoří koeficienty polynomu
  - např. ...1001... -> ...$1x^{14} + 0x^{13} + 0x^{12} + 1x^{11}$... = ...$x^{14} + x^{11}$...
- tento se vydělí tzv. charakteristickým polynomem
  - např. CRC16: $x^{16} + x^{15} + x^2 + 1$
    - Kontrolní otázka: kolika bitový je?
- data = podíl * charpolynom + **zbytek**
- **zbytek** se použije pro zabezpečení

CRC

- př.: zapište polynom pro 1011
- př.: zapište bitovou posloupnost polynomu $x^7 + x^3 + x^2$

CRC

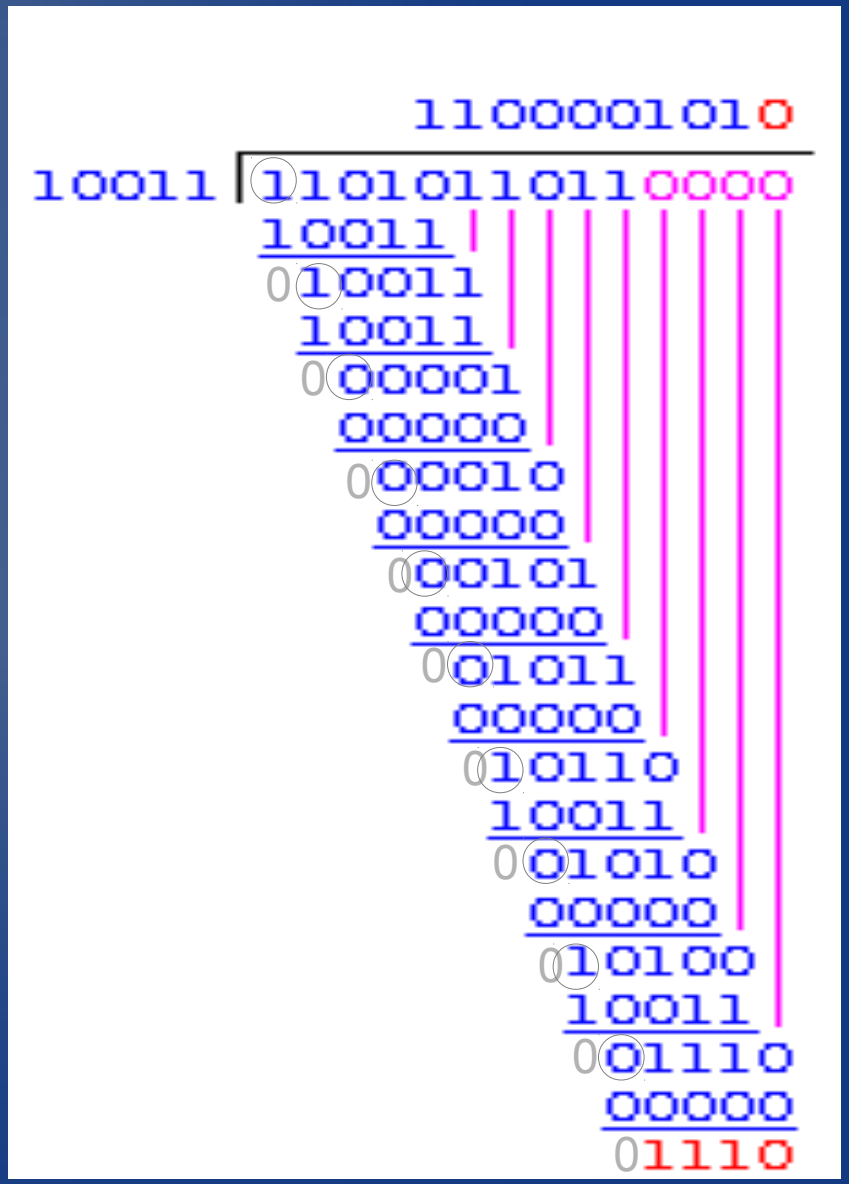
- vypočteme zbytek po dělení R(x) = M(x) % G(x)
- odesíláme T(x) = M(x) | R(x)
- po přijetí provedeme T(x) % G(x)
- pokud je zbytek nula, je přenos v pořádku
- označení jako CRC 16, 32 atp. podle stupně polynomu G(x)

CRC příklad

- M(x)=1101011011
- G(x) = 10011 ($x^4 + x + 1$)
  - polynom stupně 4
- za zprávu přidám 4 nuly (M(x) * x^4) a dělím
  - 1101011011 0000 / 10011
- R(x) = 1110

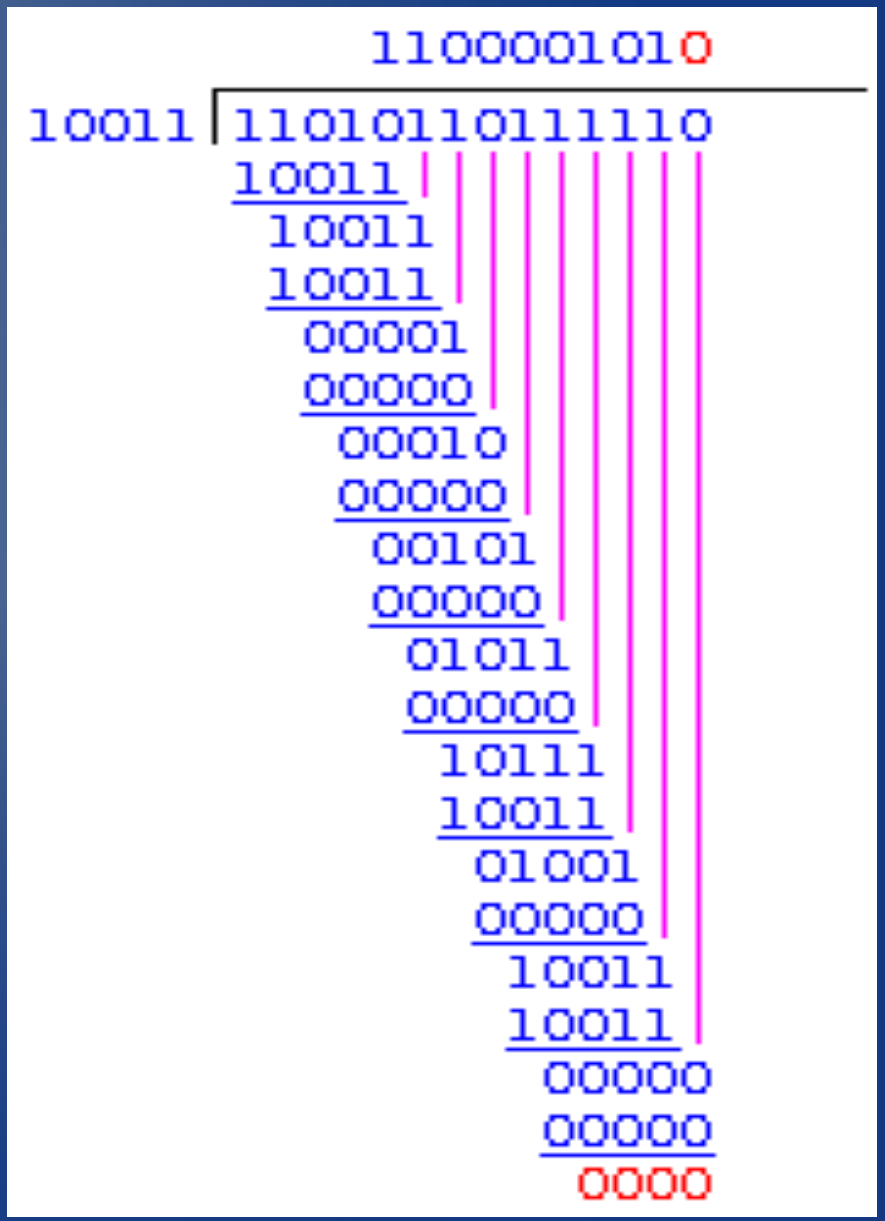
CRC příklad

- postup dělení
- stejné jako dělení pod sebe
- operaci odečítání nahrazuje operace XOR
- odesíláme M(x) | R(x)
  - 1101011011 | 1110



CRC příklad

- Ověření přijaté zprávy

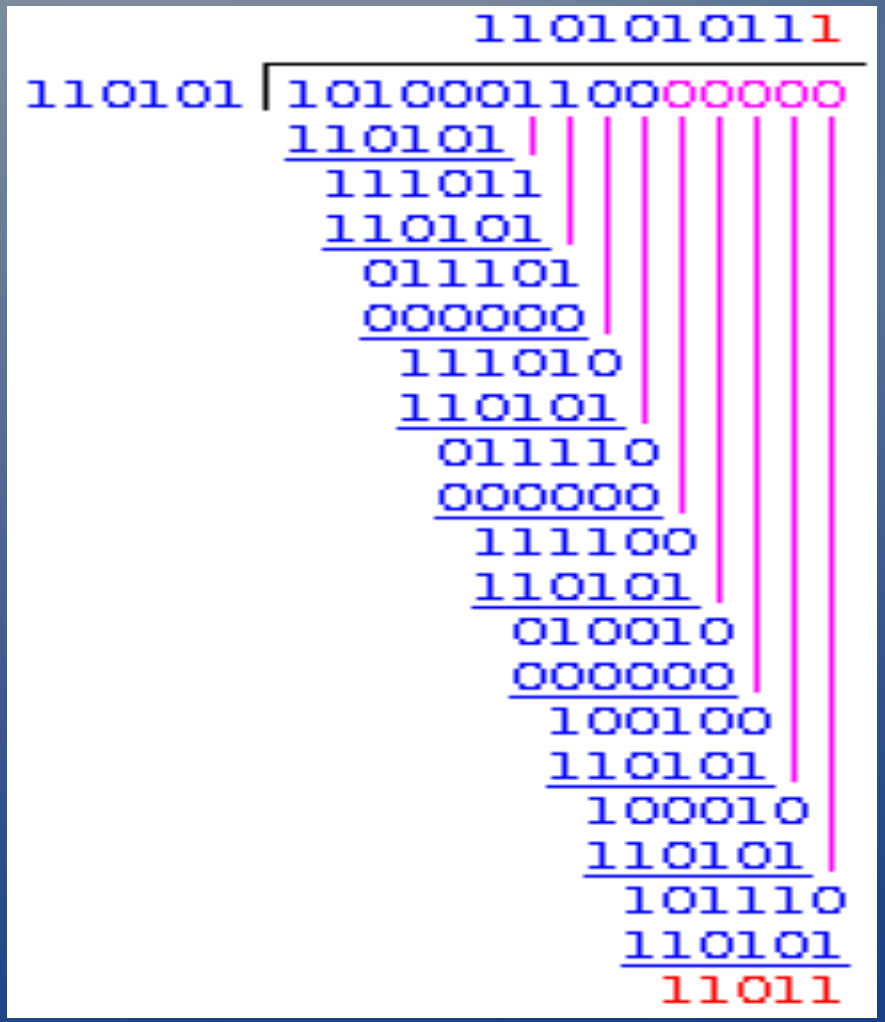


CRC samostatně

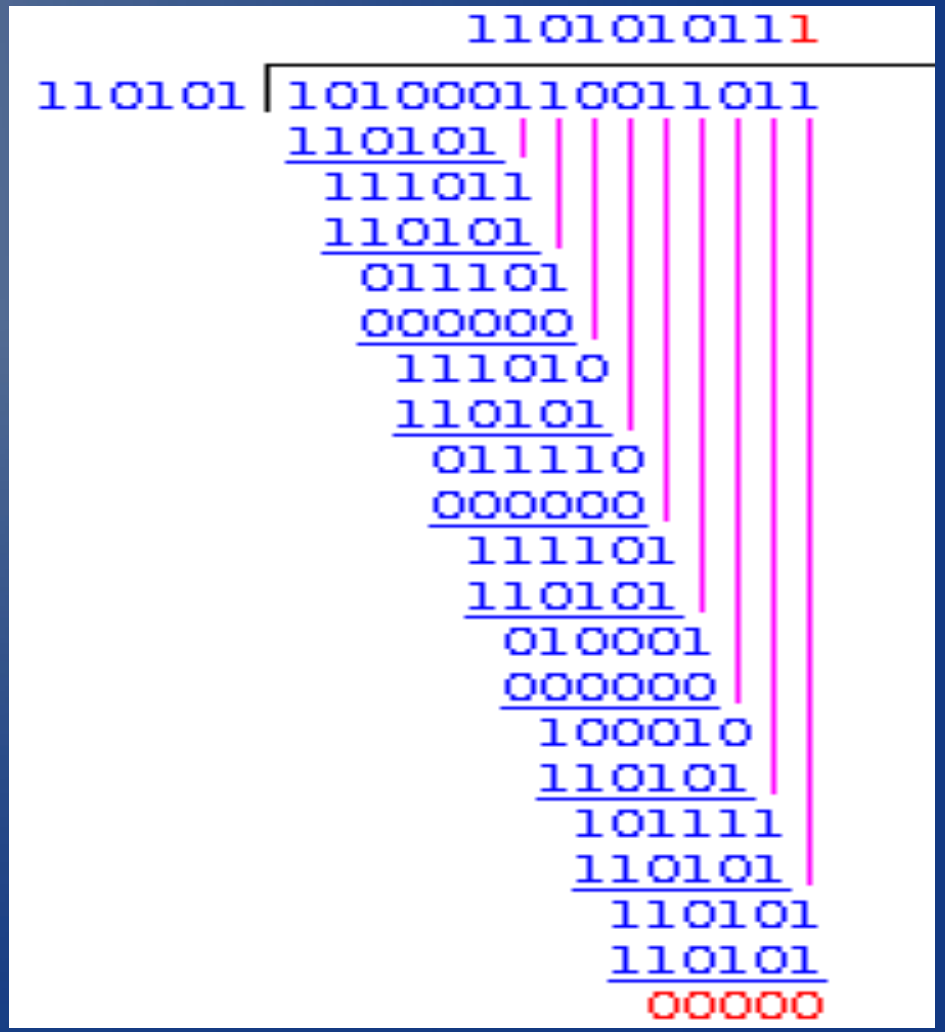
- M(x) = 1010001100
- G(x) = $x^5 + x^4 + x^2 + 1$
- R(x) =
- T(x) =

CRC samostatně

Zabezpečení



Kontrola



UPS 2014/2015

Cvičení 7

Obsah

- Domácí úkol a ukázka klienta
- Chyby
- Hammingova vzdálenost
- Parita
- CRC

Chyba přenosu

- Dojde ke ztrátě či záměně dat
  - Zkreslení signálu, rušení, šum
- Bezpečnostní kódy
  - Detekce chyb x oprava chyb
- Uvažuje symetrický binární přenosový kanál bez paměti
  - Symetrický: 0/1 se přenáší se stejnou pravděpodobností
  - Binární: Přenáší se 0/1
  - Bez paměti: Nezáleží co se přeneslo v předchozím kroku

Chyba při přenosu

- Pravděpodobnost přenosu 1 bitu $P_1 = p_1$
- Pravděpodobnost přenosu N bitů $P_N = p_1^N$
- Příklad:
  - máme SBPKBP, kolik bitů můžeme přenést, aby pravděpodobnost bezchybného přenosu byla 0,9, když pravděpodobnost přenosu 1 bitu je 0,9999 ?

Příklad I

- $P_1 = 0.9999$
- $P_n = 0.9$
- N=?

Příklad II

- $P_1 = 0.9999$
- $P_n = 0.9$
- N=?
- $0.9 = 0.9999^N$

Příklad III

- $P_1 = 0.9999$
- $P_n = 0.9$
- $N = ?$
- $0.9 = 0.9999^N$
- $\ln(0.9) = N \ln(0.9999)$

Příklad IV

- $P_1 = 0.9999$
- $P_n = 0.9$
- $N = ?$
- $0.9 = 0.9999^N$
- $\ln(0.9) = N \ln(0.9999)$
- $N = \ln(0.9) / \ln(0.9999)$

Příklad V

- $P_1 = 0.9999$
- $P_n = 0.9$
- N=?
- $0.9 = 0.9999^N$
- log(0.9) = N log(0.9999)
- N = log(0.9) / log(0.9999)
- N = 1 053

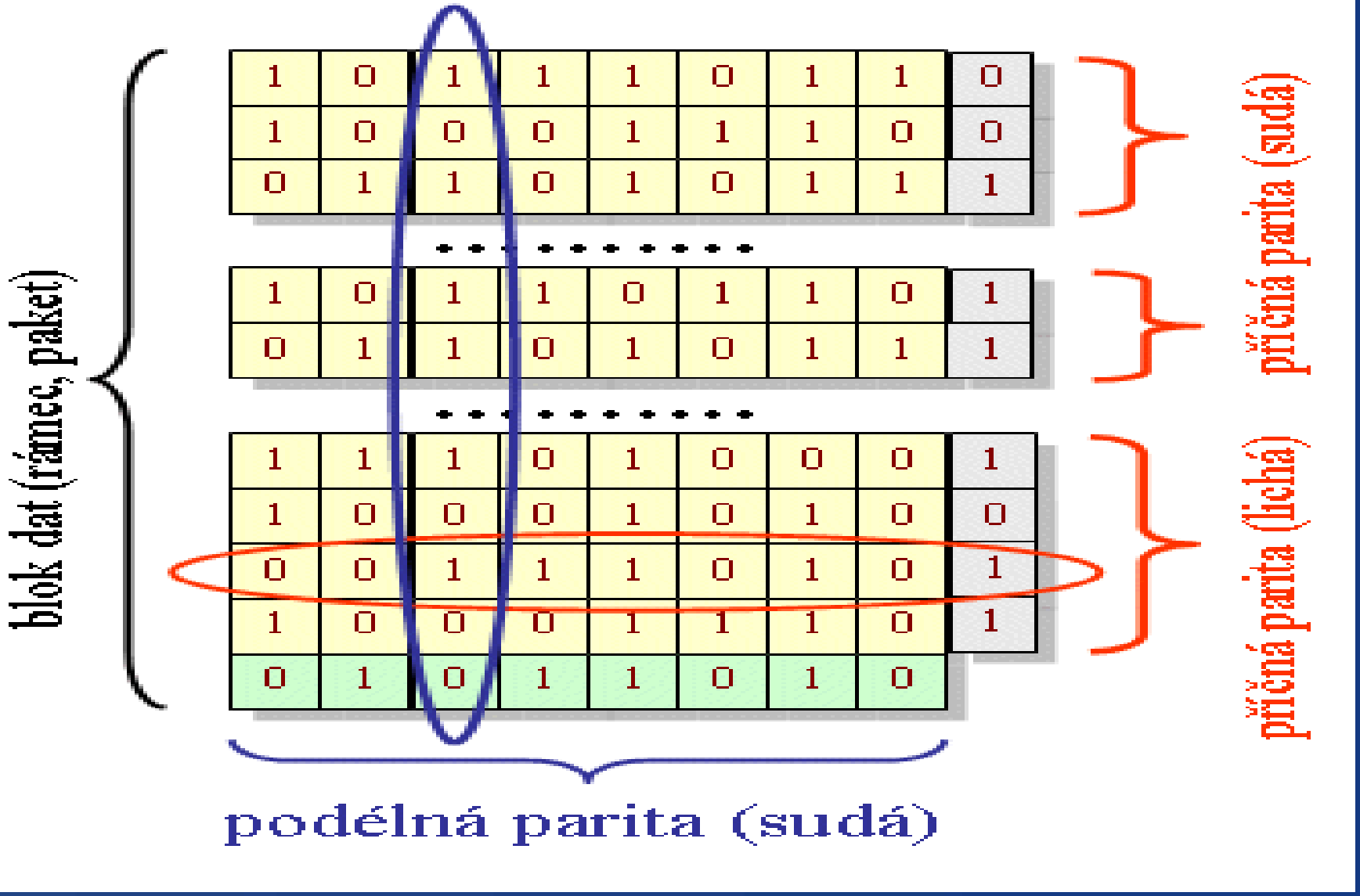
Bezpečnostní kódy

- Přidáme nějaké bity navíc nebo pozměníme data
- Čím více bitů navíc tím účinnější metoda
- Detekční – kontrola zda jsou data správně
- Samoopravné – chybu rozpoznají a opraví

Parita

- Přidáváme jeden paritní bit
- Sudá 0 = sudý počet 1, 1 = lichý počet 1
  - Vždy sudý počet 1 ve zprávě
  - Umí jen detekovat, nevíme co je špatně
- Lichá parita je analogie k sudé
- Příčná parita – paritní bit ke každému slovu
- Podélná parita – přidáváme paritní slovo, zabezpečuje celý blok, lze vyhodnocovat průběžně
- Křížová – kombinace příčné a podélné

Parita


blok dat (rámec, paket)
příčná parita (sudá)
příčná parita (lichá)
podélná parita (sudá)

Checksum

- Kontrolní součet – pro celý blok dat
- Jednotlivé znaky chápeme jako čísla bez znaménka
- Provádíme sčítání modulo 2^8 nebo 2^{16}
- Výsledek je číslo o délce 1 nebo 2 bytů
- Výpočet probíhá postupně
- Po přijetí kontrolní sumy se provede kontrola
- V případě chyby je nutné vyžádat přenos znovu

Hammingův kód (7,4)

- Dovoluje detekovat dvojitou a opravit jednoduchou chybu, 7 bitů z toho 4 datové
- Všechny bitové pozice, jejichž číslo je rovné mocnině 2, jsou použity pro paritní bit (1, 2, 4, 8, 16, 32, …).
- Všechny ostatní bitové pozice náleží kódovanému informačnímu slovu (3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, …).
- Každý paritní bit je vypočítán z některých bitů informačního slova. Pozice paritního bitu udává sekvenci bitů, které jsou v kódovém slově zjišťovány a které přeskočeny.

Hammingův kód

- Pro paritní bit p1 (pozice 1) se ve zbylém kódovém slově 1 bit přeskočí, 1 zkontroluje, 1 bit přeskočí, 1 zkontroluje, atd.
- Pro paritní bit p2 (pozice 2) se přeskočí první bit, 2 zkontrolují, 2 přeskočí, 2 zkontrolují, atd.
- Pro p3 (pozice 4) se přeskočí první 3 bity, 4 zkontrolují, 4 přeskočí, 4 zkontrolují, atd.
- http://en.wikipedia.org/wiki/Hamming_code
- http://www.uai.fme.vutbr.cz/~matousek/TIK/flashB5.html

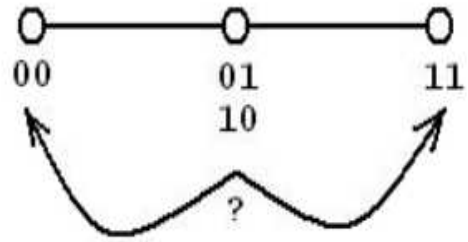
Rozšířený Hammingův kód (8,4)

- Na začátek každého slova přidáme paritu pro celé slovo
- Používá se sudá parita
- Dovoluje opravit jednu chybu, ale detekovat dvě

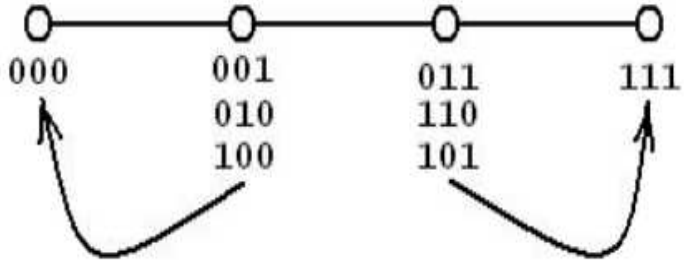
Hammingova vzdálenost I.

- Počet míst v němž se dvě kódová slova liší
  - příklad: 000 a 001 mají vzdálenost 1bit, 010 a 101 mají vzdálenost 3bity
- Charakterizuje odolnost kódu proti poruchám a schopnost identifikovat a případně opravit chyby
- Minimální Hammingova vzdálenost = minimální vzdálenost mezi všemi možnými páry vektorů

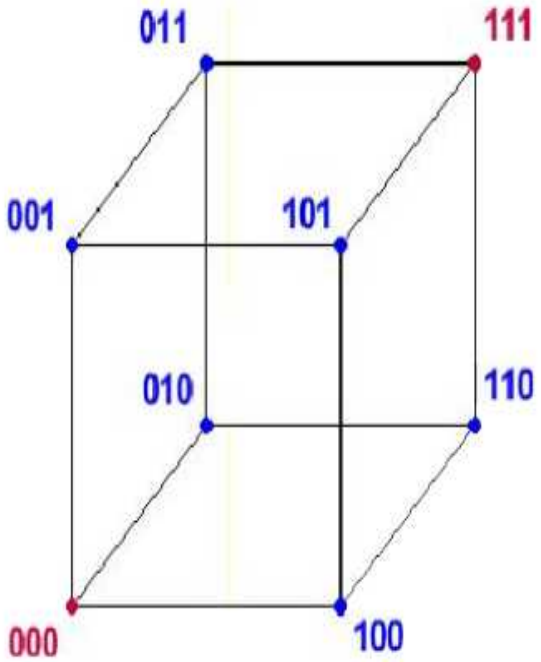
Hammingova vzdálenost II.



Minimální Hammingova vzdálenost kódu je 2.
Jednobitová chyba jde detekovat, ale nelze opravit.



Minimální Hammingova vzdálenost kódu je 3.
Jedno a dvoubitová chyba jdou detekovat.
Opravit lze pouze jednobitovou chybu.



Hammingova vzdálenost III.

- Pro detekci n bitovych chyb platí
  - $d_{min} => n+1$; tj $n <= d_{min} - 1$
- Pro detekci a korekci n bitovych chyb platí
  - $d_{min} => 2n+1$; tj $n <= (d_{min} - 1)/2$
  - D(000,001) = 1, nevíme nic
  - D(000,101) = 2, poznáme jednu chybu
  - D(000,111) = 3, 2 poznáme, 1 opravíme

Cyklické kódy CRC

- Cyklický redundantní součet
- CRC s počítá před operací kde čekáme chybu
- Odesílá se společně s daty
- Po přenosu se spočítá znovu a rozhodne se
- Někdy je možné chybu i opravit
- Např. Generující polynomy $G(x)=x^4+x+1$, tedy $(10011)_2$
- Délka zabezpečení se rovná stupni generujícího polynomu

Cyklické kody CRC

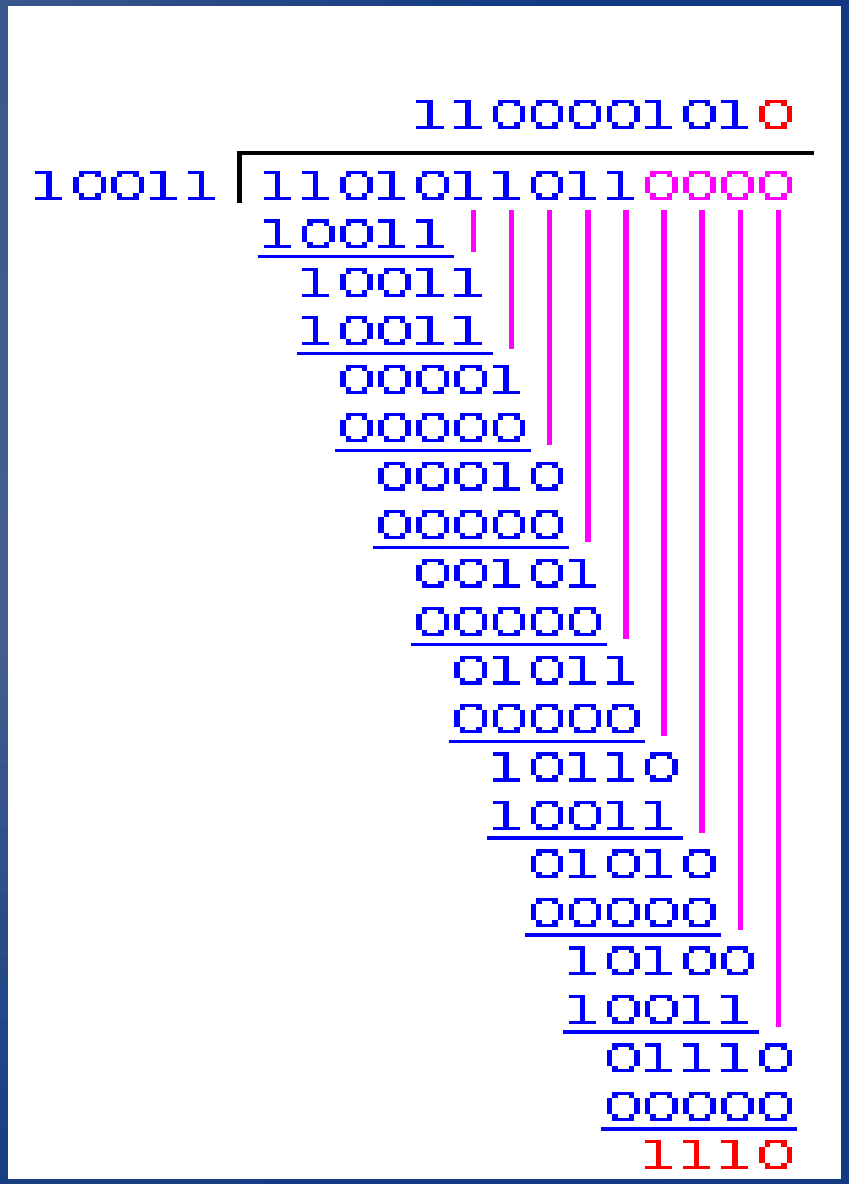
- Vypočteme zbytek po dělení R(x)=M(x)/G(x)
- Odesíláme T(x) = M(x) | R(x)
- Po přijetí provedeme T(x)/G(x)
- Pokud je výsledek (zbytek) nula, je přenos v pořádku
- Označení jako CRC 16, 32 atp. podle stupně polynomu G(x)
- http://en.wikipedia.org/wiki/Cyclic_redundancy_check

CRC příklad

- M(x)=1101 0110 11
- G(x) = 10011 = $x^4 + x + 1$
- Délka zabezpečení je rovna stupni generujícího polynomu, tj. k=4. Vypočteme zbytek po dělení $M(x) * x^4$
- 11 0101 1011 0000 / 10011
- R(x) = 1110

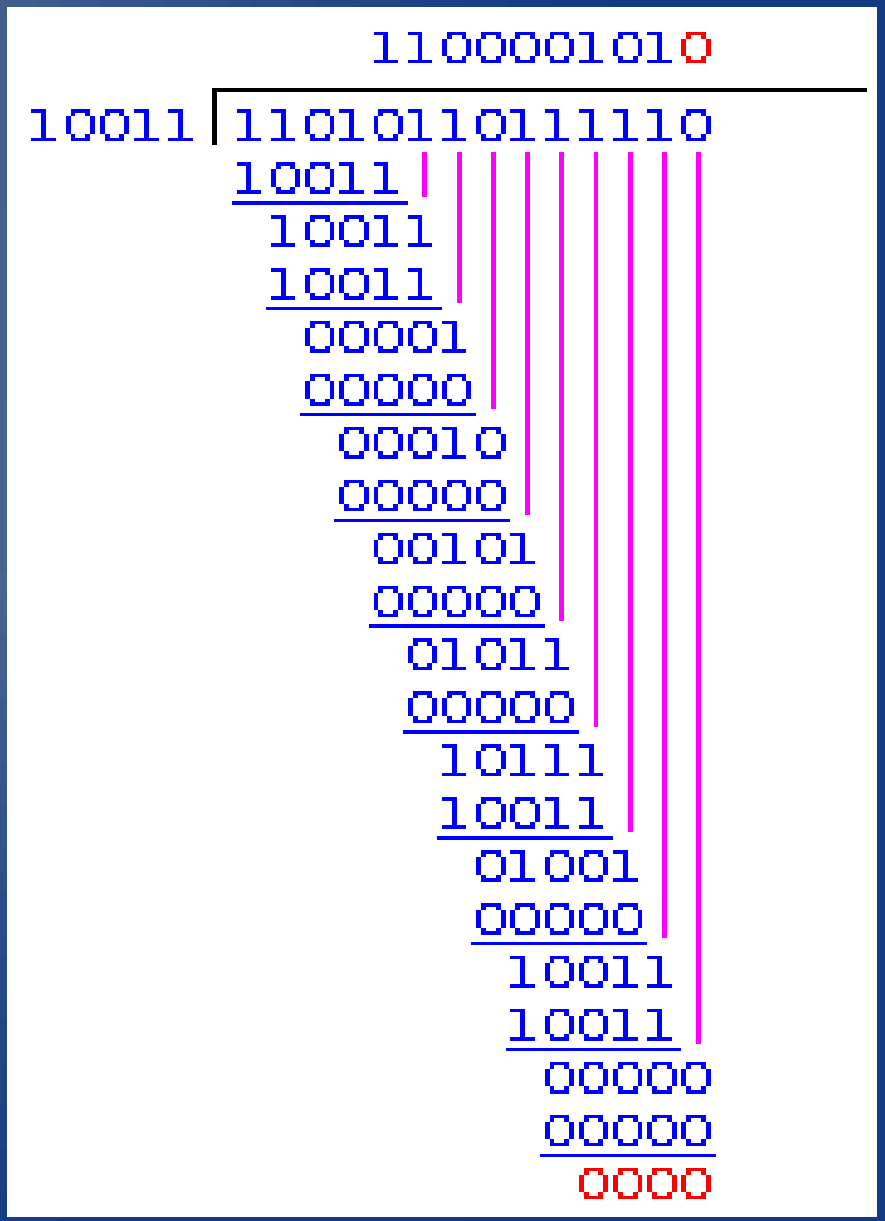
CRC příklad

- Postup dělení
- Stejné jako dělení pod sebe
- Operaci odečítání nahrazuje operace XOR
  - 1 XOR 1 = 0
  - 1 XOR 0 = 1
  - 0 XOR 1 = 1
  - 0 XOR 0 = 0
- Odesíláme M(x) | R(x)
  - 1101 0110 11 | 1110



CRC příklad

- Ověření přijaté zprávy

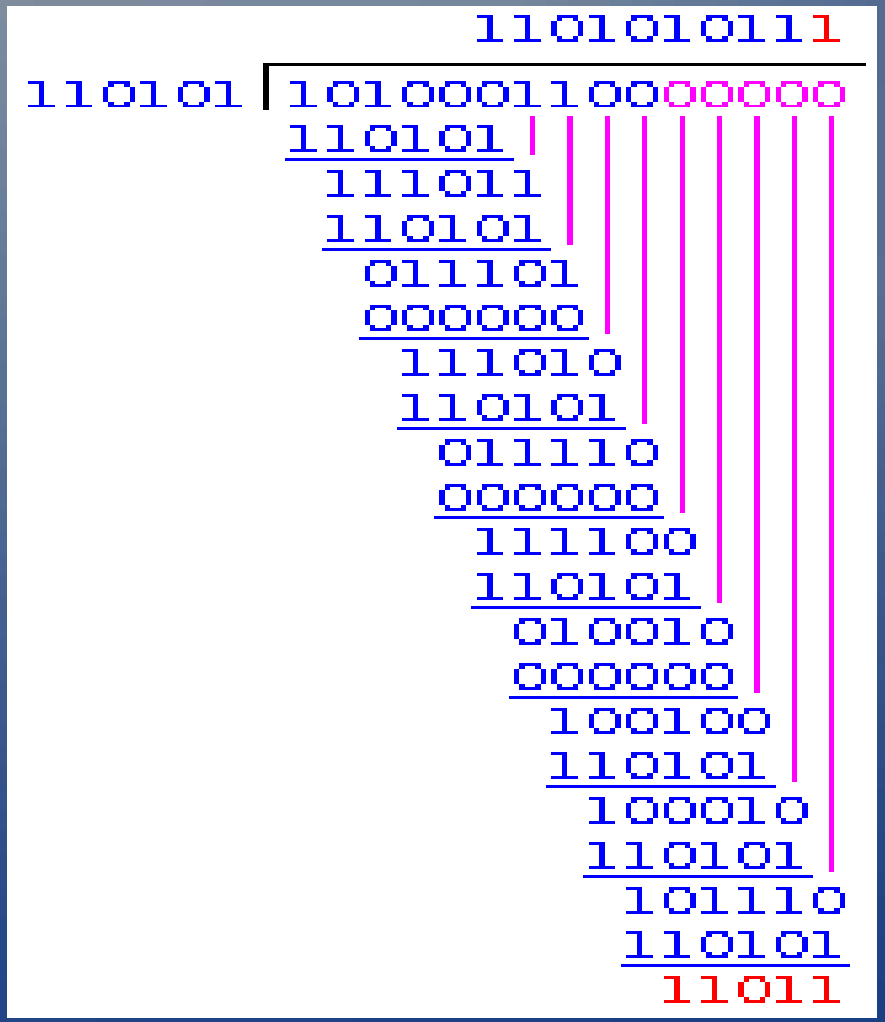


CRC samostatně

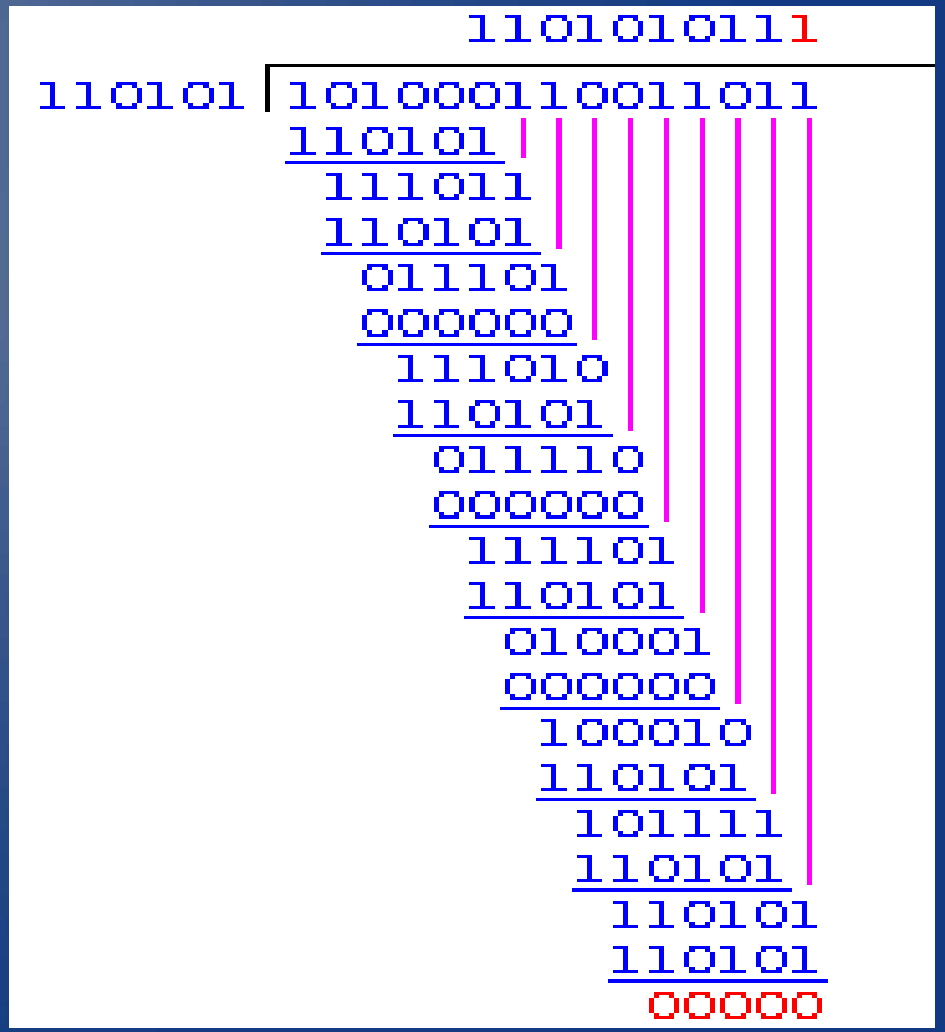
- M(x) = 10 10 00 11 00
- M`(x) = 10 10 00 11 00 00 00 0
- G(x) = 11 01 01 = $x^5 + x^4 + x^2 + 1$
- R(x) =
- T(x) =

CRC samostatně

Zabezpečení



Kontrola



CRC samostatně

- Zkoušejte si na
  - http://www.macs.hw.ac.uk/~pjbk/nets/crc/

UPS 2015/2016

Cvičení 8

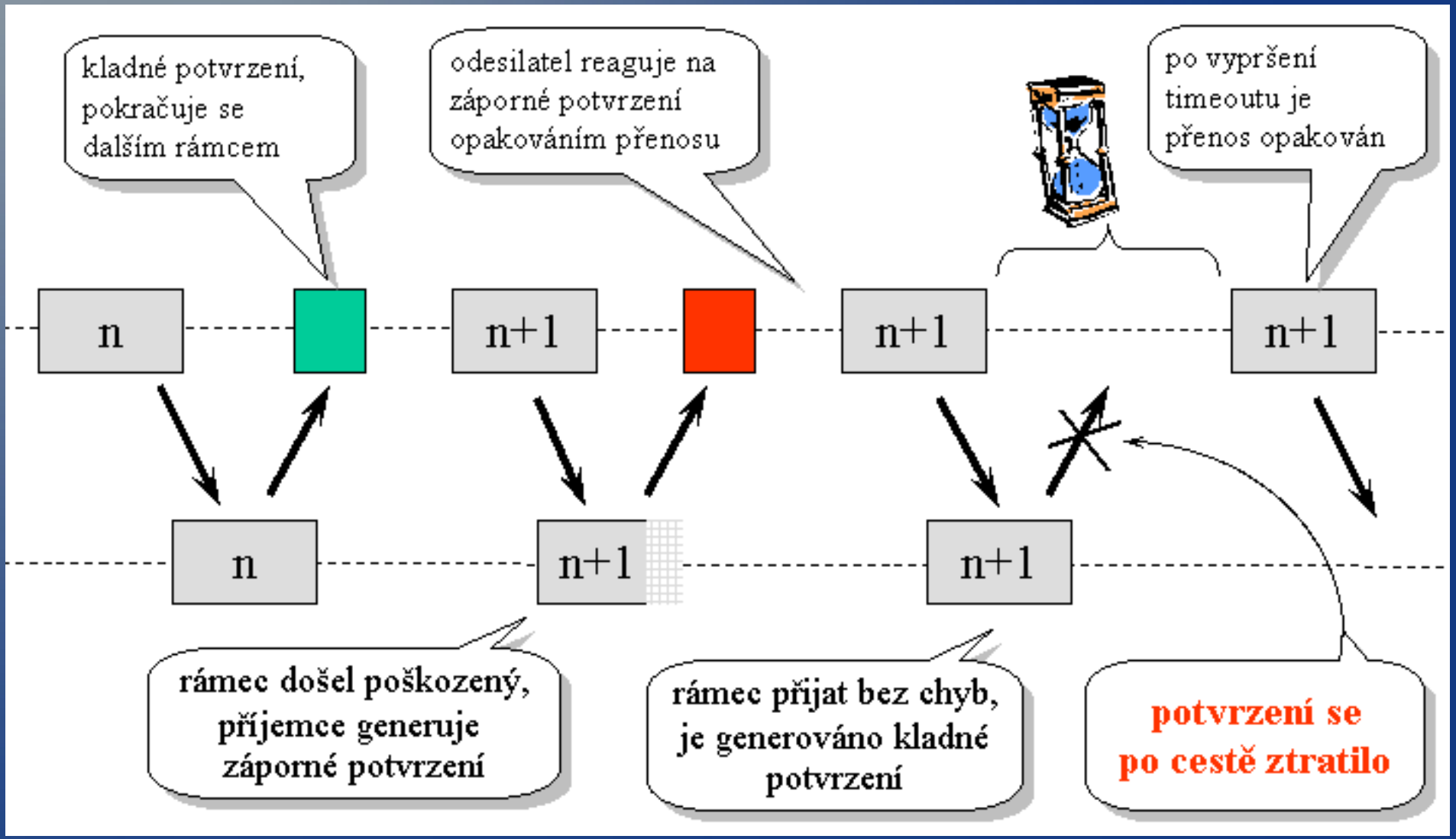
Obsah

- Kladné a záporné potvrzování
- Protokol Stop-and-wait
- Využití kapacity přenosového kanálu
- Průběžné potvrzování
  - Selective repeat
  - Go-Back-N
- Klouzající okénko
- Petriho sítě

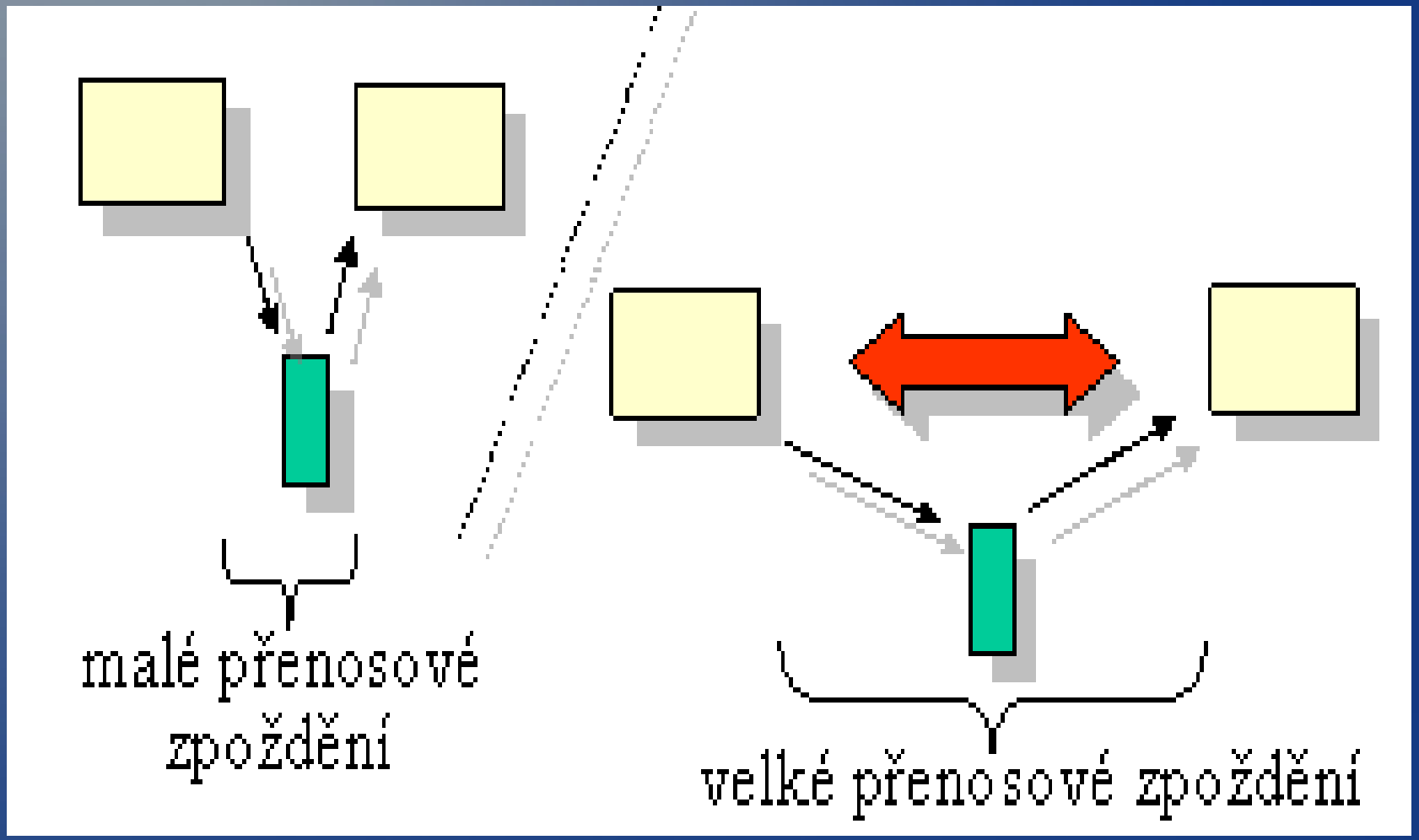
Potvrzování

- Obecně
  - pozitivní ACK
  - negativní NACK, často pouze implicitni pomocí timeoutu
  - kombinované ACK i NACK
  - s časovým limitem - timeout
- Způsob
  - Samostatné - extra rámec
  - nesamostatné - Piggybacking - přibalení
  - skupinové (samostatné/nesamostatné)
- http://webmuseum.mi.fh-offenburg.de/index.php?view=exh&src=30

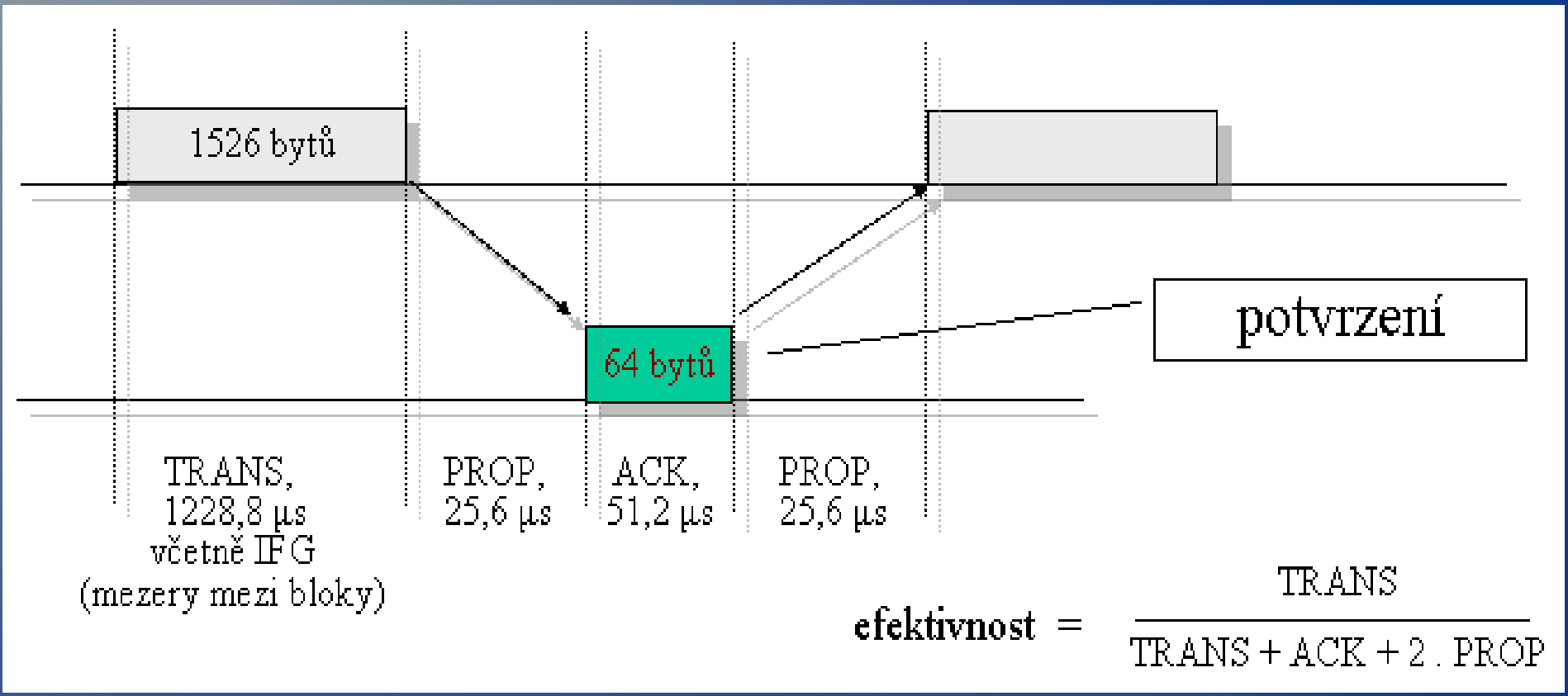
Stop and Wait



Stop and Wait



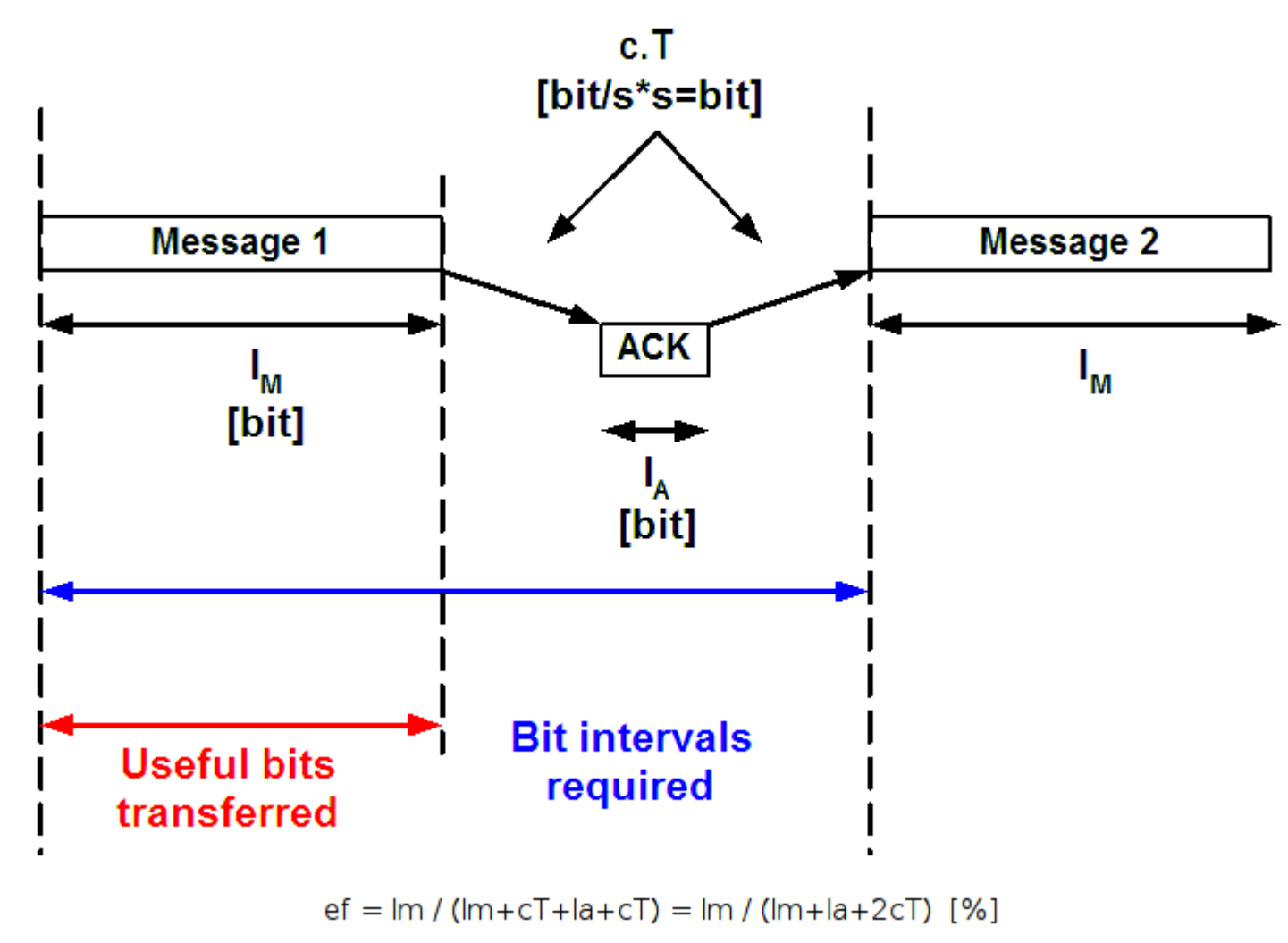
Využití kapacity přenosového kanálu



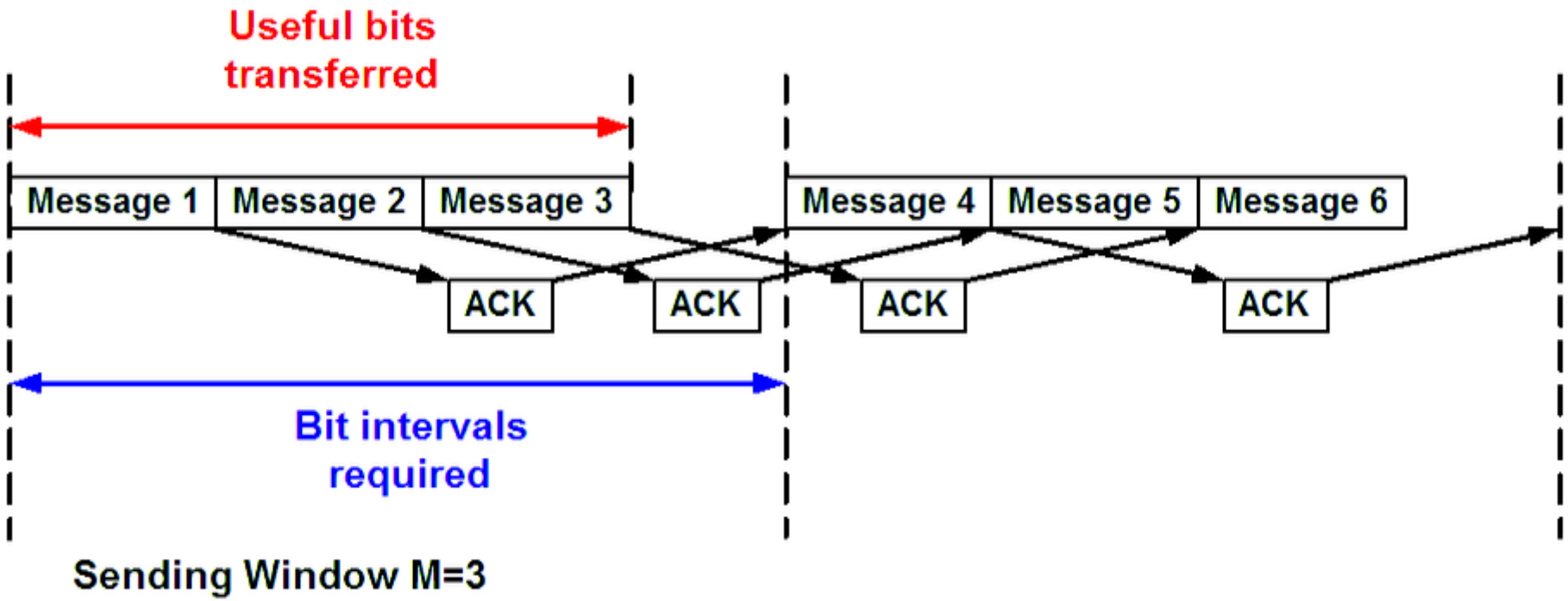
Využití kapacity přenosového kanálu

- Modemová linka
  - lm=80B, la=1B, c=14400 bps, T=1ms, ef=94.56 %
- Družicový spoj
  - lm=80B, la=1B, c=14400 bps, T=270 ms, ef=7.6 %
- 8x prodlouzeni ramce
- Modemová linka
  - lm=640B, la=1B, c=14400 bps, T=1ms, ef=99.28 %
- Družicový spoj
  - lm=640B, la=1B, c=14400 bps, T=270 ms, ef=40.38 %

Využití kapacity přenosového kanálu

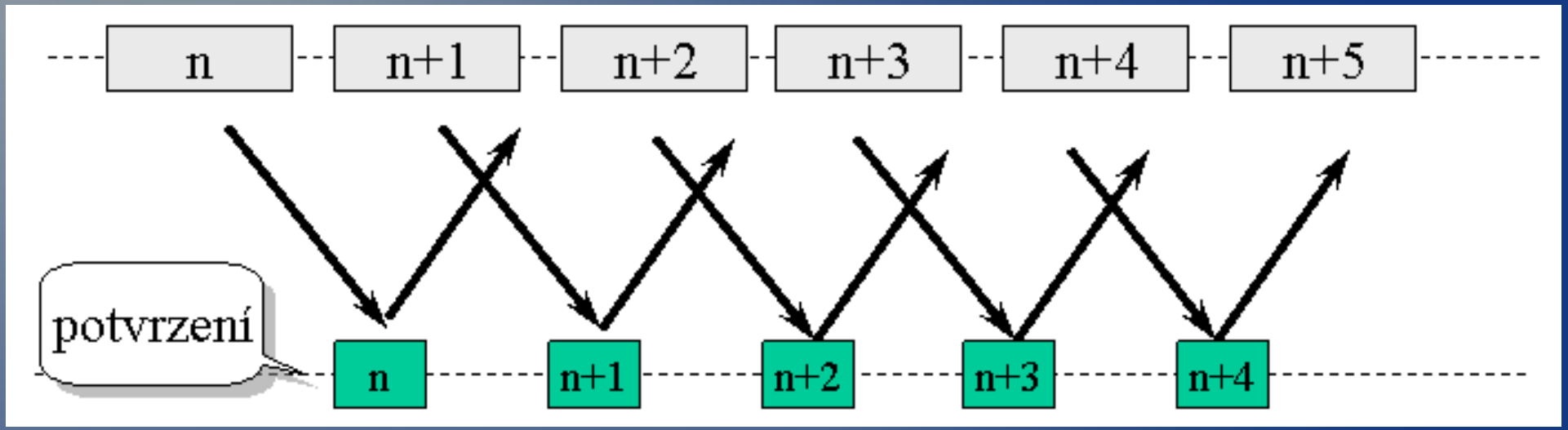


Využití kapacity přenosového kanálu



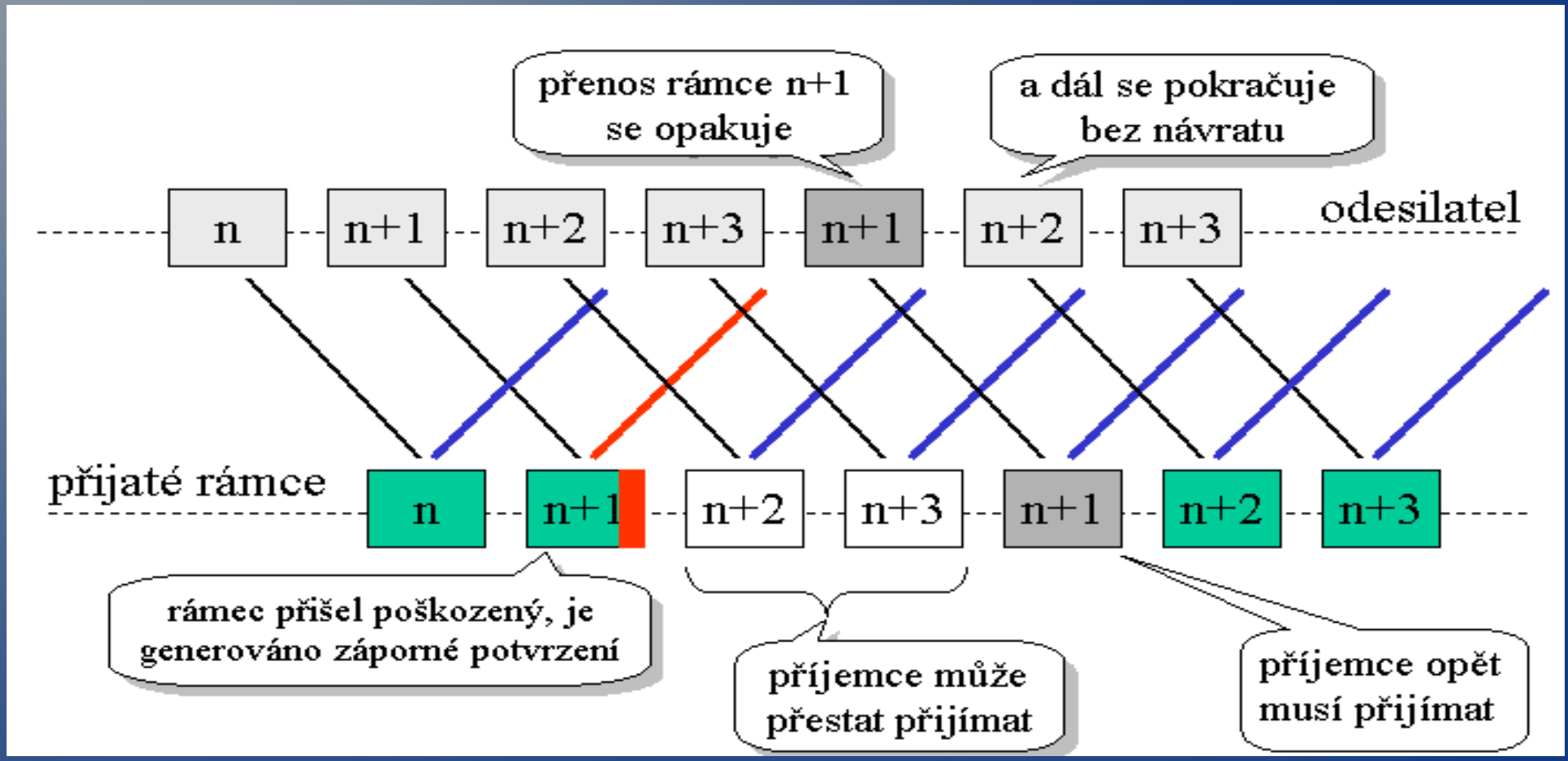
$$ef = M.lm / (lm+cT+la+cT) = M.lm / (lm+la+2cT) \ [\%]$$

Continous ARQ



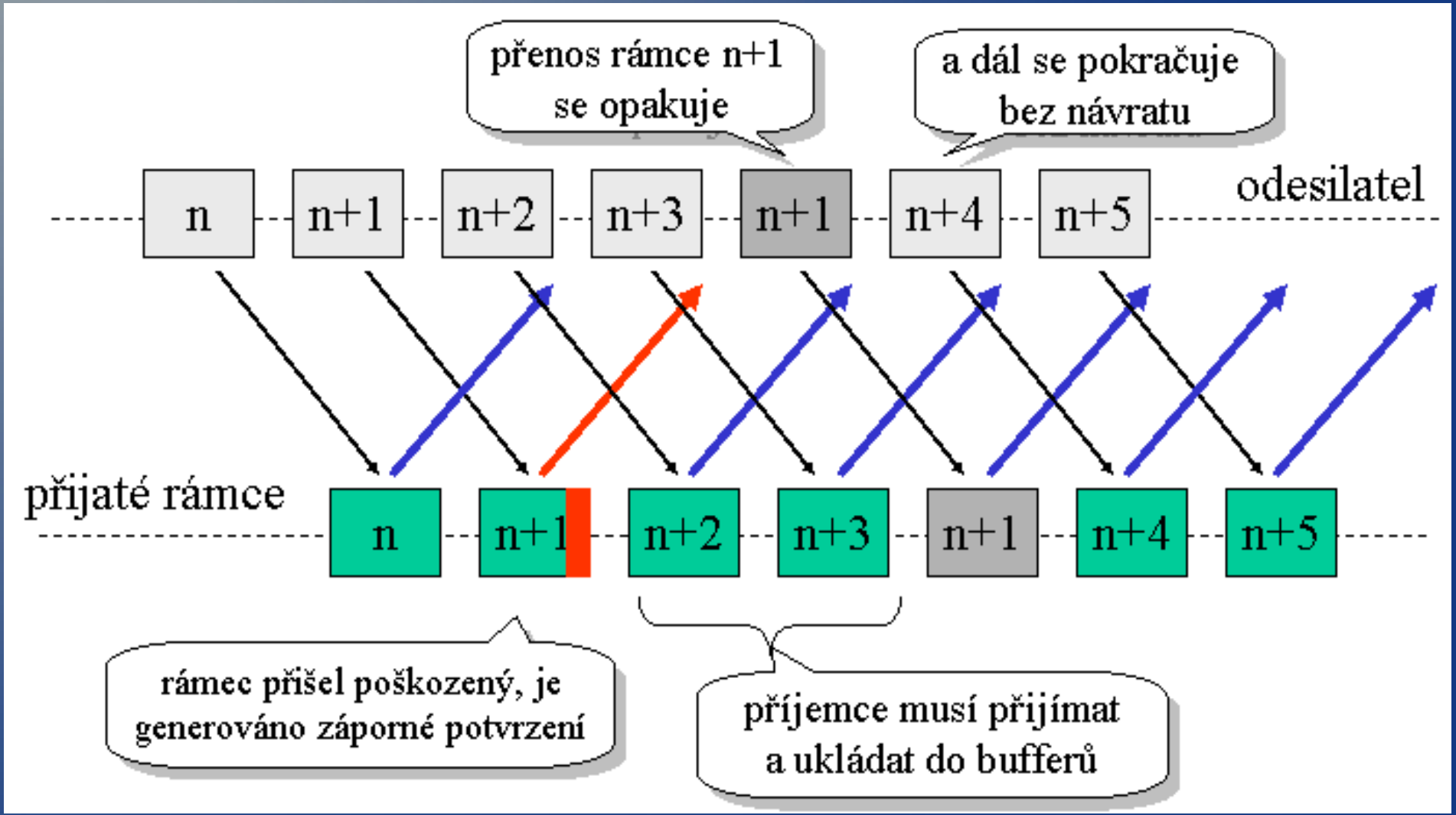
- Jak resit ztratu dat/potvrzeni
- Buffer/okenko
  - vysílací, příjímací

Go-Back-N



- http://www.eecis.udel.edu/~amer/450/TransportApplets/GBN/GBNindex.html

Selective repeat



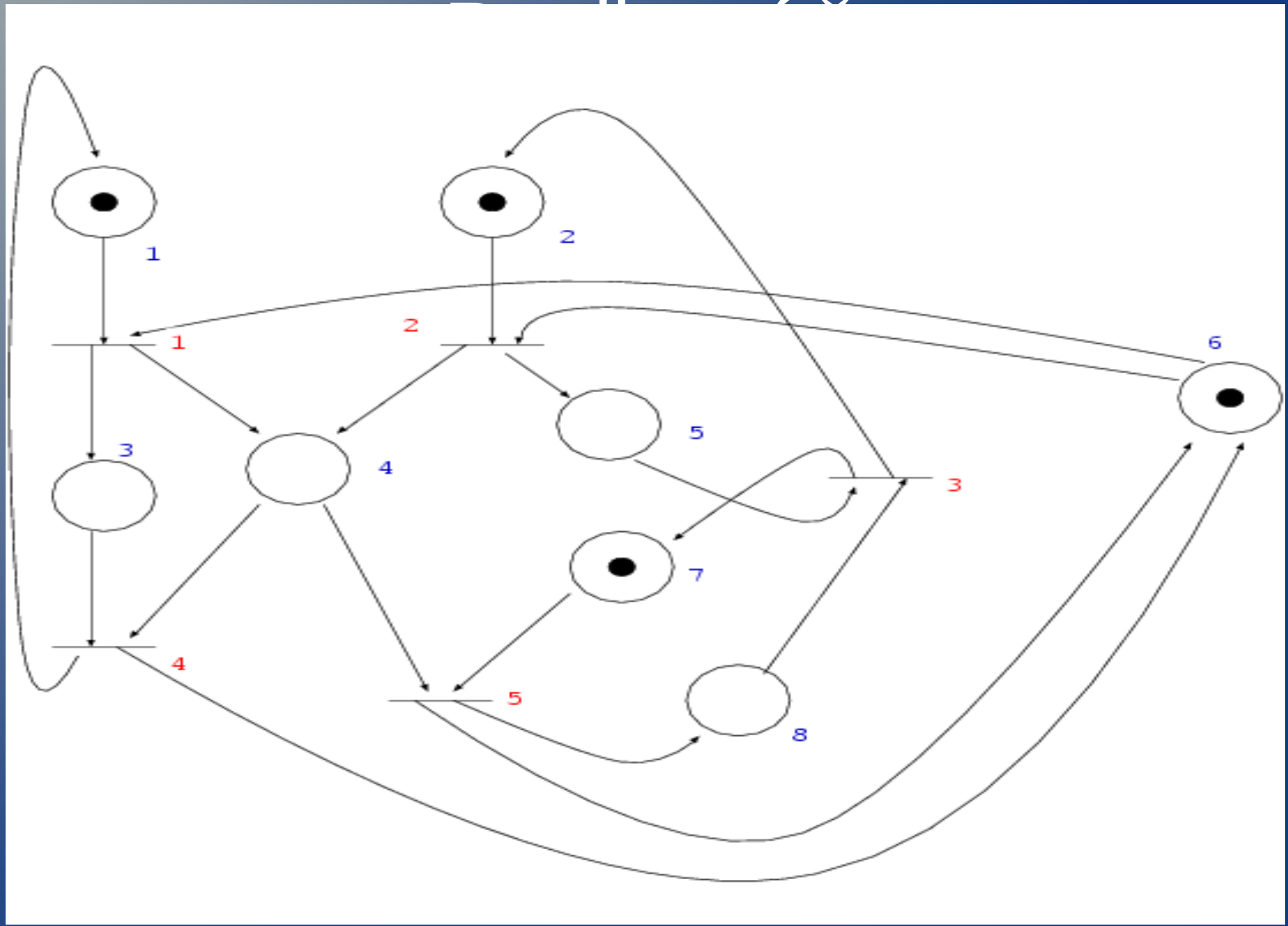
- http://www.eecis.udel.edu/~amer/450/TransportApplets/SR/SRindex.html

Klouzající okénko

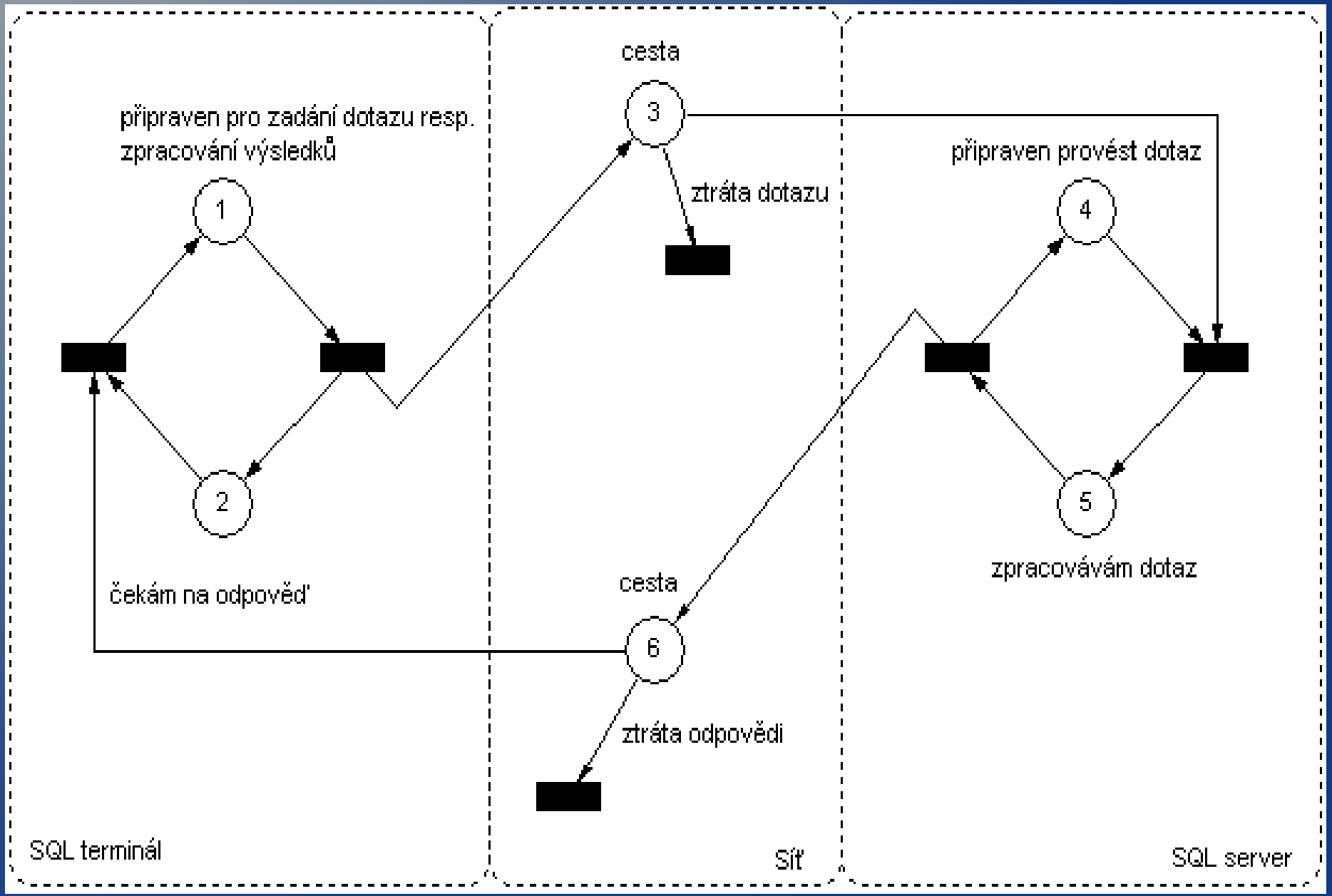
- Můžeme vysílat více rámců – nutné číslování
- Vysílací/přijímací okénko – buffer
- Každý rámec má svůj časovač
- Při správném přijetí ACK
  - Continuous ARQ – kontinuální kladné potvrzování
- Při nesprávném nic nebo NACK
- Šířka může být pevná nebo potvrzovaná protokolem
  - U TCP pro řízení toku dat

Petriho sítě

- Matematický model diskrétních distribuovaných systémů
- Místa, přechody, hrany
- Hrany jsou
  - Vstupní z místa do přechodu
  - Výstupní z přechodu do místa
- Místa obsahují libovolný počet teček
- Pokud je na každém vstupu alespoň jedna tečka dojde k odpalu/posunu v rámci kroku
- Pohyb je nedeterministický


1
2
1
2
6
3
4
5
3
7
4
5
8

Příjem a odeslání Petriho sítí



UPS 2015/2016

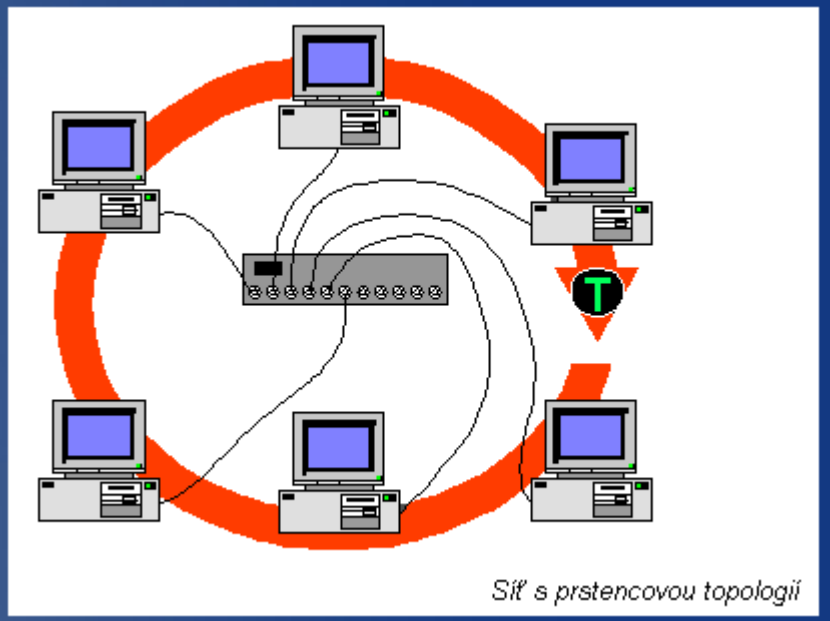
Cvičení 9

Obsah

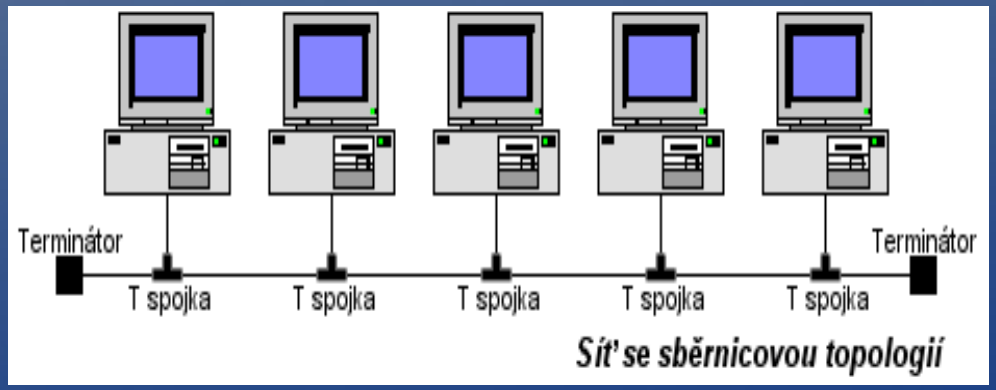
- Řízení v lokálních počítačových sítích
  - Centralizované a decentralizované
- Decentralizované přístupové metody
  - Aloha, CSMA, CSMA/CD, kolize,
  - minimální délka rámce
  - Token Ring, Token Bus
  - priority v sítích s předáváním pověření
  - sítě s prioritním přístupem, výpočet velikosti okénka
  - Ethernet.

LAN

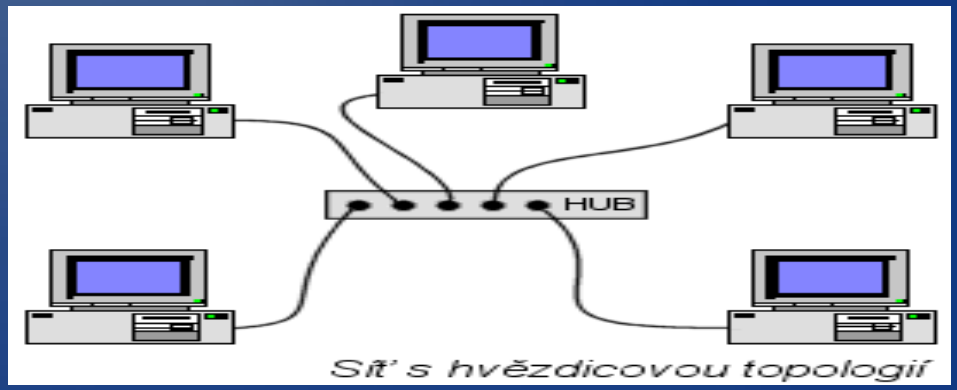
- Local Area Network
  - Typicky více bodové spoje
  - Sběrnice, hvězda, kruh
- Wide Area Network
  - Typicky dvoubodové spoje



Síť s prstencovou topologií



Síť se sběrnicovou topologií



Síť s hvězdicovou topologií

Řízení přístupu

- V rámci LAN
- Příjem nevadí, problém je vysílání
- Společné přenosové medium
- Nutnost řízení
  - Algoritmus
  - Uzel
- Kolize
  - Téměř současné vysílání
  - Nelze zpětně oddělit
  - Vyloučení x detekce

Přístupové metody

- Rozšiřuje ISO/OSI
  - Linková vrstva
    - LLC – Logical Link Control - původní
    - MAC - Media Access Control - řízení přístupu
- Typy detekcí
  - Zcela vylučuje kolize – CA, Collision Avoidance
  - Detekuje kolize – CD, Collison Detection
  - Bez detekce kolizí

Přístupové metody

- Charakter řízení
  - Řízené – deterministické
  - Neřízené - nedeterministické
- Existence arbitra - vedoucího
  - Centralizované – centrální prvek
  - distribuované
- Dektece volnosti pásma a následné vysílání
  - Stejně muže nastat kolize

Přístupové metody - centralizované

- Existuje centrální prvek, který přiděluje kanál
  - Výzva – Chces vysílat ??
    - Cykický výběr, štafeta
  - Žádosti – Chci vysílat !!
- Arbitr se může měnit
- Problém při výpadku arbitra a změna topologie
- Vždy řízený přístup
- Neřízený nemá význam

Přístupové metody - centralizované

- CMTS pro kabelové sítě
  - Rezervační rámec, kde uzel projeví zájem vysílat
- Demand Priority
  - Stromová struktura sítě, vždy mám nadřízeného, dvě úrovně priorit

Přístupové metody - decentralizované

- Řízené algoritmem
- Rezervační rámec
  - Koluje sítí a stanice se registrují
- Prioritní přístup
  - Umístění – co je vlevo má přednost
  - Čas – čím vyšší priorita tím kratší čekání po kolizi
  - Problém monopolizace
  - Kombinace dynamické a statické priority
    - Když jednou prohraji zvýším dynamickou o jedna

Přístupové metody - decentralizované

- Aloha
  - Neřízená distribuovaná metoda, 1970
  - Využívá radivý přenos v éteru
  - Nekontroluje stav, prostě pošle zprávu
  - Kontrola doručení podle potvrzení, ale na vyšší vrstvě
  - Nízské využití kanálu – cca 18%
- Slotted Aloha
  - Vysílání jen ve stanovený čas – sloty
  - Až 36% využití kanálu
- Synchronní Aloha – vysílání na písknutí centrální stanice

Přístupové metody - decentralizované

- CSMA
- Carrier Sence – detekuju nosnou vlnu, pokud je čekám
- Multiple Access – vysílá více uzlů, přijímají všichni
- Dochází ke kolizím, detekuje jen před začátkem vysílání
- Přenese se celý rámec, chybu musí odhalit příjemce
- Naléhající – čeká na konec hned vysílá
- Nenaléhající – přeplánuje se na později
- P-naléhající – s p% se chová jako naléhající
  - Ideální pro p 5-10%, využití až 95% kanálu

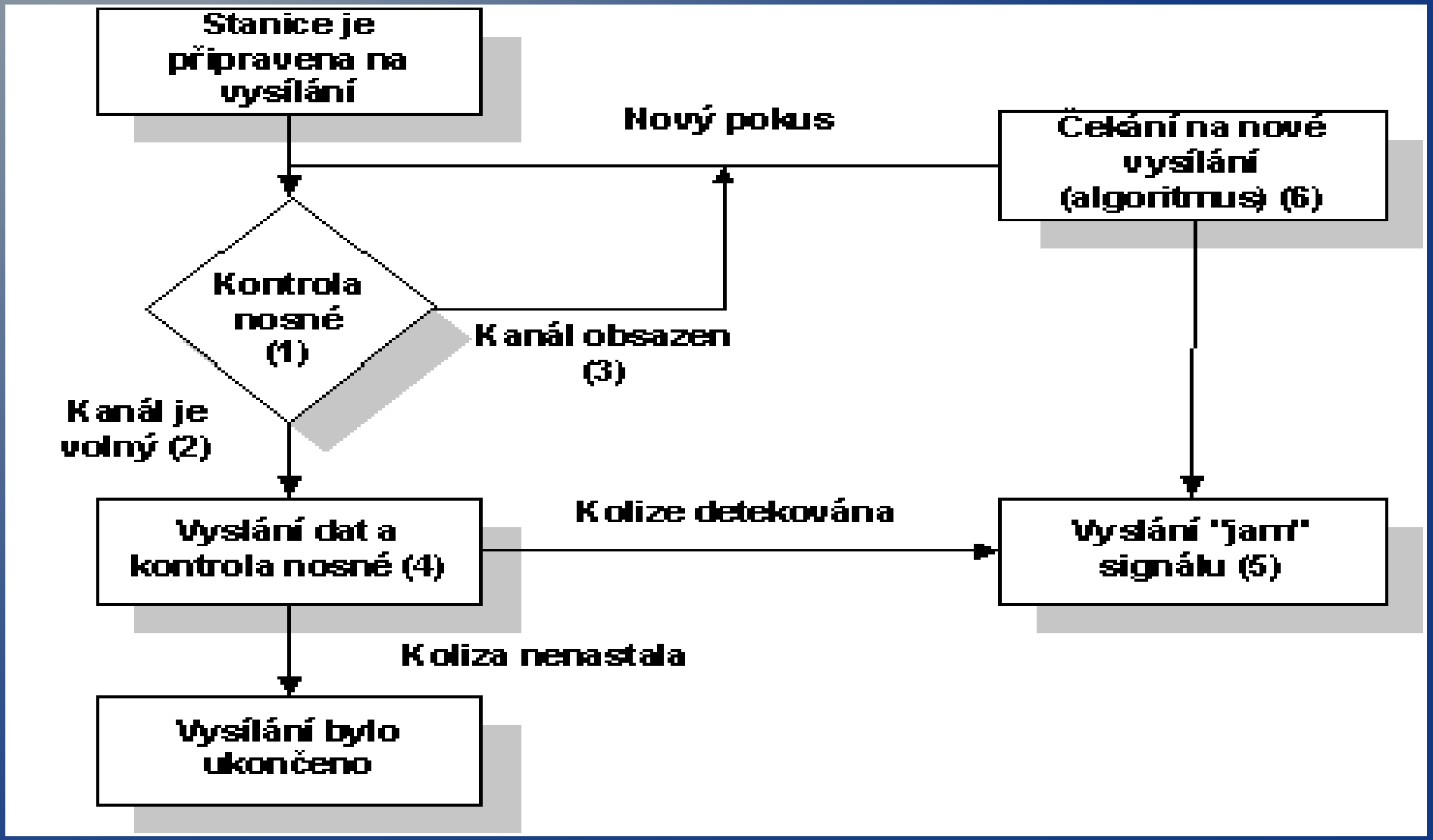
Přístupové metody - decentralizované

- CSMA/CA
- Předchází kolizím
- Každý uzel informuje ostatní o úmyslu vysílat
- Minimalizujeme kolize, ale můžou nastat
- Neumíme detekovat
- Využití v bezdrátových sítích, kde nelze provést současně vysílání i příjem nebo Apple - LocalTalk

Přístupové metody - decentralizované

- CSMA/CD
- Distribuovaná, neřízená metoda
- Detekuje kolize a okamžitě zastavuje vysílání
- Náhodný interval čekání na další vysílání
- Při opakování dobou zdvojnásobuje
- Zároveň kontroluje zda je linka volná a pokud ano vysílá
- Během přenosu detekuje aktivitu ostatních
- Mnohem lepší využití media, neplýtvá se časem při odeslání celých rámců
- Nelze použít všude, potřebuje přídavnou elektroniku na detekci kolizí

Přístupové metody - decentralizované



Kolizní okénko

- Doba po kterou signál zaplní celé přenosový kanál
- Závislost
  - Rychlost světla, délka média, zpoždění v aktivních prvcích
  - Musí být menší než minimální délka rámce
    - Předcházení nezjištěným kolizím
  - Rámec nesmí být příliš krátký
  - Maximální délka media a počet opakovačů jsou omezeny
  - Komplikuje zvyšování přenosové rychlosti

Přístupové metody - decentralizované

- CSMA/BA nebo CSMA/CR
- Bitová arbitráž
- Každý uzel má ID nebo prioritu
- Při kolizi vysílá ten s vyšším ID
  - Nemusí se čekat náhodnou dobu
- Běžné v rámci CAN – vozidla

Předávání pověření

- Pověření - token
- Token je předáván mezi uzly
- Tvoří logický kruh
- Problém ztráty tokenu

Token Ring

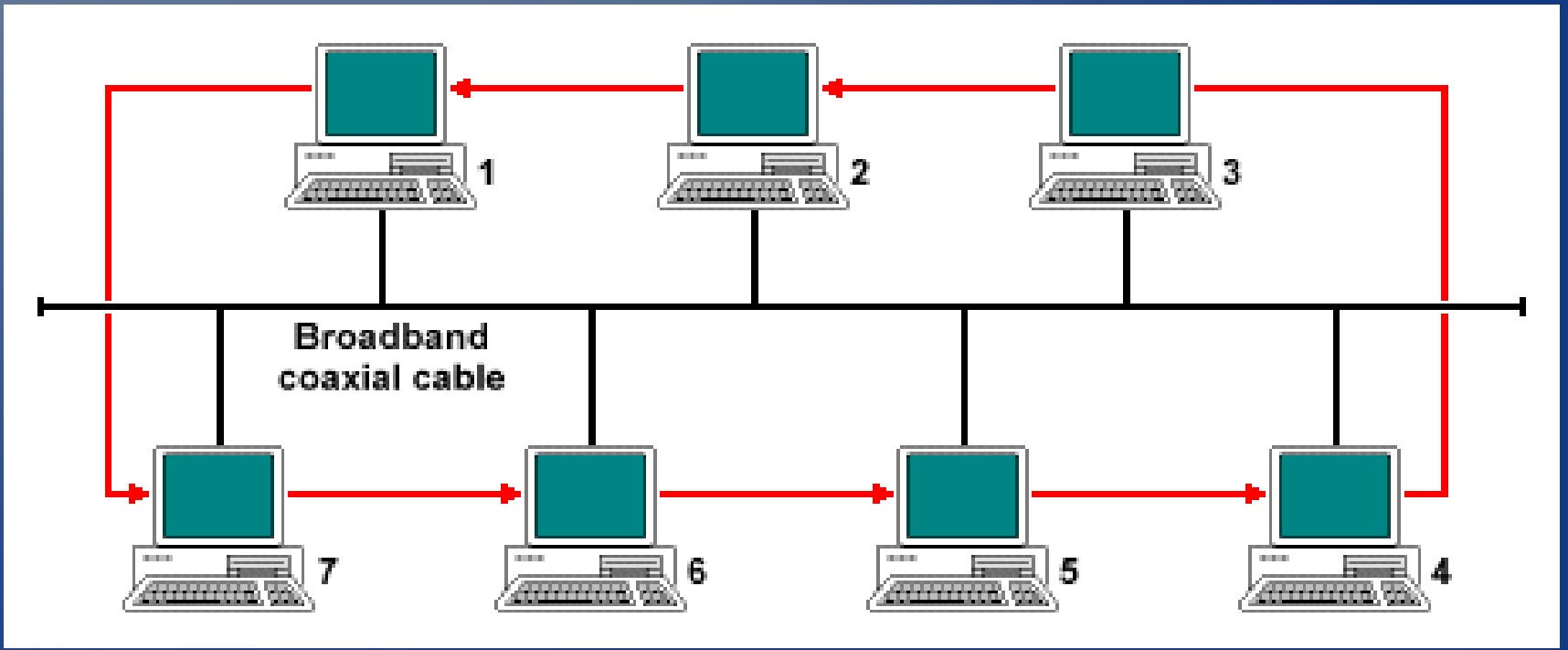
- Distribuovaná a řízená metoda předávání pověření od IBM
- IBM Token ring – zapojení do hvězdy, kroucená dvojlinka
- IEEE 802.5 – nepředepisuje žádnou topologii ani medium
- Logický kruh
- Lepší při větším zatížení než Ethernet, například ARCNET, Token Bus, FDDI
- Diferenciální manchester
- Když nikdo nevysílá posílá se jen prázdný token
- Pokud nekoluje žádný token nebo je jich více, zasáhne vyčleněná stanice - aktivní monitor, kterou může být kdokoliv – zařízení s nejvyšší MAC

Token Ring

- MAU MultiAccess Unit – rozbočovač – tvoří kruh

Token Bus

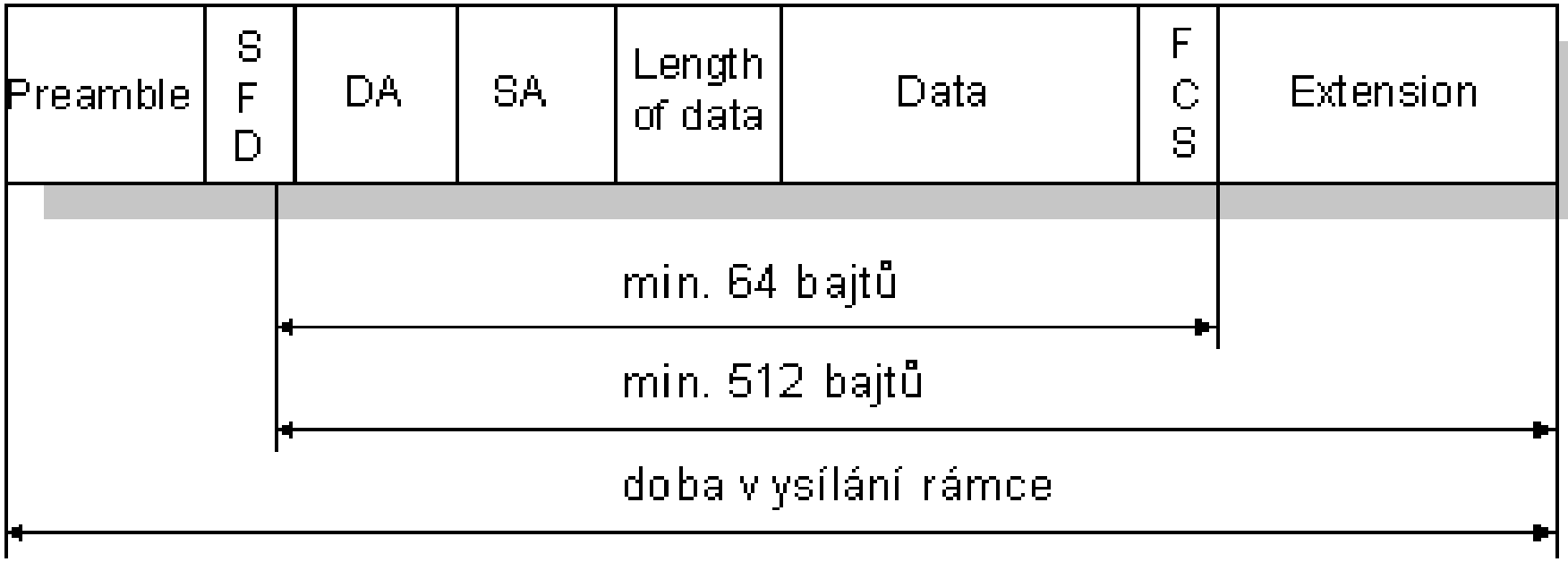
- Využívá metody předávání pověření
- Sběrnicová topologie
- Kruh je pouze logický



Ethernet

- Distribuovaná a neřízená metoda přístupu
- Využívá CSMA/CD
- Při detekci kolize se zašle JAM - 32 bitů a všichni se na chvíli odmlčí
- Čekání je náhodnou dobu, interval se při prvních deseti pokusech zdvojnásobuje
- Pokusů je celkem 16, pak se nahlásí chyba
- Velice efektivní při malém zatížení sítě
- Lepší pro delší rámce

Ethernet



SFD ... Start of Frame Delimiter
DA ..... Destination Address
SA ..... Source Address
FCS ... Frame Check Sequence

Ethernet

- Preambule – 8 bytů, strídá 0 a 1 a poslední 10101011 – SFD, slouží na synchronizaci
- Cílová a zdrojová adresa
- Typ protokolu
  - Ethernet II – typ vyššího protokolu
  - IEEE 802.3 – délka dat
- Datová 46B-1500B
- Datová výplň – doplněk na 64B
- Kontrolní součet, FCS, 32b CRC

Transparentní mosty - Bridge

- Spojuje sítě na L2
- Transparent bridging
  - Neviditelný pro koncové stanice
  - Postupně se učí co kde leží
- Source route bridging
  - Pro propojení s token-ring
  - Packet musí obsahovat i cestu přes mosty
  - Je třeba znát cestu

Transparentní mosty - Bridge

- Výhody
  - Není potřeba konfigurovat
  - Snižuje velikost kolizní domény
  - Transparentní pro vyšší protokoly
  - Lacinější než router
- Nevýhody
  - Neomezuje všesměr
  - Vyšší latence – manipulace s MAC
  - Dražší než opakovače
  - Přemosťování různých MAC vede k chybám

Spanning Tree - STP

- Mechanismus předcházení kruhu v síti
- Problém smyček
  - Broadcastové bouře
  - Problém s konektivitou
  - Násobné doručování zpráv
- STP volí root kořen a tvoří strom podle cen linek
- Vychází z TGD
- Typický problém ve větších sítích

UPS 2015/2016

Cviceni 9

Obsah

- Opakování před testem
- Zpoždění, stanovení délky okénka
- Režimy přenosu
- Řízení přístupu
  - Centralizované (výzva, žádost)
  - Decentralizované (soutěž, předávání pověření)

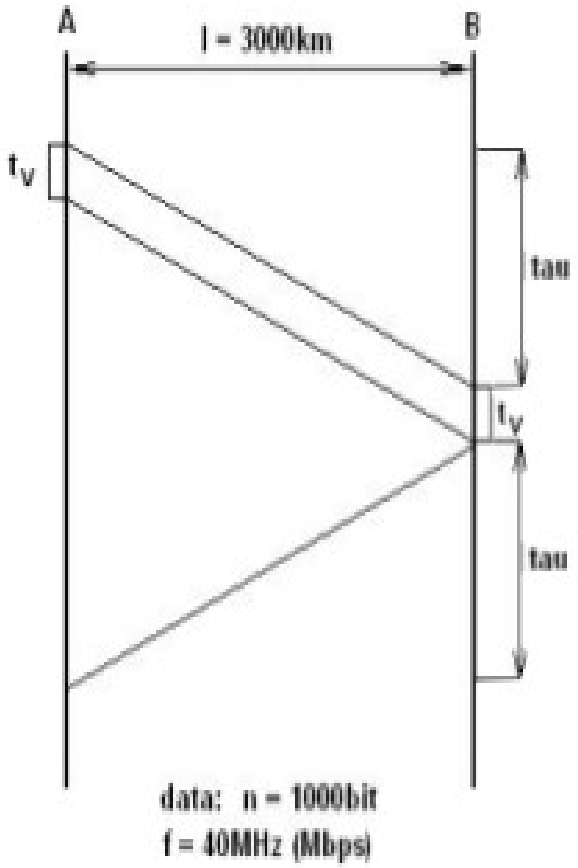
Obsah testu I.

- Kódování (RZ, RZI, NRZ, NRZI, Manchester, diferenciální Manchester)
- Využitelnost přenosové kapacity (data vs. režie)
- Transparentnost přenosu (escapování)
- Vlastnosti přenosového kanálu (Shannon, Niquist)
- Modulace (fázová, amplitudovaní, frekvenční)
- Zabezpečení přenosu (parita, kódování, kontrolní součet)

Obsah testu II.

- Protokoly TCP/IP (protokoly jednotlivých vrstev)
- Zásobník ISO/OSI, TCP/IP (vsrtvy a jejich funkce)
- Znakově / bitově orientované protokoly (formát zpráv)

Zpoždění



$t_v = n/f = 10^3/4*10^7 = 25$microsec.

tau = $l/v = 3*10^6/2*10^8 = 15$ milisec.

$t = t_v + 2*tau = 30.025$ milisec.

účinnost = $t_v/t = 25*10^{-6}/30*10^{-3} = 0.00166 = 0.2\%$

velikost okénka:

1. $t * f = 0.030025 \times 40*10^6 =$
   $= 1201000$ bitů = 1201 rámců
2. $t / t_v = 30.025 / 0.025 = 1201$ rámců

Režimy přenosu

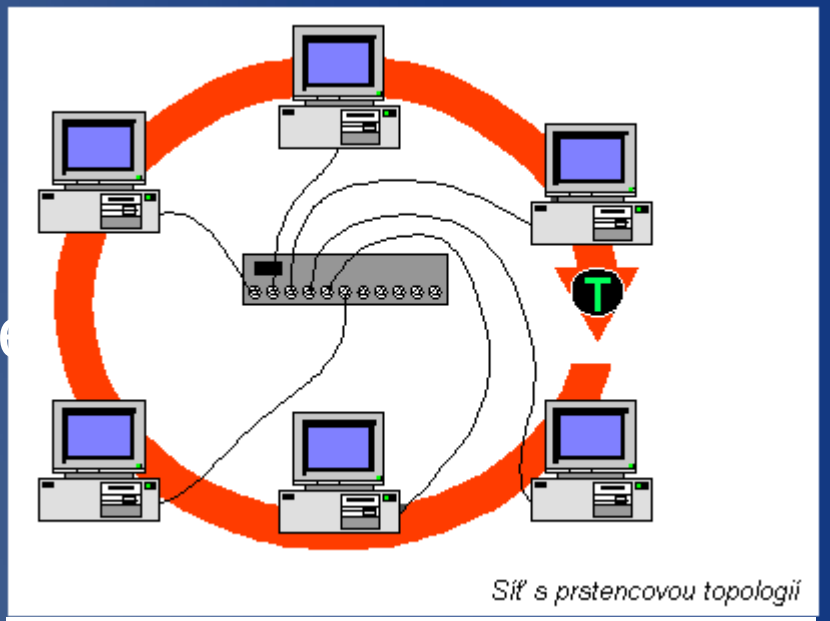
- Přepínání packetů
- Přepínání zpráv
- Přepínání okruhů

Řízení přístupu

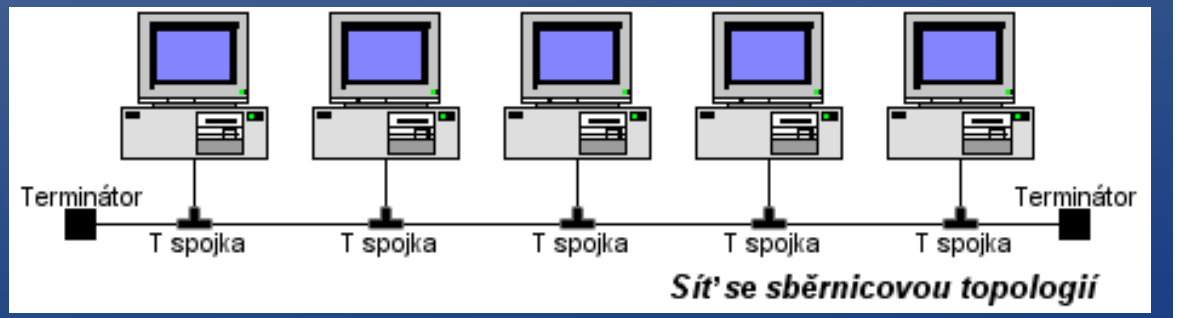
- Sdílené médium
  - Více bodový spoj vs. Dvoubodový spoj
- Kolize
  - Současné vysílání více uzlů
  - Signály nelze oddělit
  - Lze detekovat

LAN

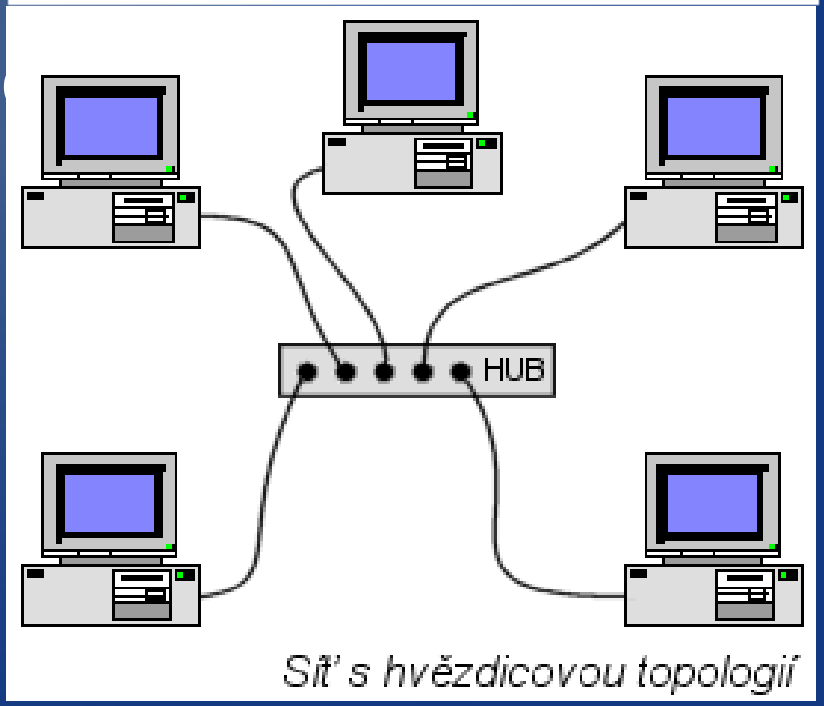
- Local Area Network
  - Typicky více bodové spoj
  - Sběrnice, hvězda, kruh
- Wide Area Network
  - Typicky dvoubodové spoj
  - Router - Router



Síť s prstencovou topologií



Síť se sběrnicovou topologií



Síť s hvězdicovou topologií

Mechanismus řešení

- Řeší linková vrstva
  - LLC
    - Logical Link Control
    - řízení rámců, dělení, kontrola, zabezpečení
  - MAC
    - Media Access Control
    - Implemtentuje způsob řízení přístupu k médiu

Řízení přenosu

- Centralizované
  - Arbitr (výzva / žádost)
- Decentralizované
  - Deterministické
    - Předávání pověření, rezervace, priority
  - Nedeterministické
    - Soutěž o právo vysílat

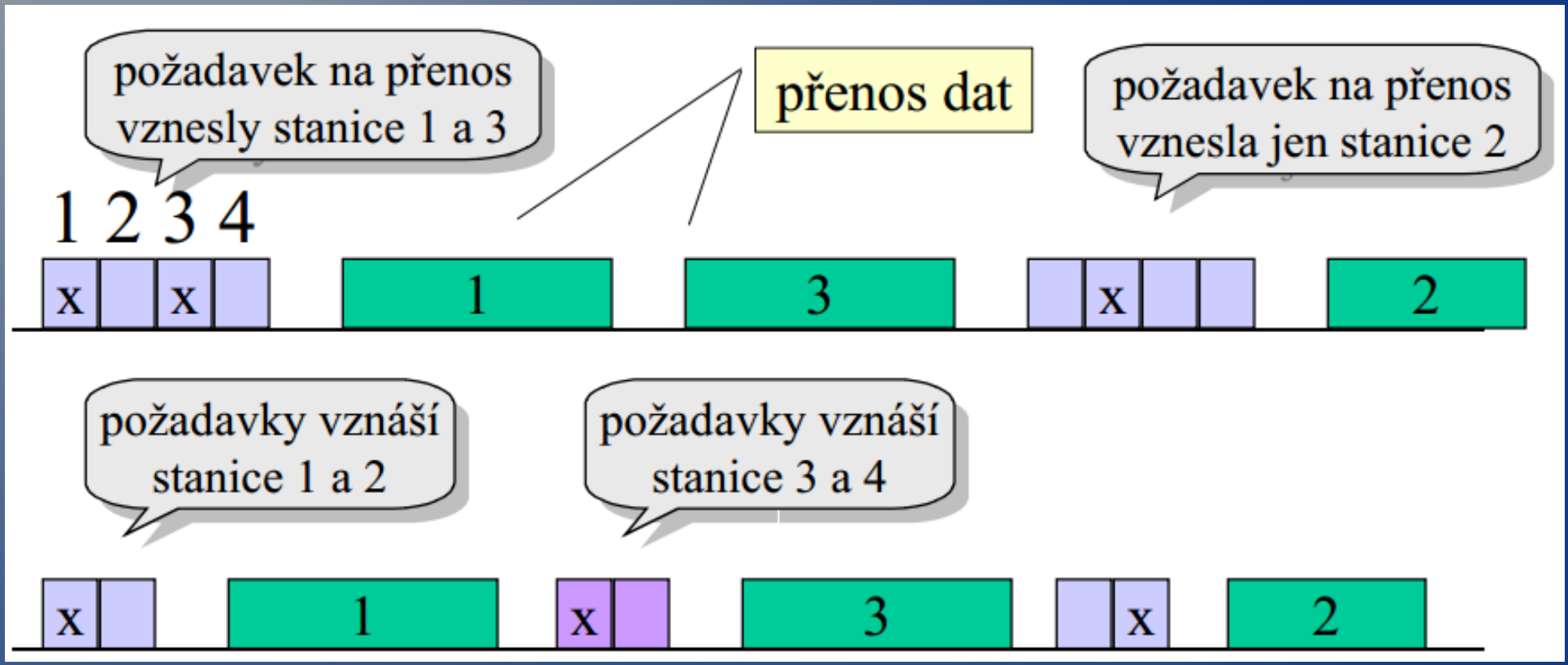
Centralizované

- Existuje arbitr
  - Výzva – chceš vysílat
  - Žádost – chci vysílat
- Změna arbitra / výpadek arbitra
- Řízená změna

Decentralizované

- Předávání pověření
  - Rezervační rámec / Token
  - TokenRing, TokenBus
- Soutěž o kanál
  - Ethernet (CSMA/CD)
  - Wifi (CSMA/CA)

Rezervační rámec



Předávání pověření

- Pověření - token
- Token je předáván mezi uzly
- Tvoří logický kruh
- Problém ztráty tokenu

Token Ring

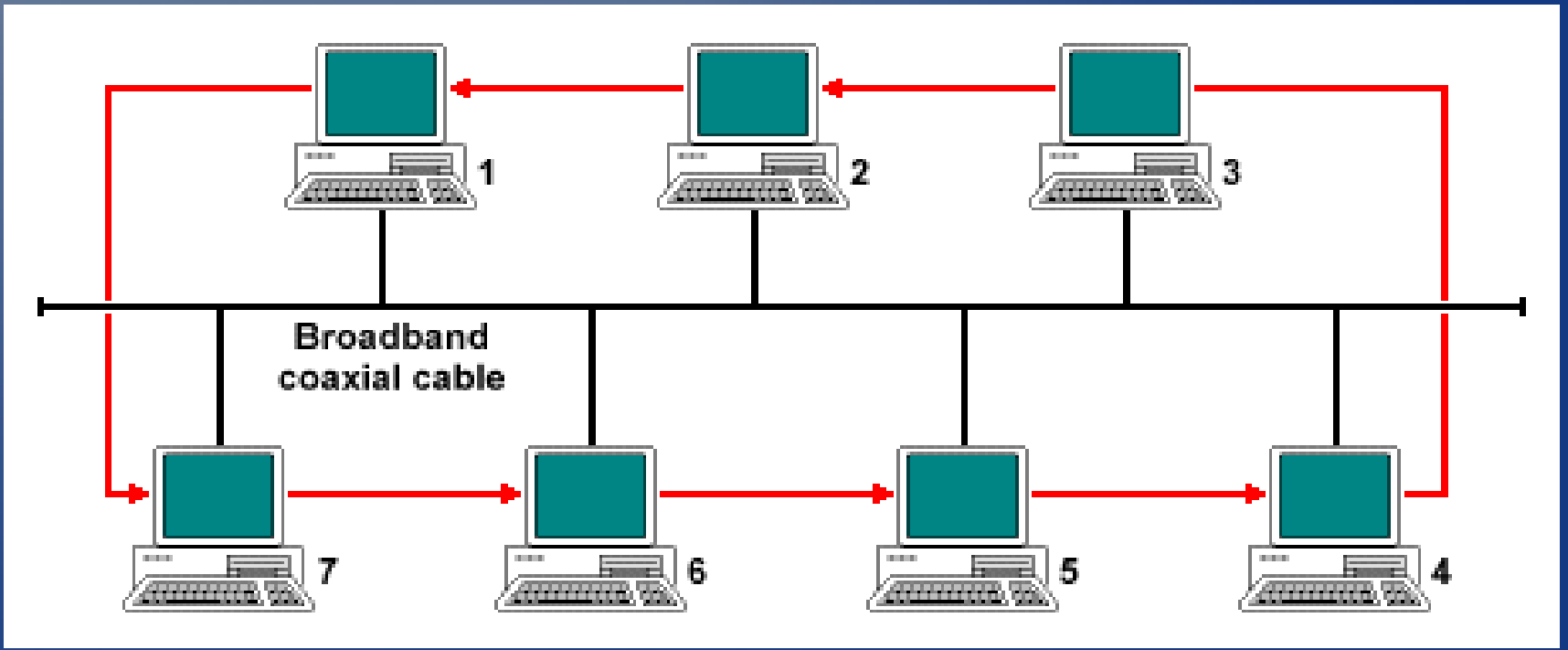
- Distribuovaná a řízená metoda
  - IBM Token ring – zapojení do hvězdy, kroucená dvojlinka, IEEE 802.5 nepředepisuje žádnou topologii ani medium
- Logický kruh
- Lepší při větším zatížení než Ethernet
- Diferenciální manchester
- Když nikdo nevysílá posílá se jen prázdný token
- Pokud nekoluje žádný token nebo je jich více, zasáhne vyčleněná stanice - aktivní monitor

Token Ring

- MAU
  - Media Access Unit
  - Multistation Access Unit, MSAU

Token Bus

- Využívá metody předávání pověření
- Sběrnicová topologie
- Kruh je pouze logický



CSMA

- Carrier Sence – detekuju nosnou vlnu, pokud je čekám
- Multiple Access – vysílá více uzlů, přijímají všichni
- Dochází ke kolizím, detekuje jen před začátkem vysílání
- Přenese se celý rámec, chybu musí odhalit příjemce
- Naléhající – čeká na konec hned vysílá
- Nenaléhající – přeplánuje se na později
- P-naléhající – s p% se chová jako naléhající

Detekce kolizí

- Typy detekcí
  - Předcházení CA (wifi)
  - Detekce kolizí CD (ethernet)
  - Bez detekce (Aloha)

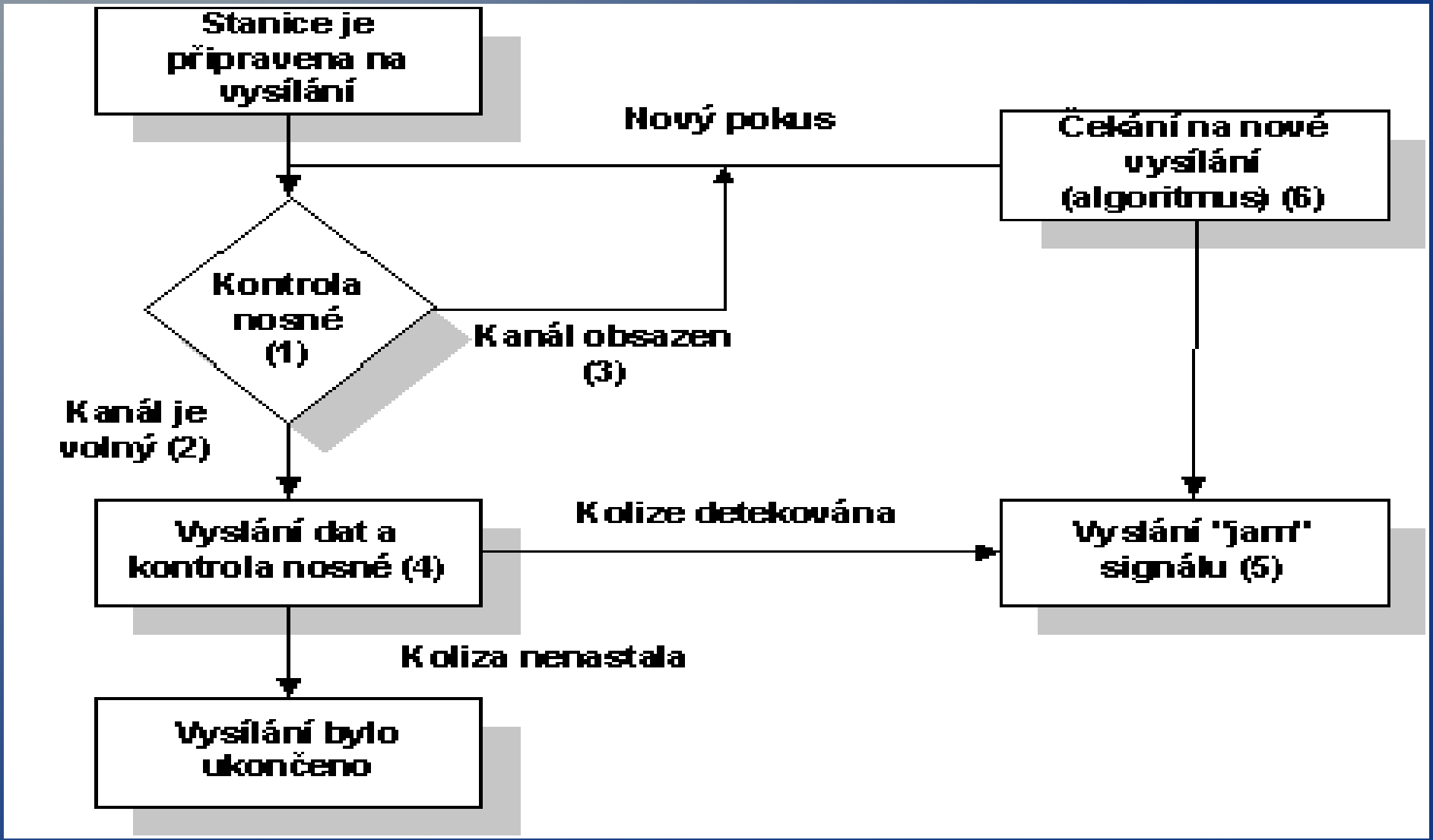
CSMA / CA

- CSMA/CA
- Předchází kolizím
- Každý uzel informuje ostatní o úmyslu vysílat
- Minimalizujeme kolize, ale můžou nastat
- Neumíme detekovat
- Využití v bezdrátových sítích, kde nelze provést současně vysílání i příjem

CSMA / CD

- Detekuje kolize a okamžitě zastavuje vysílání
- Náhodný interval čekání na další vysílání
- Při opakování dobou zdvojnásobuje
- Zároveň kontroluje zda je linka volná a pokud ano vysílá
- Během přenosu detekuje aktivitu ostatních
- Mnohem lepší využití media, neplýtvá se časem při odeslání celých rámců
- Nelze použít všude, potřebuje přídavnou elektroniku na detekci kolizí

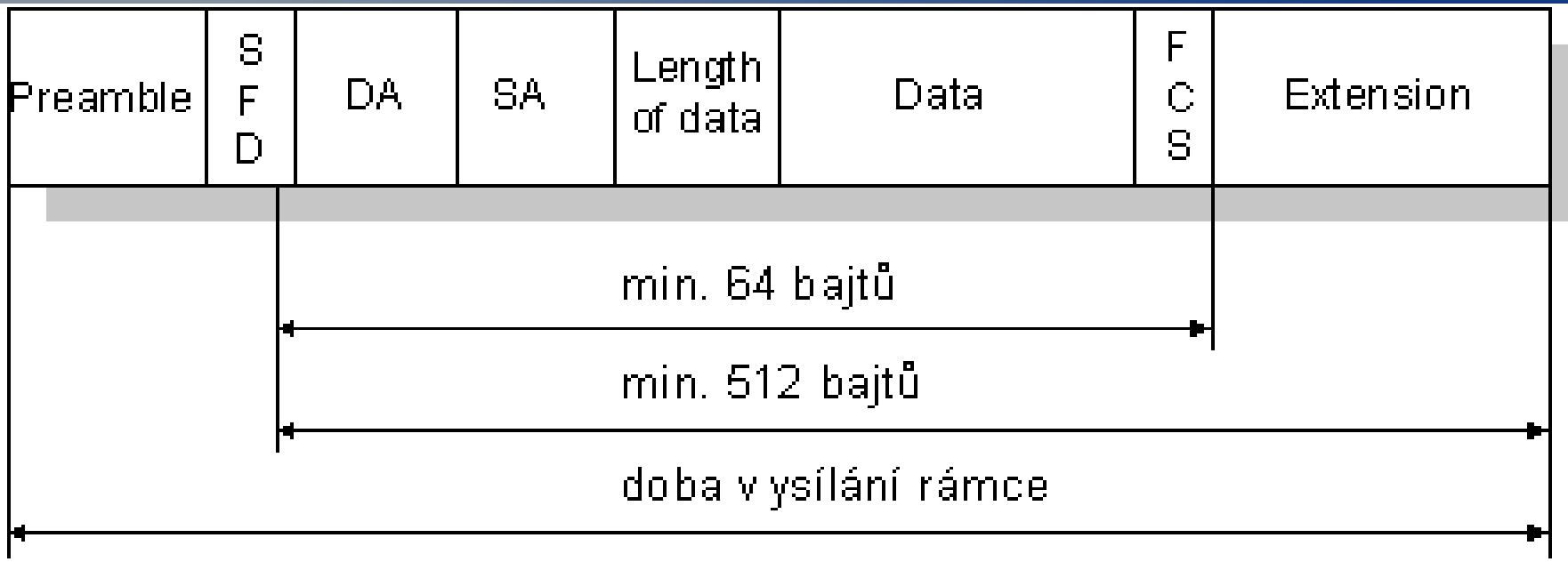
CSMA / CD


Stanice je připravena na vysílání
Nový pokus
Čekání na nové vysílání (algoritmus) (6)
Kontrola nosné (1)
Kanál obsazen (3)
Kanál je volný (2)
Kolize detekována
Vyslání dat a kontrola nosné (4)
Vyslání "jam" signálu (5)
Koliza nenastala
Vysílání bylo ukončeno

Ethernet

- Distribuovaná a neřízená metoda přístupu
- Využívá CSMA/CD
- Při detekci kolize se zašle JAM - 32 bitů a všichni se na chvíli odmlčí
- Čekání je náhodnou dobu, interval se při prvních deseti pokusech zdvojnásobuje
- Pokusů je celkem 16, pak se nahlásí chyba
- Velice efektivní při malém zatížení sítě
- Lepší pro delší rámce

Ethernet



SFD ... Start of Frame Delimiter
DA ..... Destination Address
SA ..... Source Address
FCS ... Frame Check Sequence

Ethernet

- Preambule – 8 bytů, střídá 0 a 1 a poslední 10101011 – SFD, slouží na synchronizaci
- Cílová a zdrojová adresa
- Typ protokolu
  - Ethernet II – typ vyššího protokolu
  - IEEE 802.3 – délka dat
- Datová 46B-1500B
- Datová výplň – doplněk na 64B
- Kontrolní součet, FCS, 32b CRC

UPS 2012/2013

Cvičení 11

Obsah

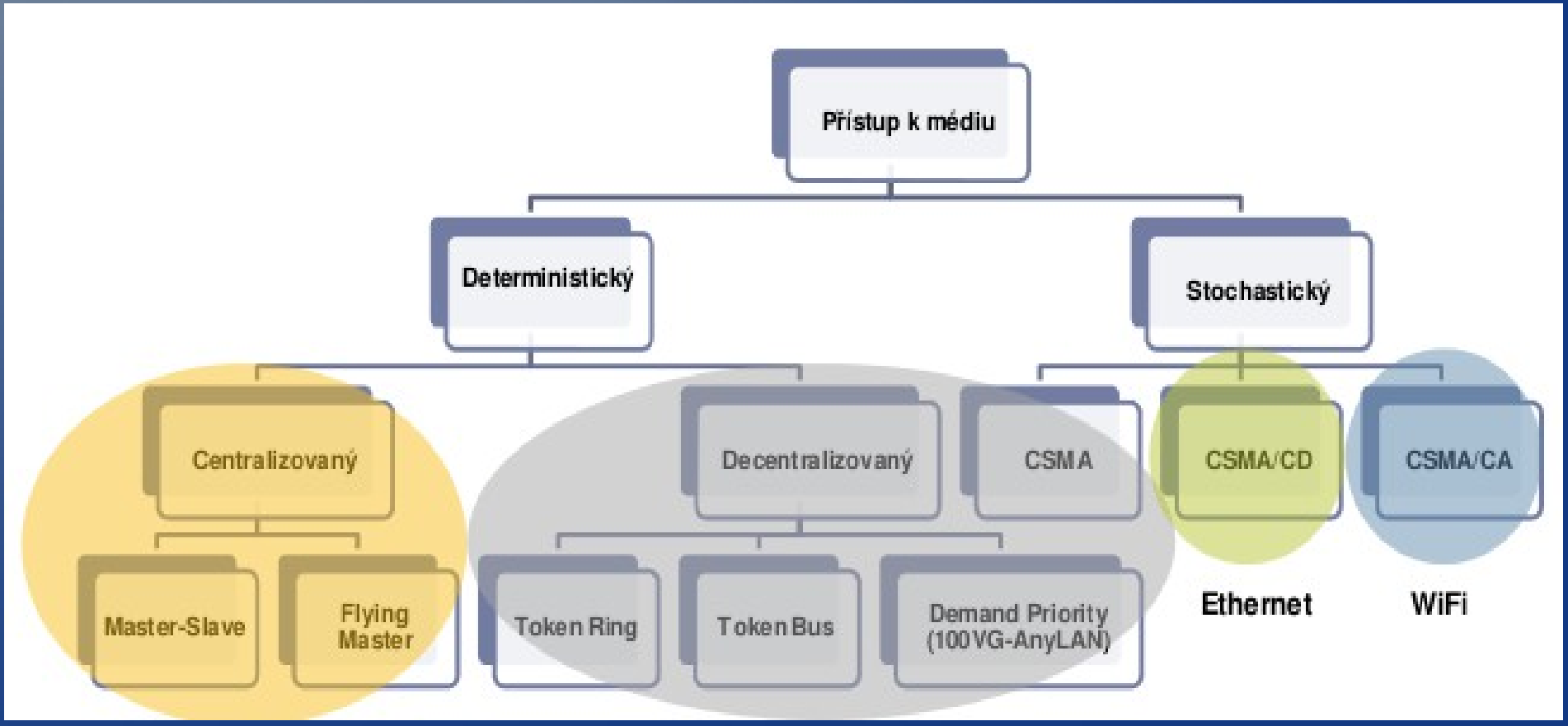
- Přístupové metody z minula
- Bridge
- STP
- Routing

Přístupové metody

- Aloha
- CSMA
- CSMA/CD
- CSMA/CA
- TokenRing
- TokenBus
- Centralizované

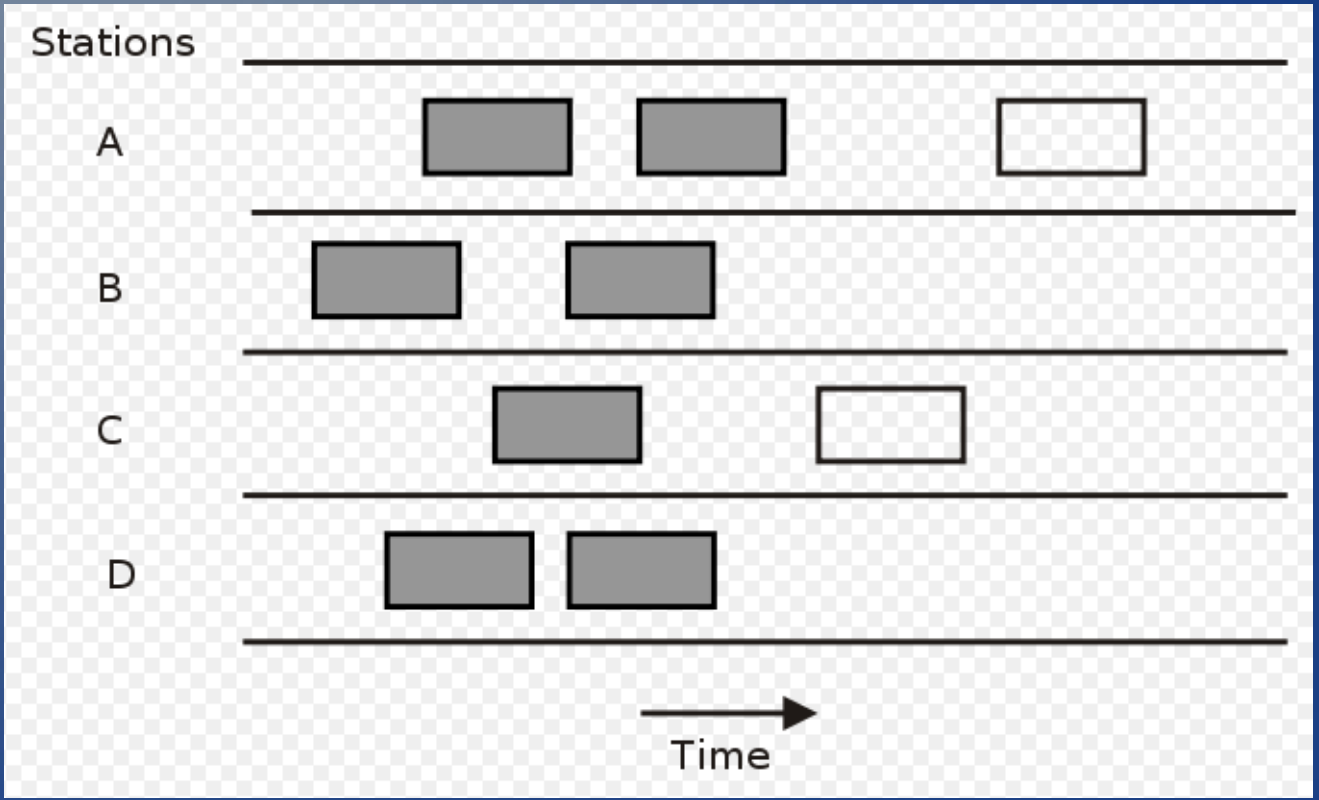
Přístupové metody

- Deterministický, nedeterministický
- Centralizovaný, decentralizovaný



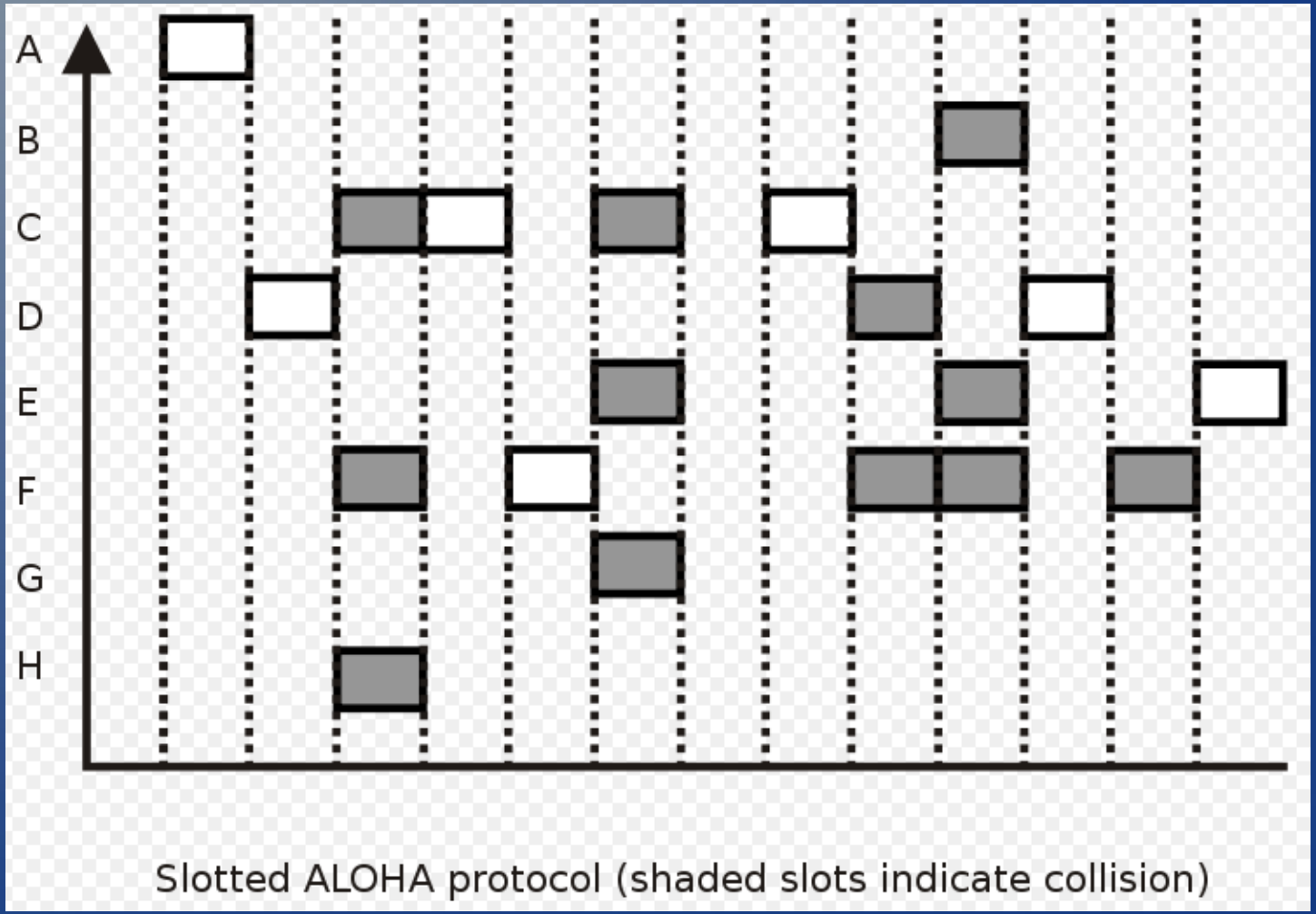
Aloha I.

- Vyšle rámec a čeká na doručení potvrzení



Aloha II.

- Čas rozdělen na sloty, ve kterých se vysílá



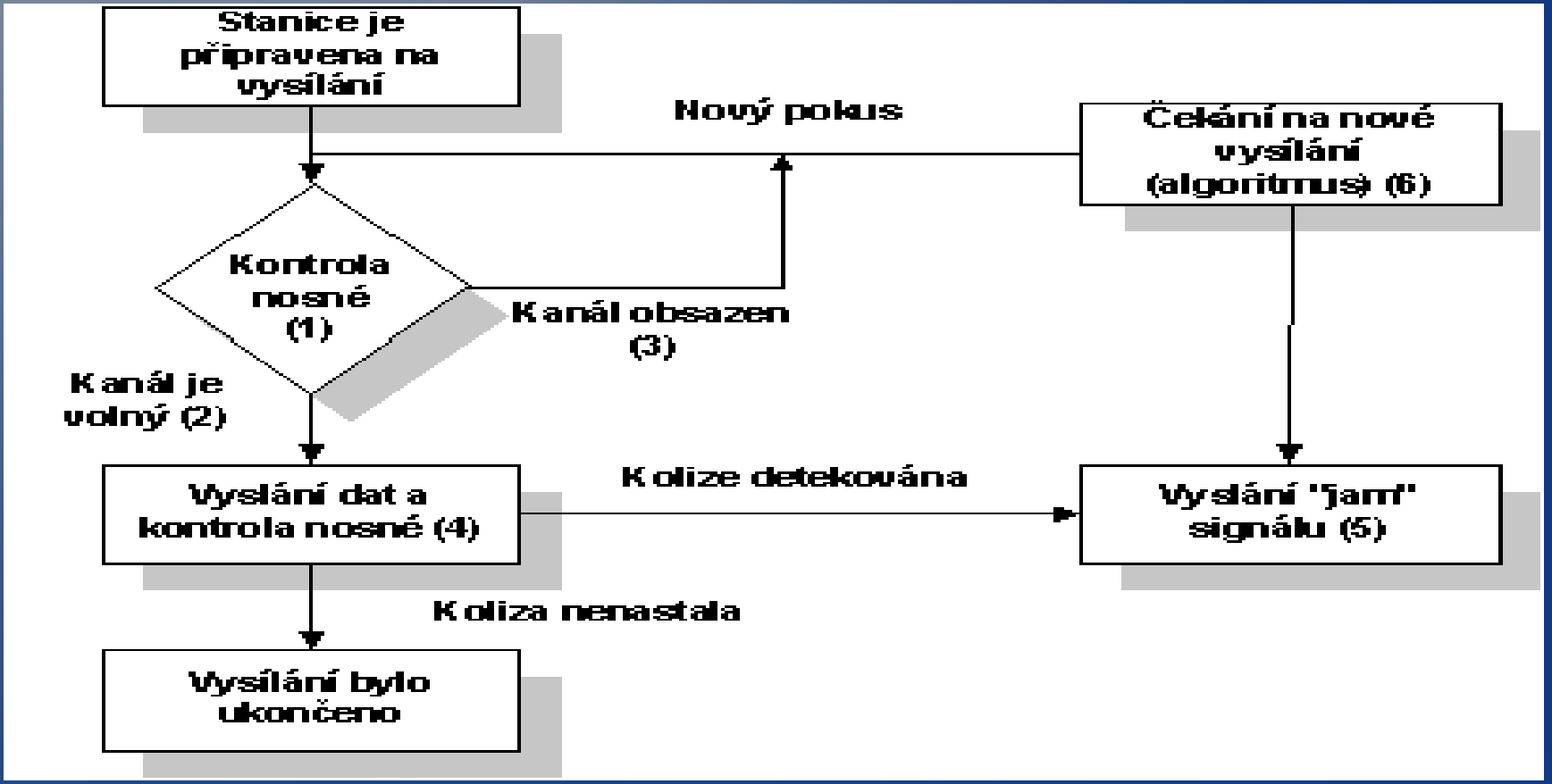
Slotted ALOHA protocol (shaded slots indicate collision)

CSMA

- Carrier Sence – detekuju nosnou vlnu, pokud je čekám
- Multiple Access – vysílá více uzlů, přijímají všichni
- Dochází ke kolizím, detekuje jen před začátkem vysílání
- Přenese se celý rámec, chybu musí odhalit příjemce
- Naléhající – čeká na konec hned vysílá
- Nenaléhající – přeplánuje se na později
- P-naléhající – s p% se chová jako naléhající
  - Ideální pro p 5-10%, využití až 95% kanálu

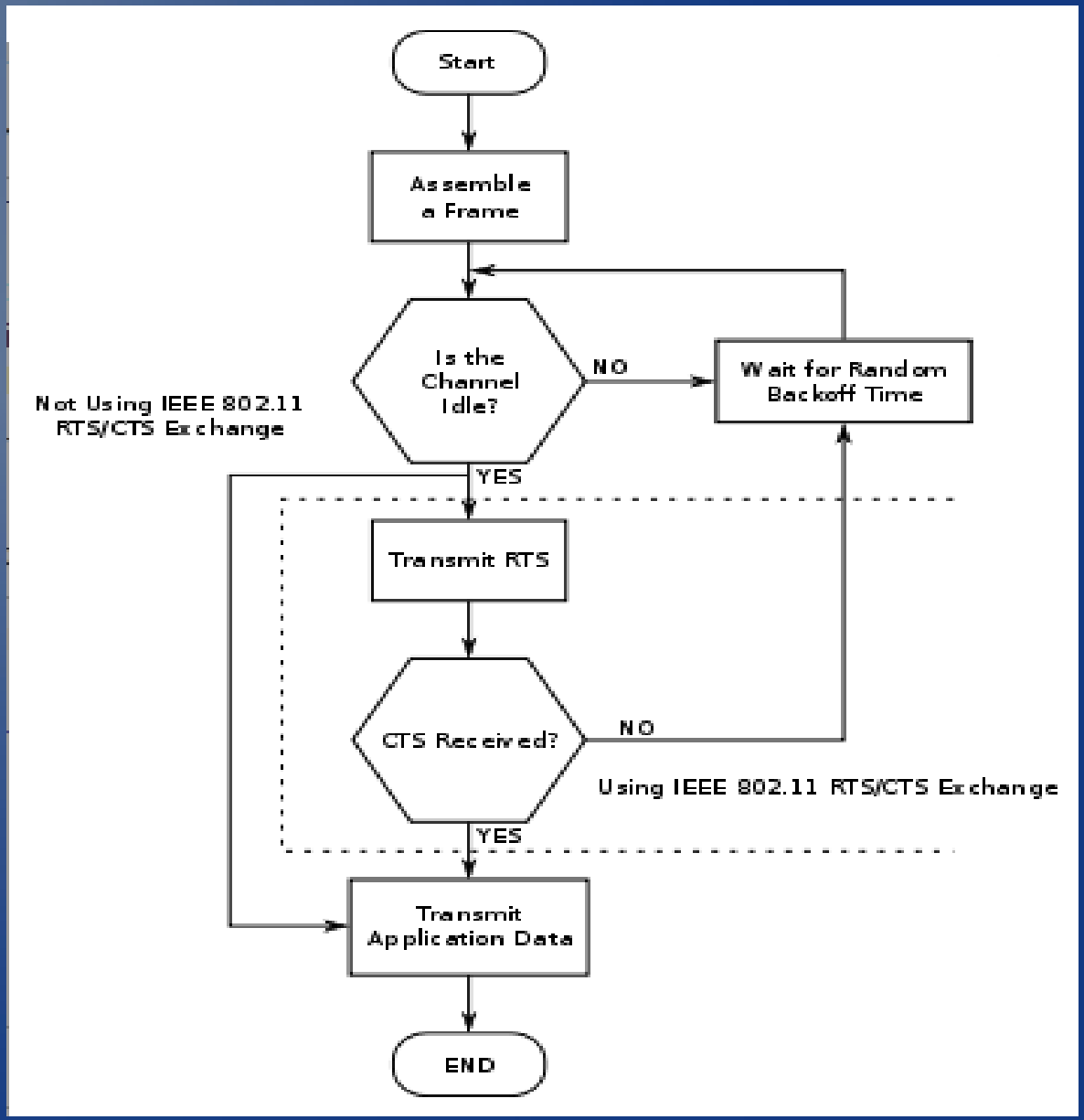
CSMA/CD

- Naslouchá na médiu, detekuje kolizi



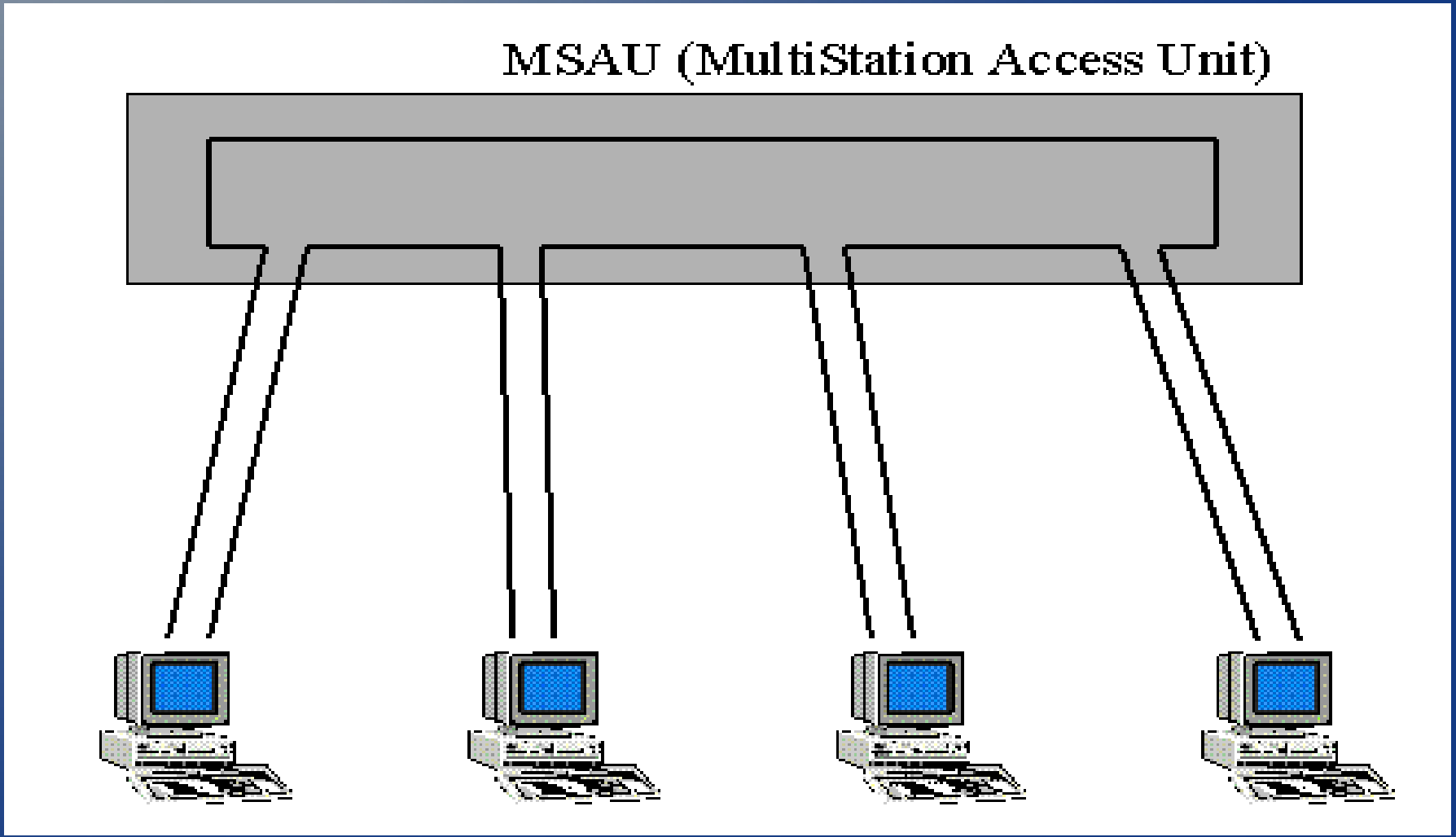
CSMA/CA

- Naslouchá na médiu, nedetekují kolizi, snaha jí předcházet



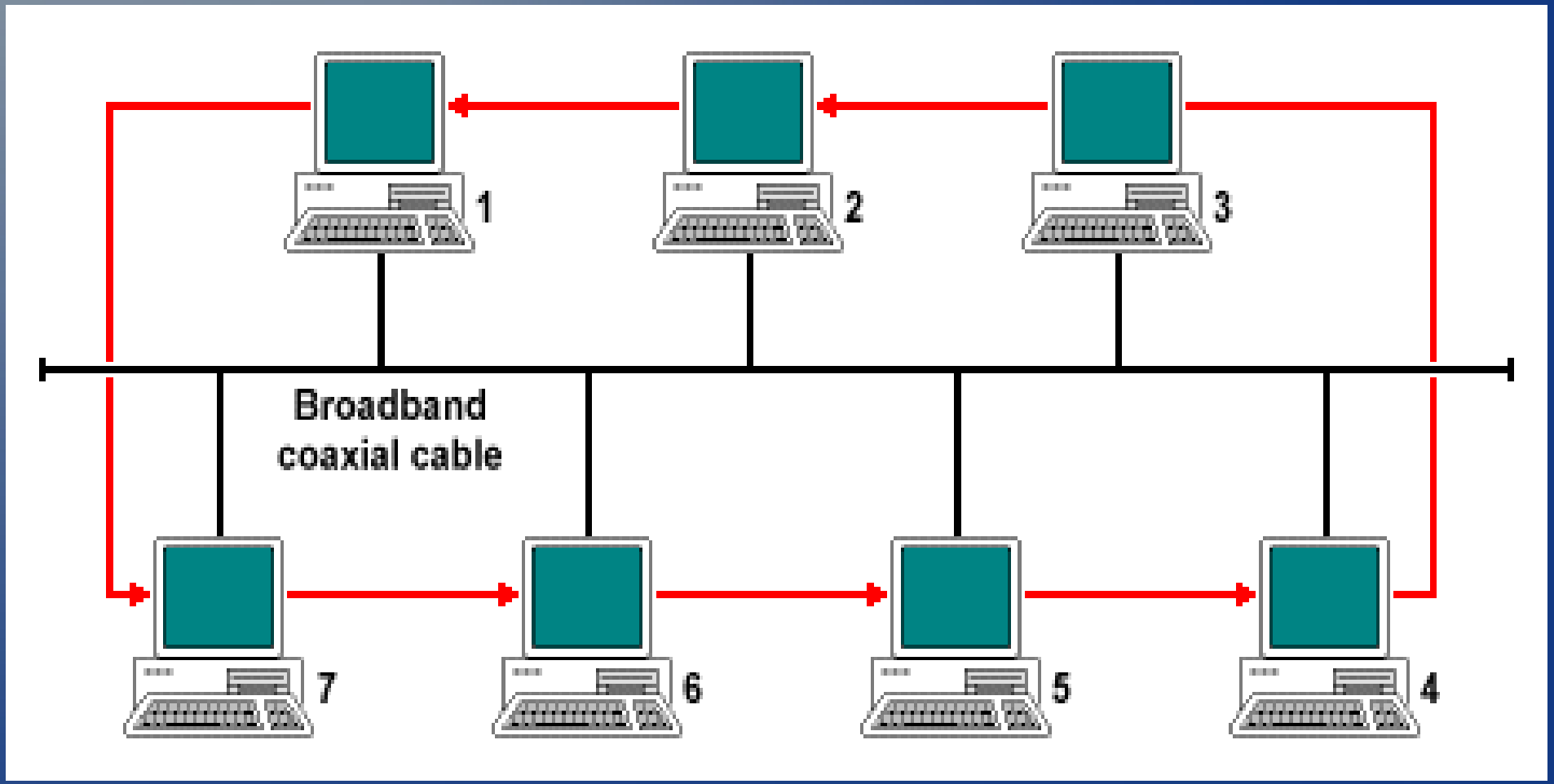
TokenRing

- Předávání pověření - tokenu



TokenBus

- Sběrnice, logikcý kruh, předávání tokenu



Transparentní mosty - Bridge

- Spojuje sítě na L2
- Transparent bridging
  - Neviditelný pro koncové stanice
  - Postupně se učí co kde leží
- Source route bridging
  - Pro propojení s token-ring
  - Packet musí obsahovat i cestu přes mosty
  - Je třeba znát cestu

Transparentní mosty - Bridge

- Výhody
  - Není potřeba konfigurovat
  - Snižuje velikost kolizní domény
  - Transparentní pro vyšší protokoly
  - Lacinější než router
- Nevýhody
  - Neomezuje všesměr
  - Vyšší latence – manipulace s MAC
  - Dražší než opakovače
  - Přemosťování různých MAC vede k chybám

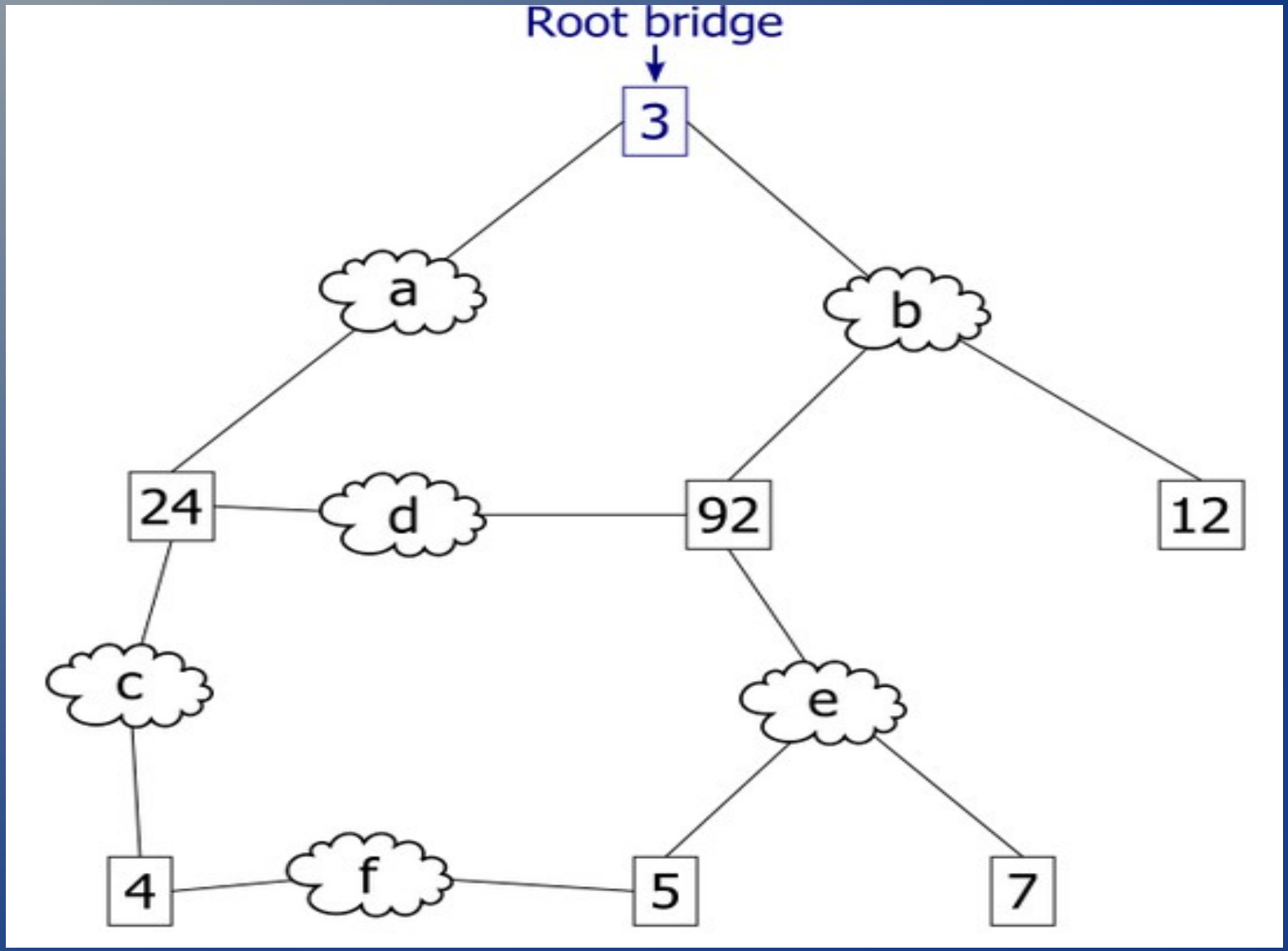
Spanning Tree - STP

- Mechanismus předcházení kruhu v síti
- Mechanismus failover linek
- Mechanizmus pro load balancing trunk portu v rámci VLAN
- Problém smyček
  - Broadcastové bouře
  - Problém s konektivitou
  - Násobné doručování zpráv
- Vychází z TGD
- Typický problém ve větších sítích
- Při výpadku portu dochází k přepočítávání

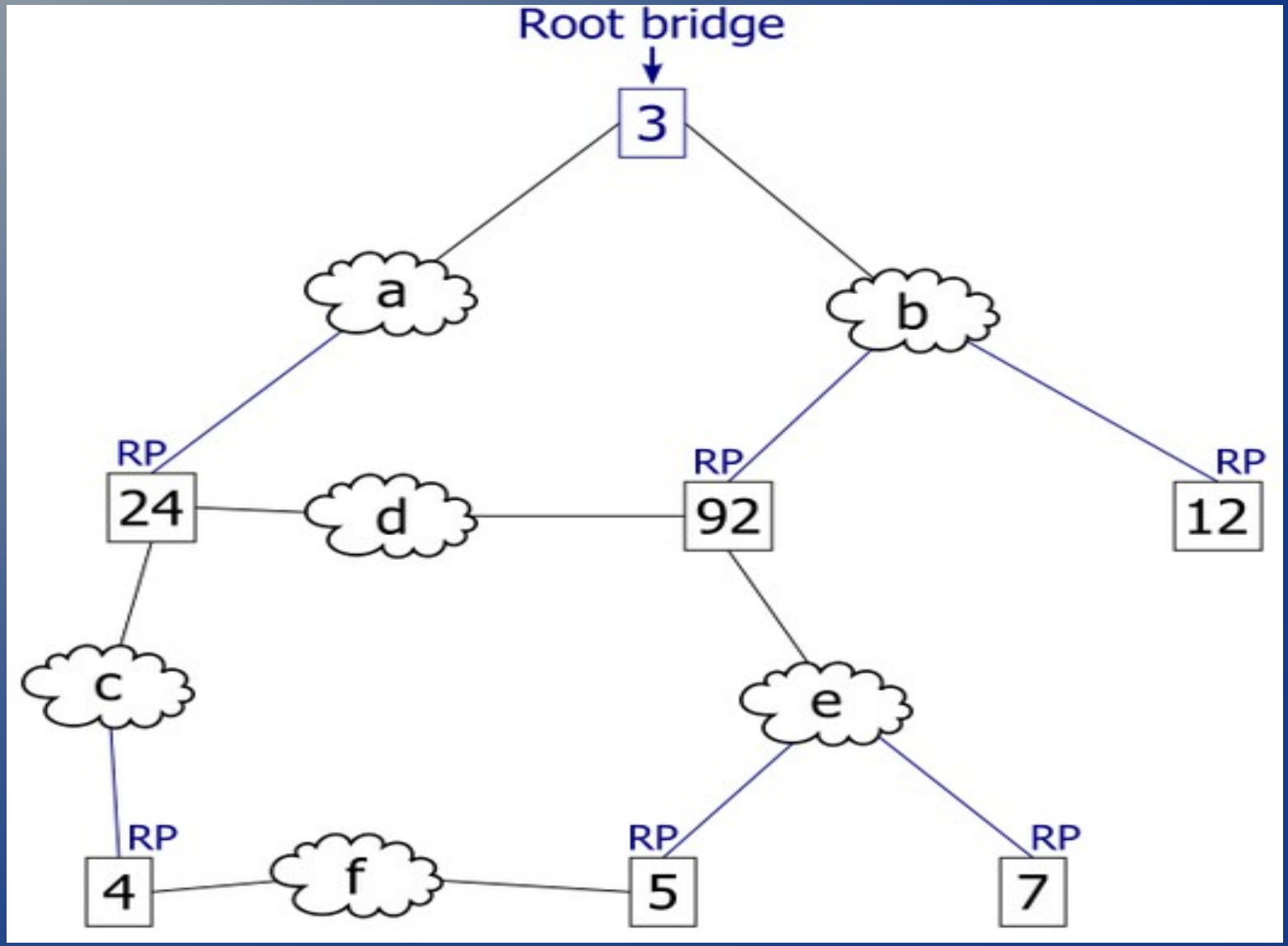
STP - algoritmus

- Volí se root bridge – podle ID z MAC
- Tvoříme strom podle ceny linek
  - Cena přiřazena
  - Rychlost jako cena – implicintí
  - Aktivní linky sou součástí stromu, ostatní blokované
- Cyklické posílání BPDU zpráv
  - Bridge Protocol Data Unit
  - Posílá root bridge
  - Všechny mosty kontrolují, že zprávu dostaly
- Po změně nastává přechodový stav
  - Porty nemusí být dostupné
- Ustálení po určité době podle varianty

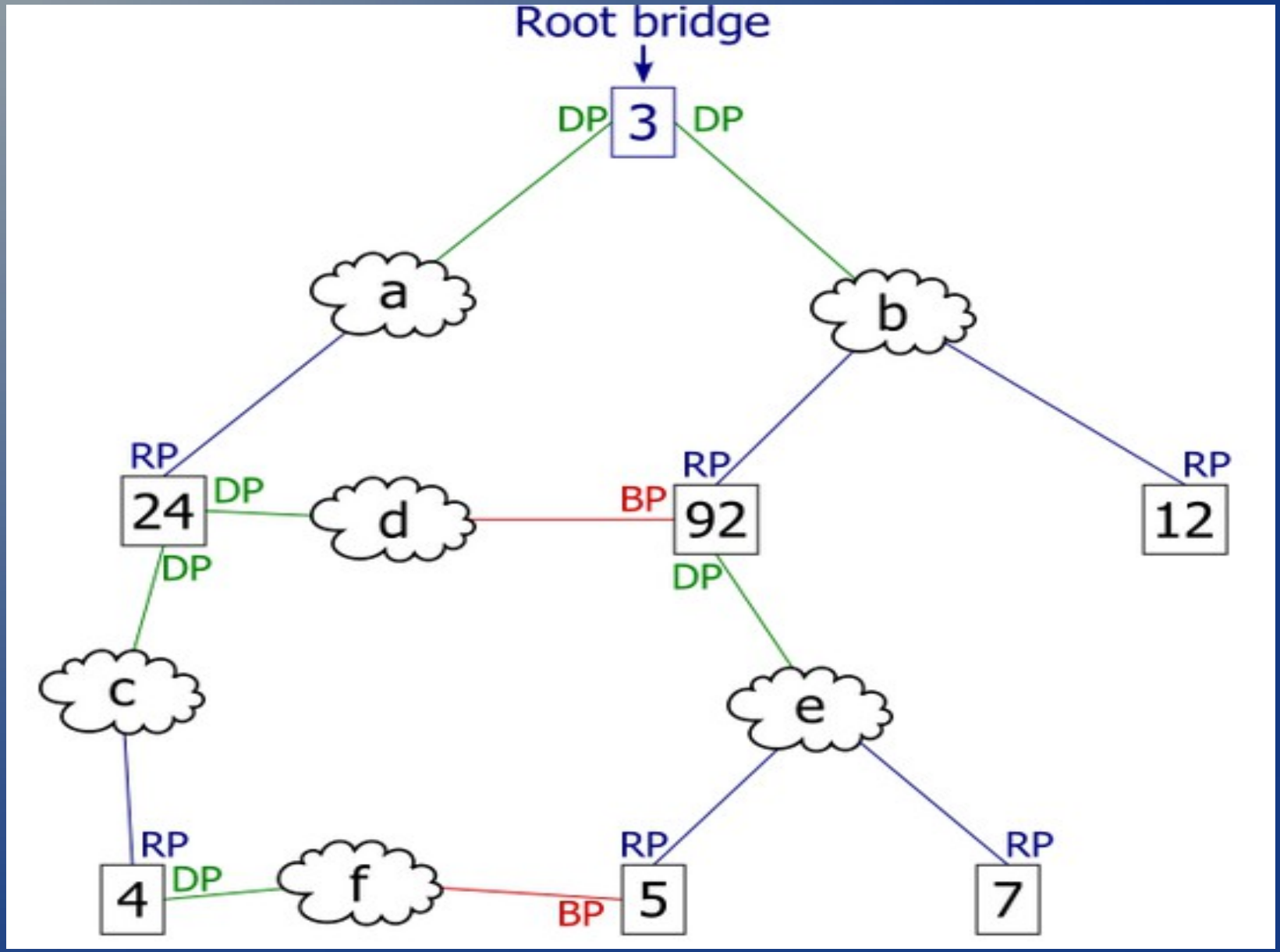
STP I.



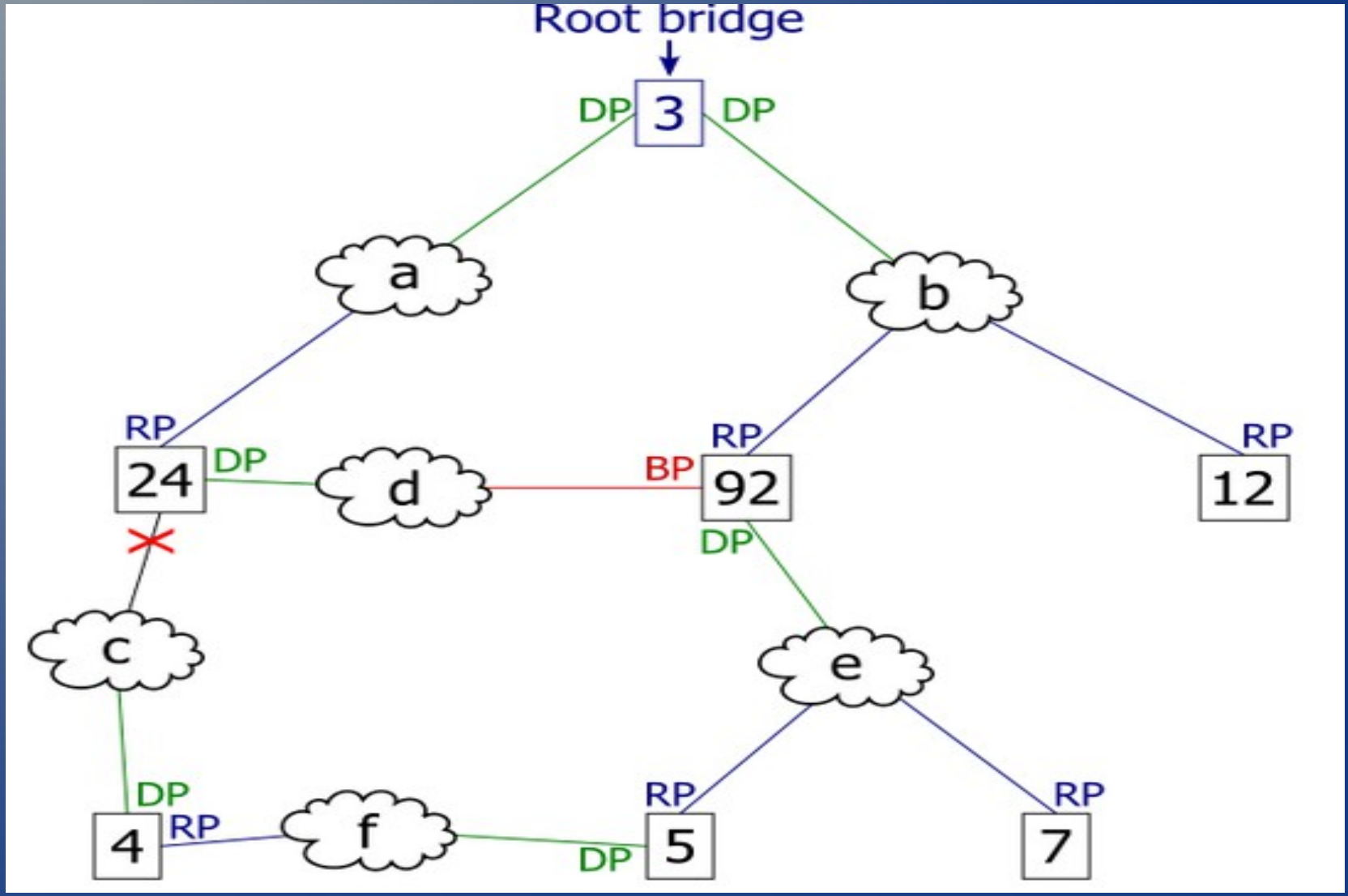
STP II.



STP III.


Root bridge
DP
3
DP
a
b
RP
24
DP
d
BP
RP
92
RP
12
DP
DP
c
e
RP
4
DP
f
BP
RP
5
RP
7

STP IV.


Root bridge
DP
3
DP
a
b
RP
24
DP
d
BP
92
RP
RP
12
DP
c
e
DP
4
RP
f
DP
RP
5
RP
7

STP – Porty

- Stav Blokující
  - Přijímá pouze BPDU, nevysílá
- Stav Naslouchací
  - Přijímá a vysílá pouze BPDU
- Stav Učící se
  - Přijímá a posílá BPDU a učí se MAC
- Stav Přeposílací
  - Přijímá a posílá vše

Směrování - Routování

- Požadavky
  - Jednoduchost
  - Stabilita, Robustnost
  - Optimalita
- Cílem je nalezení cesty
  - Nejkratší, Nejrychlejší
  - Nejlevnější
- Metrika – ohodnocení spojů
  - Počet skoků, rychlost, propustnost, konstanty

Routing - algoritmy

- Neadaptivní – statické
  - Cesta je určena předem
  - Nedochází k doplňování routovacích tabulek
  - Výpadek některých spojů může vést k rozpojení sítě
  - Jednoduché, nenízbytečný trafic
- Adaptivní – dynamické
  - Reagují na situaci v síti
  - Aktualizují routovací tabulky
  - Generují přenos na síti
    - Cena
    - firewall

Rouring x Forwarding

- Routing - určím co kam budu posílat
- Forward – přeposílám packety
- Centralizované
  - Jeden server rozhoduje o cestách a předává info dále
  - Forwarding provádějí koncové routery
  - Výpadek ochromí provoz sítě – nepoužívá se
- Distribuované
  - Každý provádí routing i forwarding
  - Vzájemná spolupráce uzlů – na výpočtu i předávání informací
  - Důležitá je rychlost

Rouring x Forwarding

- Isolované
  - Uzly nespolupracují při hledání optimální cesty
  - Méně efektivní
  - Provádí se routing i forwarding
  - Záplavové směrování
  - Metoda horké brambory
  - Náhodné routování
  - Zpětné učení
- Hierarchické
  - Rozdělení prostoru na menší části - area
  - Rouring v rámci oblasti se řeší samostatně – typicky ISP
  - Vymezený počet vstupů do oblasti

Záplavové směrování

- Pošlu data všude krom toho odkud přisla
- Pokud cesta existuje, najde se vždy
- Snadná realizace – nejsou routovací tabulky
- Nevýhody
  - Nadbytečné packety – TTL, nebo pamatování
- Použití
  - Běžný provoz vzácně – vojenské či speciální sítě
  - Aktualizace infromací, hledání cest
  - Distribuované služby

Zpětné učení

- Na začátku neznám nic a funguju záplavově
- Postupně se učím z dat co krz mě prochází
- Použití na linkové vrstvě u ethernetových mostů
- Nevhodné na větší sítě – pomalý náběh

Distance Vector Protocol

- Metrikou je vzdálenost
- Udržuji si info o vzdálenosti k ostatním uzlům
- Data se vyměňují jen mezi přímými sousedy
- Pro velké sítě velké objemy dat
- Nebere v potaz rychlosti linek
- Existuje limit kdy už je cesta prohlášena za nedostupnou
- Pomala konvergence
- RIP1, RIP2, RIPng, IGRP, EIGRP

Distance Vector Protocol

- RIP 1
  - Metrikou je vzdálenost - 16 je nekonečno
  - Každých 30s se rozesílá vektor vzdáleností
    - Pokud info přijde za 180s a více, spoj se bere jako mrtvý
  - Komunikuje na portu 520 UDP
  - Snadná konfigurace
  - Nepodporuje podsítě, funguje jen podle tříd IP
- RIP 2
  - Podporuje podsítě a zabezpečení
  - Port UDP 521
- RIPng
  - Pro IPv6

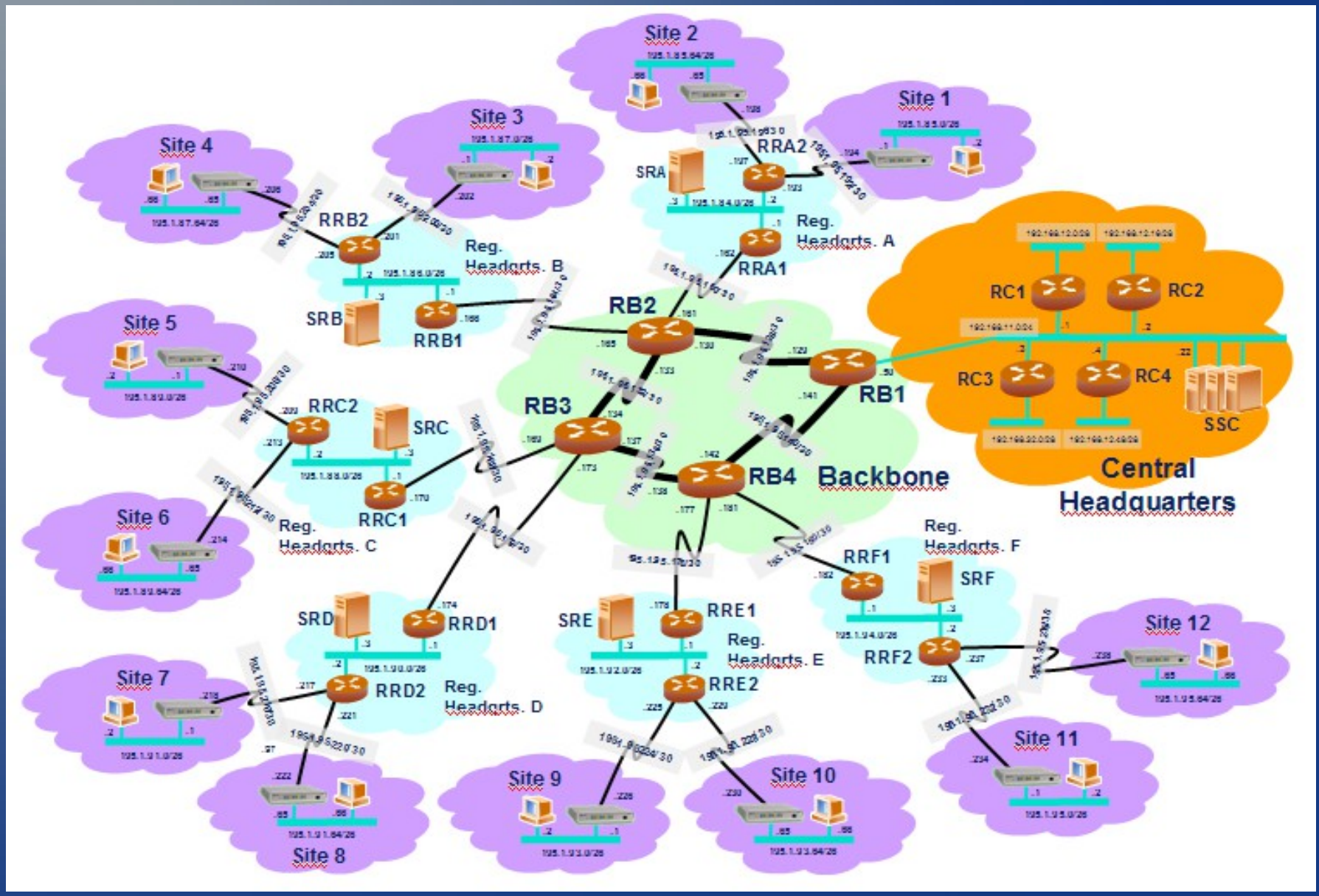
Link State Protokol

- Metrika je složená z více složek
  - Rychlost linky, cena, delay,.....
- Menší režie – data jen při změně
- Každý uzel má kompletní informaci o stavu sítě
- Každý uzel si počítá cesty sám – nepřenáší se chyba
- Vhodná pro velké sítě
  - OSPF, IS-IS

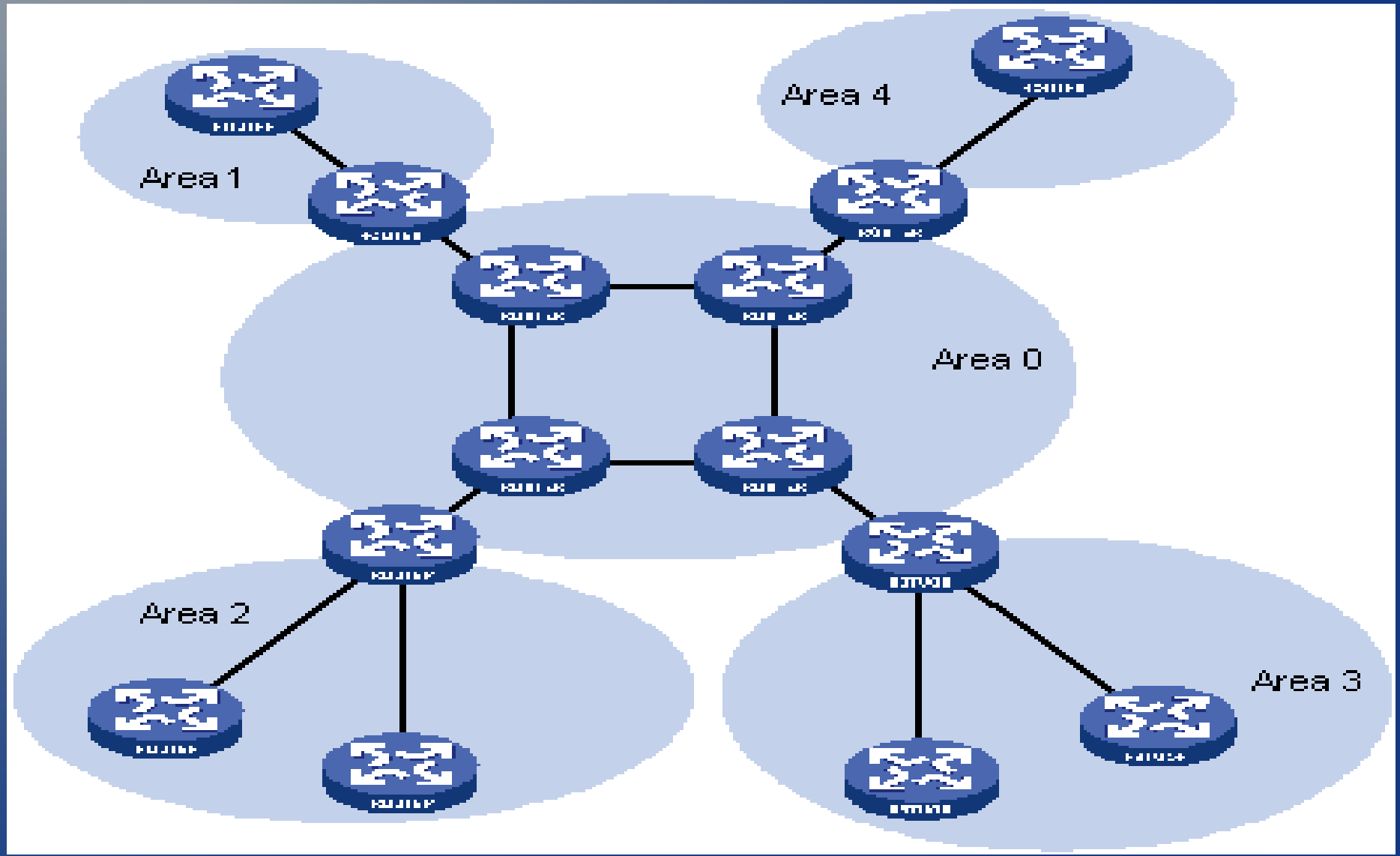
Link State Protokol

- OSPF
  - Nejprve se zjistí sousedi - HELLO
  - Zjišťují se odezvy sousedů – ECHO
  - Každý uzel pravidelně nebo při změně posílá data
  - Postupně se zjišťuje topologie
  - Základem je rychlost a cost
    - Základem je jen rychlost
    - Cost mužu měnit – problém s příchodem 10Gbps
  - Hierarchický model

Link State Protokol



Link State Protokol



Hierarchický model

- AS
  - Autonomní systémy providera - ISP
  - Routování pomocí EGP – externí rotovací protokoly - BGP
- BackBone Area
  - Páteřní síť v rámcí AS
  - Routování pomocí IGP – interní routovací protokoly - RIP, OSPF, EIGRP
- Area
  - Oblast v rámci AS připojená k backbone area

Více rourovacích protokolů

- AD - administrative distance
- Váha protokolu – který bude mít přednost

Protocol	Administrative distance
Directly connected	0
Static route	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
ODR	160
External EIGRP	170
Internal BGP	200
Unknown	255

UPS 2014/2015

Cvičení 12

Obsah

- Skupinové směrování
- Protokoly transportní vrstvy
- TCP
- UPD

Skupinové směrování

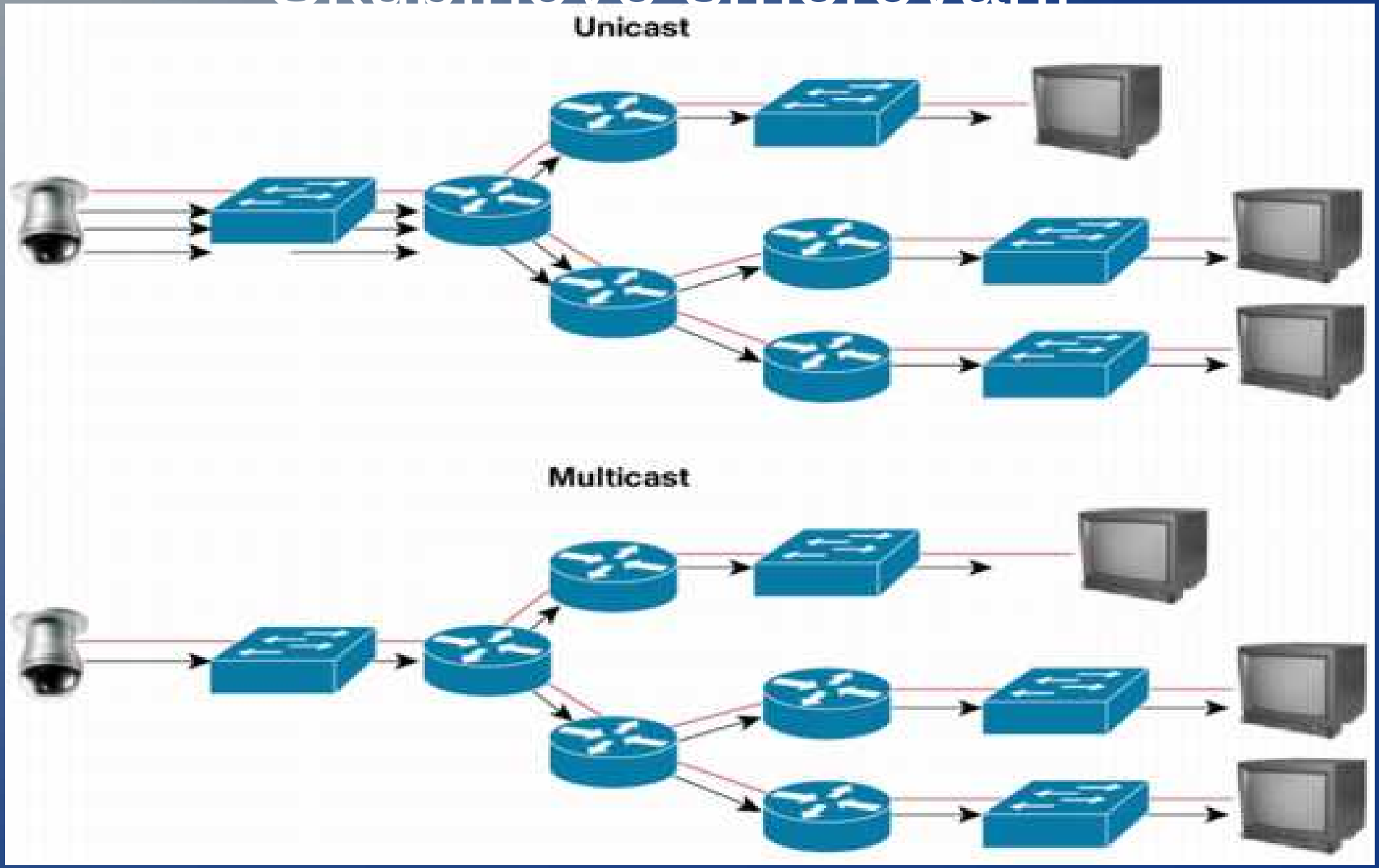
- Multicast
- Úspora datových toků
- Musí podporovat směrovače
- IP třídy D 224.0.0.0 – 239.255.255.255
  - Rezervované adresy (224.0.0.0 – 224.0.0.255)

    jedna LAN
  - Adresy s limitovaným rozsahem (239.0.0.0 – 239.255.255.255)
  - Veřejné adresy - jsou ostatní multicastové adresy

Skupinové směrování

- IGMP
  - Registrace uzlů na routerech
- Směrování multicastu – vytváření cest
  - MOSPF
    - Vypočítává cestu od zdroje ke každému členu skupiny
    - Cesta je uložena až do změny topologie
    - Join zprávy
  - PIM
    - Hustý provoz – reverzní záplavová cesta
    - Řídký provoz – mnoho dat, ale málo LAN
      - rendevouz point, Join
  - DVRMP
    - Reverzní záplavová cesta

Skupinové směrování



Protokoly transportní vrstvy

- Zajištění kvalitnějších služeb než nabízí síťová vrstva
  - Transparentní spolehlivý přenos
  - Řízení toku dat – vyrovnává nestabilitu sítí
  - Převod transportních adres na síťové
  - Nestará se o směrování
- Identifikací je port a protokol
  - TCP/53 a UPD/53 je něco jiného
- Nejběžnější TCP a UDP

Protokoly transportní vrstvy

- AEP, AppleTalk odráží protokol
- ATP, AppleTalk transakční protokol
- CUDP, cyklický UDP
- DCCP, Datagram ucpání řídící protokol
- FCP, vláknový kanálový protokol
- FCIP, kanál vlákna přes TCP/IP
- IL, IL protokol
- **iSCSI, internetové/síťové SCSI**

Protokoly transportní vrstvy

- NBP
- NetBEUI, NetBIOS Windows sdílení a další
- SPX, Sequenced paketová výměna
- RTMP, oponovat stolnímu servisnímu protokolu
- SCTP, rozdělit kontrolní přenosový protokol
- SCSI, malé počítačové systémové rozhraní
- **SSL, zabezpečit vrstvu zásuvky**
- **TCP, přenosový řídící protokol**
- **TLS, bezpečnost transportní vrstvy**
- **UDP, uživatelský Datagram protokol**

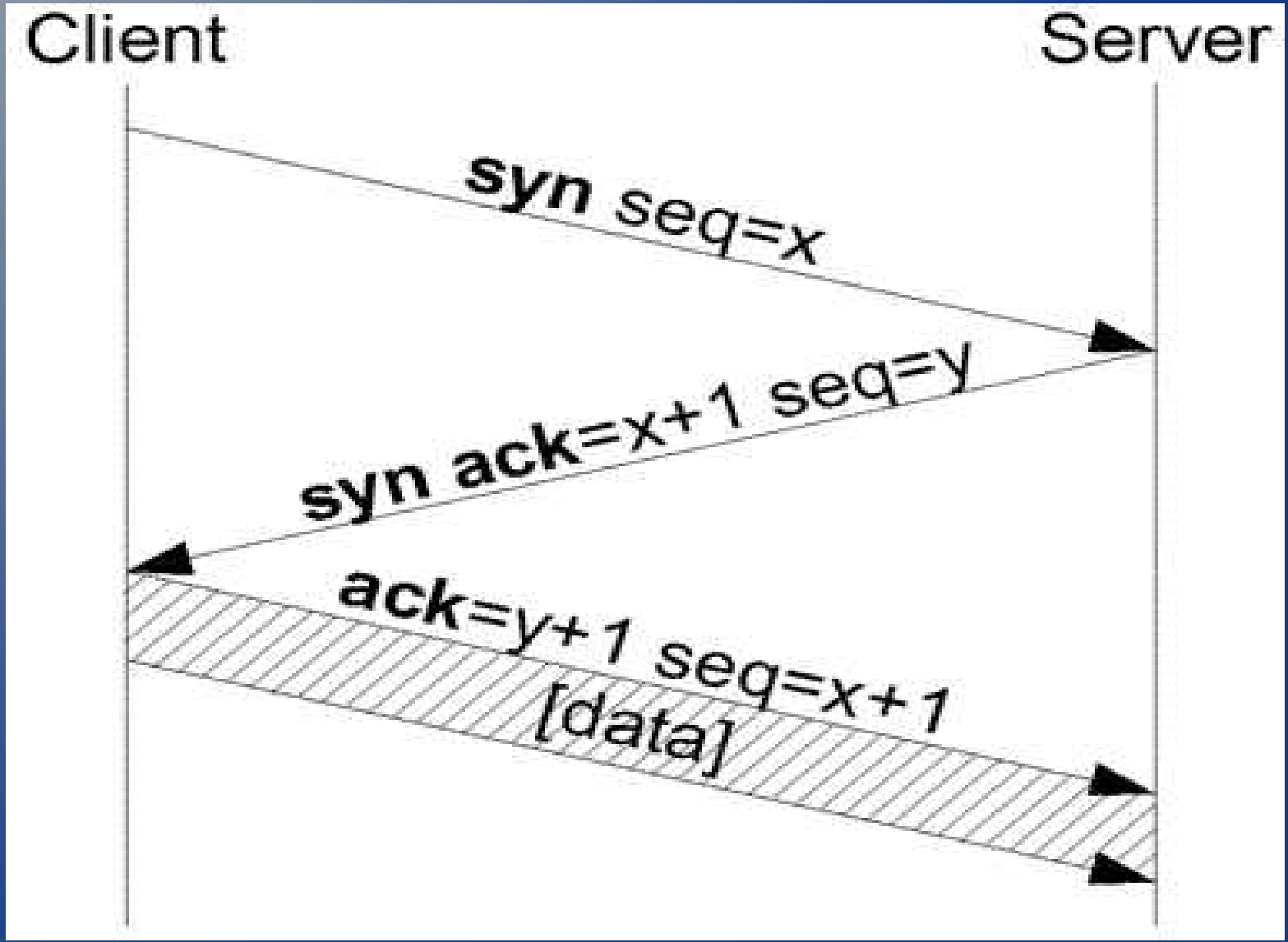
TCP

- Spolehlivé navázané spojení
- Potvrzování přenosu
- Znovu poslání při chybě
- Řazení packetů
- Multiplex pro koncové uzly – procesy
- Řízení toku dat podle parametrů sítě, klouzavé okénko

TCP

- Navázání spojení
  - Procedura „Three Way Handshake“
    - SYN, SYN/ACK, ACK
    - Nastavení sekvenčních čísel
- Ukončení spojení
  - Zašle se a potvrdí FIN

TCP



UDP

- Rychlý, ale nezabezpečený přenos packetů
- Bez potvrzení příjmu a opakování přenosu
- Typicky pro přenos video/zvuku
- Pro přenos souboru TFTP, OpenAFS !

Závěr

Toť vše a budem se těšit na PSI / SPOS ;)