

# Řízení datového spoje

# O čem přednáška je?

2

- Dosavadní přednášky se zabývaly
  - ✓ obecnými principy činností a zásadami výstavby systémů přenosu dat (architektury)
  - ✓ vlastnostmi fyzické vrstvy přenosového systému (signály a data, přenosová média, kódování, synchronizace, detekce/oprava chyb, ...)
- Zaměření této přednášky
  - ✓ Děje a mechanismy implementované na úrovni vrstvy datového spoje
    - synchronizace na úrovni rámců
    - řízení toku
    - chybové řízení
    - správa spoje: adresování, ustanovení, rušení, rozlišování dat z vyšších vrstev a protokolárních informací

# Proč nestačí služby fyzické vrstvy

3

- Služby fyzické vrstvy
  - a. **zajišťují** přenos signálových prvků přenosovým médiem (ze zdroje do cíle), do kterých jsou zakódované bity dat
  - b. **neumí** rozlišovat zda přenášejí data nebo řídicí informaci
  - c. **nezajišťují** opakování chybně přenesené informace
  - d. **nepodporují** ovládání toku informace ze zdroje do média
  - e. **nepodporují** určení entity mající právo vysílat do média
  - f. **nepodporují** komunikaci mezi nesousedními partnery na přenosovém médiu přenosového média
- Co nezajišťují, neumí nebo nepodporují služby (nejnižší) fyzické vrstvy, musí zajišťovat, umět, podporovat vrstvy vyšší

# Co řeší vrstva datového spoje?

4

- ❑ Logiku přenosu dat mezi partnery přímo (nezprostředkovaně jinou entitou) propojených přenosovým médiem.
- ❑ Pomocí signálů šířených fyzickou vrstvou.
- ❑ Logikou přenosu dat se rozumí
  - ✓ Synchronizace rámců – bloků dat tvořících PDU datového spoje
  - ✓ Řízení toku dat mezi komunikujícími partnery
  - ✓ Chybové řízení
  - ✓ Adresování na stanic na přenosovém médiu
  - ✓ Rozlišování řídicích informací a dat
  - ✓ Správa datového spoje – ustanovení, udržování, rušení
- ❑ Logiku přenosu dat mezi nesousedními partnery a zajištění požadované úrovně kvality takového přenosu dat řeší vrstvy vyšší

# Logikou přenosu dat se rozumí ...

5

- Synchronizace rámců – PDU datového spoje
  - ✓ rozpoznávání začátků a konců rámců
- Řízení toku dat mezi komunikujícími partnery
  - ✓ vysílač by neměl vysílat větší objemy dat, než je přijímač schopen zpracovat
- Chybové řízení
  - ✓ opakování přenosu chybně přenesených rámců
- Adresování na stanic na přenosovém médiu
  - ✓ identifikace stanic na přenosovém médiu
- Rozlišování řídicích informací a dat v komunikačním kanálu
- Správa datového spoje
  - ✓ ustanovení, udržování, rušení datového spoje

# Reprezentanti protokolů DLC typu

6

- **HDLC**, *High-Level Data Link Control*
  - ✓ základ téměř všech ostatních DLC protokolů
  - ✓ **PPP**, *Point to Point Protocol* (podpora rodiny protokolů TCP/IP)
  - ✓ **LAPB**, *Link Access Procedure, Balanced* (z rodiny protokolů X.25)
  - ✓ **LAPD**, *Link Access Procedure, D-Channel* (z rodiny protokolů ISDN)
  - ✓ **LAPF**, *Link Access Procedure for Frame-Mode Bearer Services* (z rodiny protokolů Frame Relay)
  - ✓ **LLC**, *Logical Link Control* (definovaný normami LAN IEEE 802)
  - ✓ ...
- **ATM Layer Protocol**,  
(z rodiny protokolů **ATM**, *Asynchronous Transfer Mode*)

# Řízení toku dat

7

- vysílač by neměl vysílat větší objemy dat, než je přijímač schopen zpracovat, nemá-li docházet ke ztrátám dat
  - ✓ jedná se o problém kapacit vyrovnávacích pamětí přijímače
  - ✓ rámce jsou vysílány sekvenčně
  - ✓ rámec se vysílá po dobu **TT**, *Transmission Time*
  - ✓ každý bit rámce se šíří médiem od vysílače k přijímači po dobu **PT**, *Propagation Time*
- Protokoly pro řízení toku dat
  - ✓ **Stop and Wait**
  - ✓ **Sliding Windows**

# Protokol řízení toku dat, Stop-and-Wait

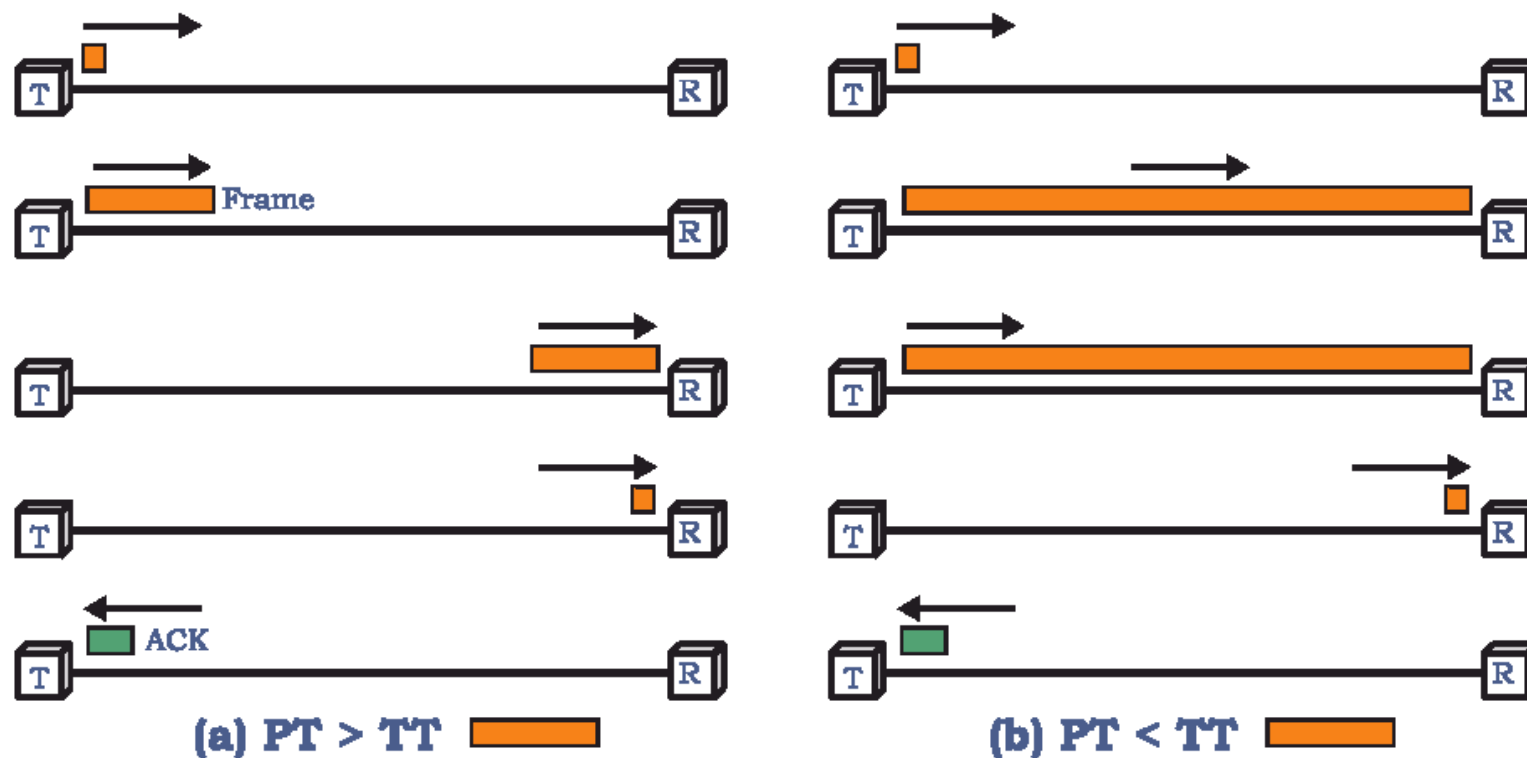
8

- Vysílač smí vyslat další rámeček pouze když přijme od přijímače potvrzení práva vyslat rámeček (**ACK**, *Acknowledgment*)
- Aplikace řízení toku dat typu Stop-and-Wait na vysílání „dlouhých“ zpráv nemusí být efektivní
  - ✓ Pokud  $PT > TT$   
(případ vysokých rychlostí přenosu dat a velkých vzdáleností),  
pak valná část spoje po dlouhou dobu neobsahuje žádné signálové prvky
  - ✓ I v případě  $PT < TT$  pro rámce,  
pro zpětné vysílání ACK opět vesměs platí  $PT > TT$



# Využití spoje, Stop-and-Wait

9



Protokoly typu *Stop-and-Wait* nejsou efektivní při současných rychlostech vysílání (i Gb/s až Tb/s) a délkách spojů (např. satelity, 36 000 km)

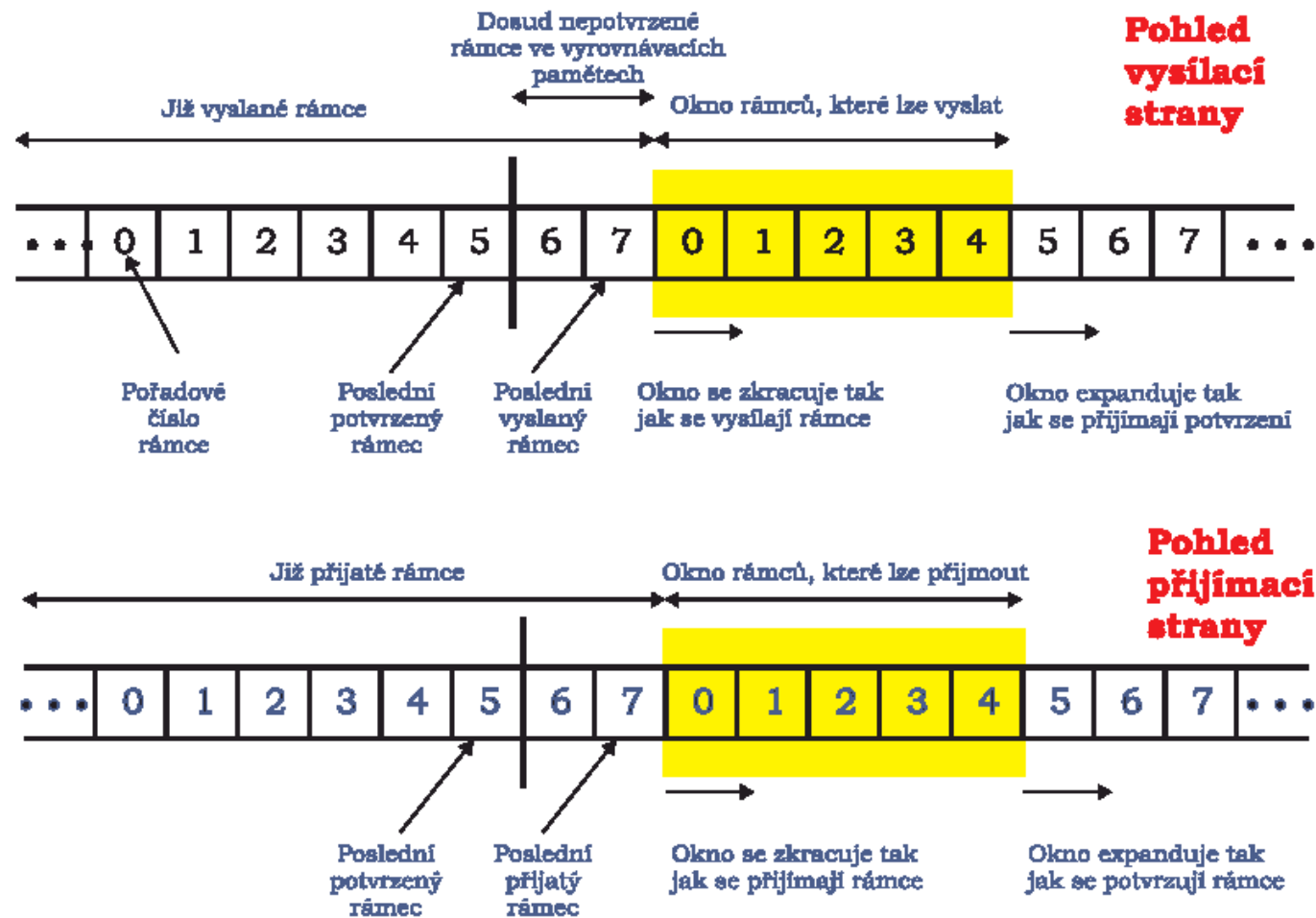
# Protokol řízení toku dat, Sliding Window

10

- Vysílač může vyslat až  $W$  rámců bez čekání na jejich individuální potvrzování
- Přijímač má vyrovnávací paměť na  $W$  rámců
- Po vyslání  $W$  rámců bez čekání na jejich individuální potvrzování smí vysílač vyslat další  $W$ -tici rámců až získá od přijímače ACK
- Každý rámeček je číslovaný
- ACK obsahuje číslo příštího očekávaného rámce
- Pořadová čísla jsou omezena bitovou šířkou pole čísla v rámci
  - ✓ Rámce jsou číslovány *modulo*  $2^k$

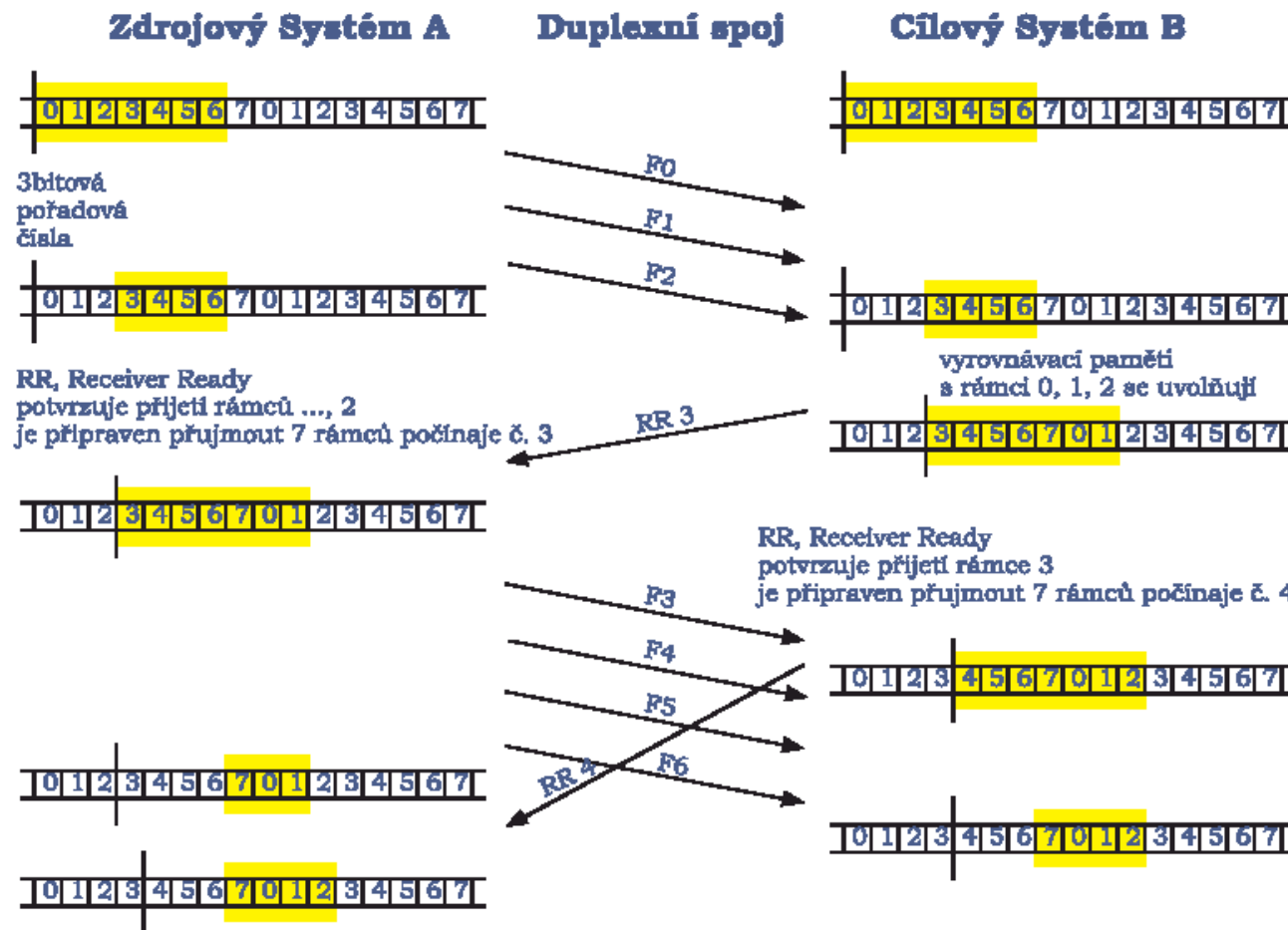
# Princip Sliding Window

11



# Příklad protokolu Sliding Window

12



# Vylepšování principu Sliding Window

13

- Příjímač může rámce potvrzovat bez povolení dalšího přenosu (**Receive Not Ready, RNR**)
- Pro pokračování přenosu musí příjímač vyslat normální potvrzení (**Receive Ready, RR**)
- V duplexním režimu příjímač používá pro potvrzování **piggybacking**:
  - ✓ Pokud má připravena k vyslání data a chce přijatá data potvrzovat, pošle **datový rámec** ve kterém uvede v potvrzovacím poli potvrzovací číslo
  - ✓ Pokud nemá připravena data pro vyslání, použije pro zaslání potvrzovacího čísla **potvrzovací rámec**, RR nebo RNR
  - ✓ Pokud má připravena k vyslání data a nechce přijatá data potvrzovat, pošle **datový rámec** ve kterém zopakuje v potvrzovacím poli poslední potvrzovací číslo

# Chybové řízení, typy chyb

14

## □ Porušené rámce

- ✓ Rámec přijímač rozpoznal, ale detekoval v něm chybné bity
- ✓ Detekce chyby (viz přednáška *Mechanismy přenosu dat*)
- ✓ Rámec obsahuje pole pro uvedení hodnoty kódu detekujícího výskyt chyby
- ✓ lze použít detekci chyb kontrolou parity
- ✓ lze použít detekci chyb kódy typu *Cyclic Redundancy Check*

## □ Ztracené rámce

- ✓ přijímač nerozpoznal rámec
- ✓ a chybu odhalí až z narušené posloupnosti pořadových čísel rámců

# Chybové řízení, techniky oprav

15

- Detekce chyby (viz přednáška *Mechanismy přenosu dat*)
- **ARQ**, *Automatic repeat request*, **mechanismy**:
  - ✓ Pozitivní potvrzování
  - ✓ Opakování vysílání po uplynutí časového limitu (timeout)
  - ✓ Negativní potvrzování a opakování vysílání
- **ARQ**, *Automatic repeat request*, **varianty**:
  - ✓ **Stop and wait**
  - ✓ **Go back N**
  - ✓ **Selective reject (selective retransmission)**
- Cílem ARQ je přeměna nespolehlivého datového spoje na spolehlivý datový spoj

# Stop-and-Wait

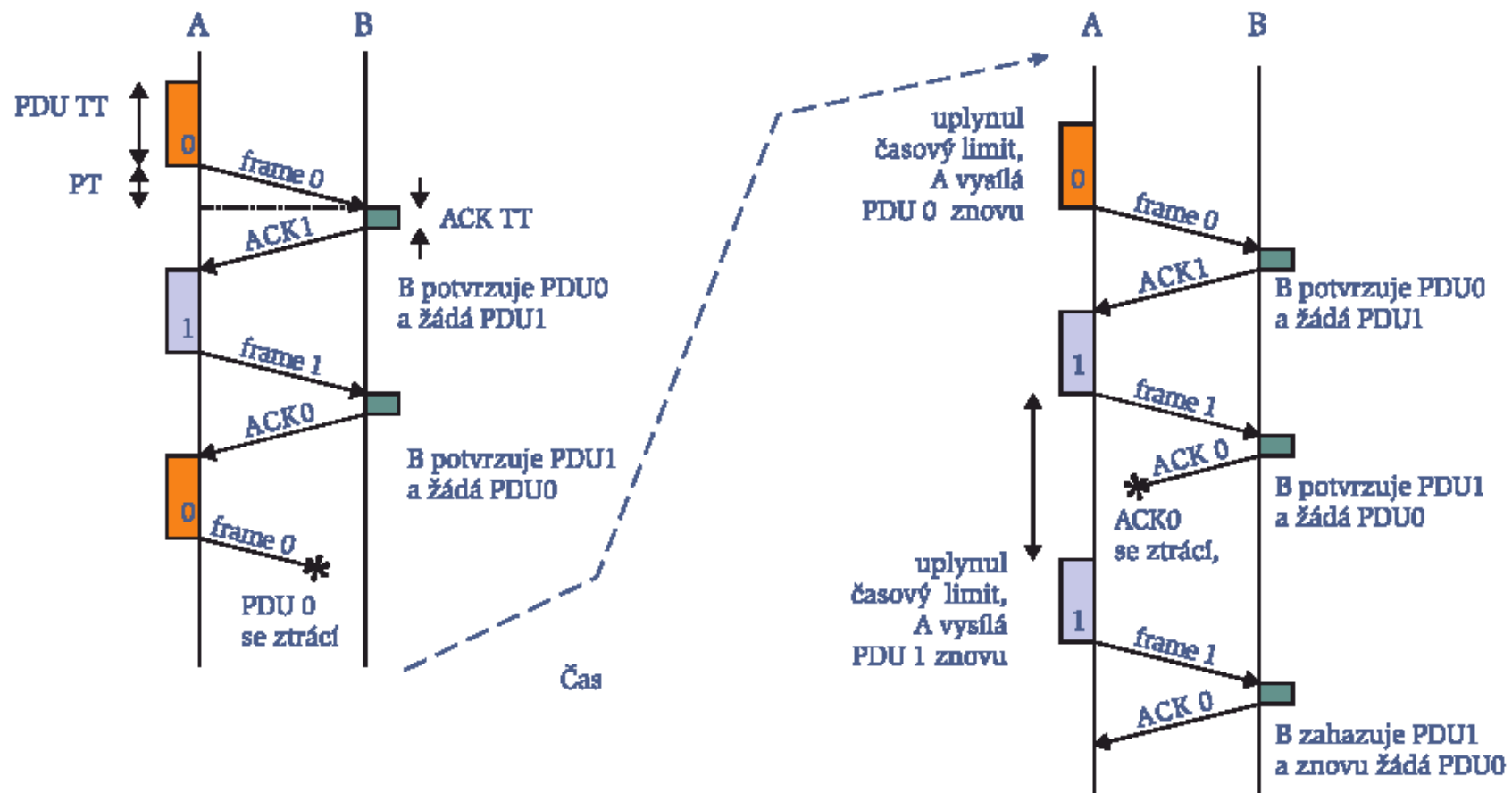
16

- Zdroj vyšle jeden datový rámec
- Čeká na přijetí potvrzovacího (rámce) ACK
- Postup po přijetí porušeného datového rámce
  - ✓ Zdroj rámce má definovaný **vysílací časový limit**
  - ✓ Pokud zdroj do uplynutí časového limitu po vyslání rámce nepřijme potvrzovací ACK, datový rámec vyšle znovu
- Porušení/ztrátu ACK zdroj není schopný rozpoznat, proto:
  - ✓ jakmile uplyne vysílací časový limit
  - ✓ vysílač zopakuje vyslání rámce
  - ✓ přijímač přijme též rámec 2×
  - ✓ řešení problému – alternativní používání **ACK0/ACK1** + alternativní číslování rámců 0/1
- Charakteristika – jednoduchý, neefektivní protokol



# Příklad protokolu Stop-and-Wait

17



# Go-Back-N, poškozený rámec dat

18

- Založeno na principu *sliding window*
- Pokud při příjmu nedojde k chybě dat,  
rámec ACK uvádí číslo očekávaného rámce
- Pro pozitivní potvrzování lze použít
  - ✓ samostatný **rámec RR**
  - ✓ **piggybacking** – potvrzování v **datových rámcích** protistrany
- Pokud dojde k chybě dat – posílá se **negativní potvrzení** rámce
  - ✓ pomocí rámce **REJ**, *Reject*
  - ✓ poškozený rámec (*i*) přijímač potvrzuje negativně (REJ *i*) a
  - ✓ poškozený rámec *i* a všechny následně přijaté rámce likviduje dokud správně nepřijme rámec *i*
  - ✓ vysílač se musí vrátit k rámcí *i*  
a tento a všechny následující vyslané rámce vyslat znovu

# Go-Back-N, ztracený rámec dat

19

- vysílač vyšle rámec  $i$  a rámec  $i + 1$  a rámec  $i$  se ztratí
  - ✓ přijímač při přijetí rámec  $i + 1$  rozpozná chybu v pořadovém čísle
  - ✓ přijímač si vyžádá zopakování přenosu rámců počínaje rámcem  $i$  a rámců po vyslaných po rámci  $i$  – pomocí REJ  $i$
- vysílač vyšle rámec  $i$  a ten se ztratí
  - ✓ přijímač nic nedostane, nic proto žádnou formou nepotvrzuje
  - ✓ vysílači uplyne časový limit a vyšle RR s indikací žádosti o partnerovo potvrzení (tzv. P bit = 1)
  - ✓ přijímač reaguje vysláním RR s číslem očekávaného rámce ( $i$ )
  - ✓ vysílač zopakuje vyslání rámce  $i$
  - ✓ nebo vysílač automaticky vysílá rámec  $i$  po uplynutí **přijímacího časového limitu**

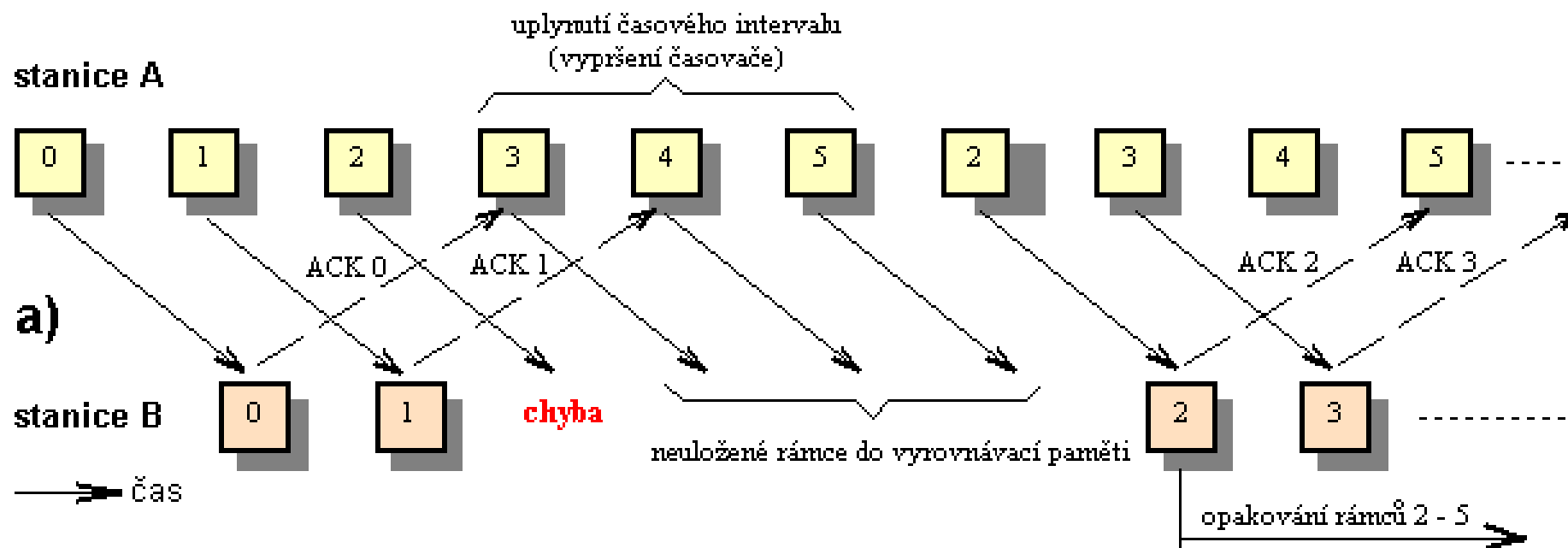
# Go-Back-N, ztracené potvrzení

20

- přijímač po přijetí rámeč  $i$  posílá potvrzení RR  $i + 1$
- potvrzení RR  $i + 1$  se ztratí
- potvrzení jsou kumulativní, může přijít RR  $i + n$  dříve než uplyne časový limit pro rámeč  $i$
- pokud uplyne časový limit,  
vysílač  $i$  vysílá RR s P-bitem, viz předchozí obrázek
  - ✓ toto se opakuje po uplynutí časového limitu reakce na P-bit až do vyčerpání limitu počtu opakování – pak se dělá *reset*
- ztratí se potvrzení REJ
  - ✓ stejné situace jako při ztrátě rámce –  
vysílač vyšle rámeč  $i$  a ten se ztratí

# Go-Back-N

21



# Selektivní odmítání, Selective Reject

22

- opakuje se přenos pouze odmítnutých rámců
- po sobě přicházející rámce jsou přijímačem přijímány a pamatovány ve vyrovnávacích pamětech
- minimalizují se opakovaná vysílání
- přijímač musí udržovat velkou vyrovnávací paměť
- složitější logika ve vysílači

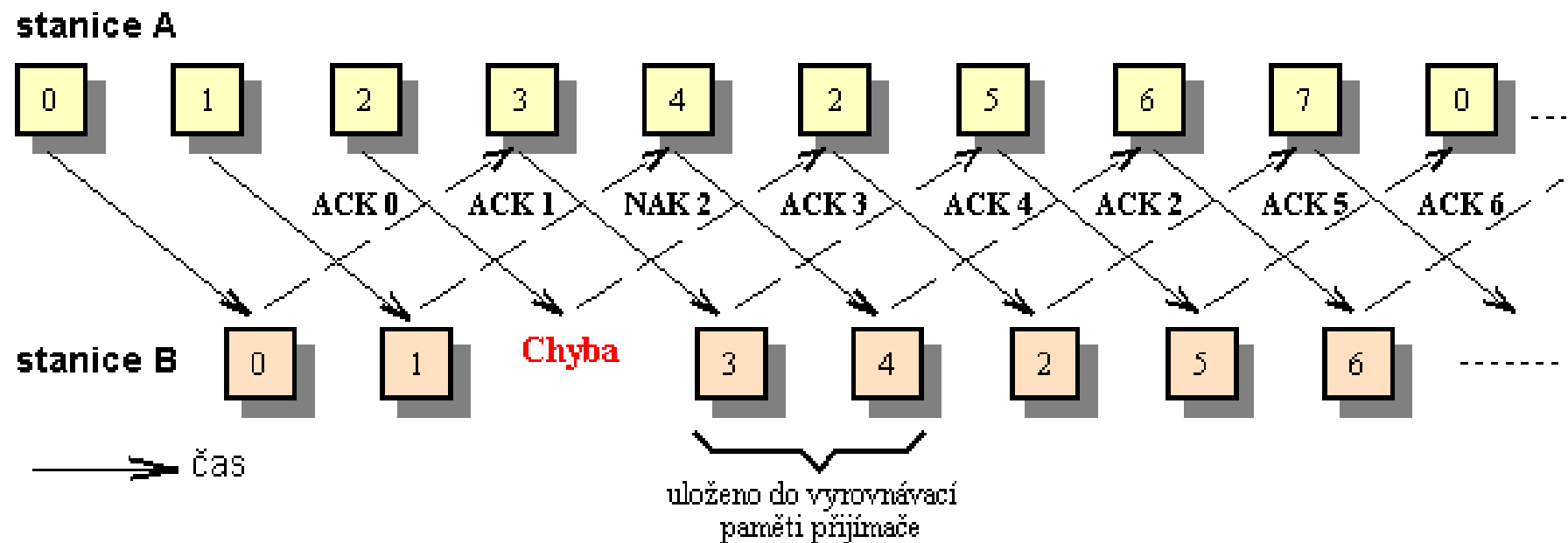
# Realizace Selective Reject

23

- .. **implicit retransmission** – vysílač zjistí z čísla potvrzení, že se ztratil rámeček a pošle jej znovu
- .. **explicit request** – přijímač zjistí z čísla rámce, že se předchozí ztratil a pošle negativní potvrzení
- .. přijímač po vyslání NAK přechází do tzv. **retransmission state**
  - ✧ nevysílá žádné ACK, čeká na rámeček, který negativně potvrdil (může být pouze jeden nevyřízený NAK)
  - ✧ jinak by mohlo dojít k definitivní ztrátě rámce, protože ACK potvrzuje i všechny předchozí rámce

# Selective Reject

24





# Číslování rámců modulo N

25

- .. jaká musí být velikost okének, aby nedocházelo k záměně?
- .. **Stop-and-wait**
  - ✧ stačí dvě hodnoty (0,1) pro rozpoznání tj. modulo 2
- .. **Go-back-N**
  - ✧ při velikosti vysílacího okénka K stačí  $N=K+1$
  - ✧ při ztrátě všech potvrzení v okénku musí přijímač rozpoznat, že se jedná o rámeček v novém okénku
- .. **Selective-reject**
  - ✧ při vysílání okénko nepřekročíme, ale můžeme ho překročit při potvrzování  $\Rightarrow N=2K+1$