

1. Charakterizujte simplexní, duplexní a poloduplexní přenos

Týka sa prenosu v oboch smeroch

Duplexny prenos: je možný v oboch smeroch, a to súčasne.

Poloduplexny prenos: je možný v obou smerech, ale nikoli súčasne

Simplexny prenos: je možný len v jednom smere

priklady: optické vlákno bez WDM

digitálne TV vysielanie systémom DVB-T

obecné R a TV vysielanie, jednosmerne satelitne prenosy

Týka sa komunikácie všeobecne: neide iba o to, čo umožňuje prenosové médium, napr. nad plne duplexnou prenosovou cestou je možné komunikovať len poloduplexne (stylom otázka - odpoveď)

2. Jaký problém řeší synchronizace? Co hrozí v případě ztráty synchronizace?

Každý bit je prenášaný v rámci určitého bitového intervalu

- tj. prenos bitu nie je okamžitý, ale trvá určitú dobu (bitový interval)
- prenášaná data reprezentuje stav signálu počas bitového intervalu

Prijemca vyhodnocuje stav prenášaného signálu niekde v rámci bitového intervalu

- rozhodujúci je okamžik vyhodnotenia signálu
- na základe vyhodnotenia okamžitého stavu signálu potom usudzuje na to, aké data sú prenášané

Problém synchronizácie: prijemca sa musí trafiť do správneho bitového intervalu (inak prijme nezmyselné data)

3. Charakterizujte asynchronní a arytmiický přenos. Jaký je problém s obvyklou terminologií?

Asynchronny prenos: jeden z možných spôsobov zaistenia synchronizácie, úplne chyba synchronizácia, tj. bitový interval nemá konštantnú dĺžku, začiatok a koniec každého bitového intervalu musí byť explicitne vyznačený (potrebne aspoň trojhodnotová logika)

Terminologický problém: keď sa povie asynchronny, nemyslí sa tým táto varianta, ale to, čo je správne označované ako arytmiický, táto varianta sa dnes prakticky nepoužíva

Arytmický prenos:

- snažia sa prenášať celé skupiny bitov, tvoriace znaky (4 až 8 bitov, dnes skor 8),
- na začiatku každého znaku je tzv. start-bit: slúži k tomu, aby si prijemca zrovnal svoje hodinky, predpoklad: po zrovnaní na začiatku každého znaku budú hodinky prijemca tikat po celú dobu trvania daného znaku, tj. prijemca bude správne vyhodnocovať jednotlivé bity v rámci znaku
- časové prodlevy medzi jednotlivými znakmi môžu byť rôzne veľké: preto arytmiický prenos (chyba mu rytmus prenosu jednotlivých znakov), počas prodlevy medzi znakmi sa hodinky prijemcu môžu ľubovoľne rozísť, na začiatku ďalšieho znaku budú znovu zosynchronizované pomocou start-bitu

4. Jaké jsou možnosti udržování trvalé synchronizace?

Bloky sú ľubovoľne dlhé = už nie je možné spoľahnúť na to, že hodinky prijemcu vydržia a nerozídu sa počas prenosu.

Synchronizáciu udržiavať priebežne: priebežne zrovnávať hodinky

- skrz samostatný synchronizačný (časový) signál: prenáša tkanie hodín odosielateľa, príliš nákladne, samostatný signál nie je k dispozícii
- skrz redundantné kódovanie: zahrnutie časového signálu do kódovania jednotlivých bitov, príklad kódovanie Manchester

- synchronizacia z dat

5. Co je izochronní přenos? Které varianty multiplexu jsou izochronní a které nikoli?

To zn. Prebiehajúce v rovnakom case, vhodné pre multimedialne prenosy (obraz, zvuk), môže byť určite prenosové oneskorenie (500 ms), ale je požadovaná vysoká pravidelnosť (prenosové oneskorenie je konštantné a nemení sa).

Data majú zaručené, za ako dlho sa dostanu k cieľu, nemusí byť hneď ale je to pravidelné. Idú asynchronne data, vkladajú sa do synchronného prenosového mechanizmu (napr. do časových slotov), medzi časťami dát sú vždy celistvé násobky prázdnych slotov intervalov. Prepojovanie okruhov je izochronné.

Časový multiplex (TDM) zachováva izochronný charakter.

Statický multiplex a prepojovanie paketov nie sú izochronné.

6. Co je bitstream? K čemu se hodí? Je dostupný v ČR?

Bitový prúd je telekomunikačná služba: synchronný prenos bitov medzi dvoma lokalitami, má konštantnú prenosovú rýchlosť a prenosové oneskorenie.

Môže sa chápať ako služba fyzickej vrstvy: službu charakteru odoslí/prijmi bit so synchronným spôsobom fungovania.

Je vhodným podložením pre prenosové služby vyšších úrovní, nad bitovým prúdom je možné realizovať prenos (linkových) rámcov, rôzne druhy prenosov (paketový/best effort, izochronný, s QoS)

7. Co zajišťuje tzv. framing? Jaký je princip bitově a znakově orientovaných protokolů linkové vrstvy?

Framing: synchronizácia na úrovni rámcov, ide o správne rozpoznanie linkového rámca.

Znakovo orientované: prenasajú data členené na znaky, pre vyznačenie začiatku a konca používajú špeciálne znaky ASCII sady.

Bitovo orientované: prenasajú data ako postupnosť bitov (nečlenia ich na znaky), pre vyznačenie začiatku a konca využívajú špeciálne bitové postupnosti (tzv. krídlové značky)

8. Jak se zajišťuje transparence dat u znakově a bitově orientovaných protokolů?

Problém: ako vždy spoľahlivo spoznať, ktoré data to sú: riadiace (hlavičky, patičky, príkazy...) a majú byť interpretované alebo užitočné data a nemajú byť nijak interpretované.

- Samostatné prenosové kanály pre riadiacie príkazy a pre data (nie vždy možné)
- slúčenie príkazov a dát do jedného prenosového kanálu (častejšie, nutné mať schopnosť rozpoznať, kedy ide o užitočné data)

Znakovo orientované protokoly: Prefixácia špeciálnym ESCAPE znakom:

- pred každým znakom, ktorý má mať význam riadiaceho znaku sa umiestni špeciálny escape znak (napr. DLE data link escape)
- prípadný výskyt špeciálnych escape znakov v užitočných dátach sa rieši jeho zdvojením
- tzv. charakter stuffing

Bitovo orientované protokoly: špeciálna bitová postupnosť (krídlová značka)

- prípadný výskyt špeciálnej bitovej postupnosti v užitočných dátach sa rieši pomocou bit-stuffingu

9. Jaké jsou možnosti zajištění spolehlivosti (v závislosti na dostupnosti zpětné vazby)?

Môže byť realizované na ktorejkoľvek vrstve okrem fyzickej. Princíp a spôsob realizácie je v zásade rovnaký na všetkých vrstvách.

Podmienkou je rozpoznať, že došlo k nejakej chybe pri prenose.

Pri nespolahlivom prenose, sa neda robit nic.

Pri spolahlivom sa postarat o napravu. **Moznosti:**

- pouzitie samoopravných kodov, napri Hammingove kody, problemom je velka redundacia, ktora zvsuje objem prenasanych dat, pouziva sa vynimocne
- pomocou potvrdzovania: prijemca si necha znovu zaslat poskodene data, podmienkou je existencia spatnej vazby (aspon polovicny duplex, aby prijemca mohol kontaktovat odosielatela).

10. Jak se používá parita a kontrolní součet pro detekci chyb při přenosech?

Parita:

- **Paritny bit:** bit pridany navyc k datovym bitom. Suda parita (paritny bit je nastaveny tak, aby celkovy pocet 1 bol sudy), licha parita (1 lichy), jednickova parita (paritny bit pevne nastaveny na 1, neni zabezpecovaci efekt), nulova parita (nastaveny na 0).
- **Priecna parita:** po jednotlivých bytoch/slovach, informacie o tom, ktory byte je poskodeny je nadbytocna, aj tak sa posielala rovno cely blok (ramec, pakec).
- **Podelna parita:** parita zo vsetkych rovnolahlych bitov vsetkych bytov/slov

Kontrolny sucet: jednotlivé byty/slova/dvojslova tvoriace prenasany blok sa interpretuju ako cisla a scitaju sa, vysledny sucet sa pouzije ako zabezpecovaci udaj (obvykle sa pouzije iba cast suctu, napr nizsi byte alebo nizsie slovo).

Alternativa: miesto suctu sa pocita XOR jednotlivých bitov.

Je ucinnejsi ako parita, ale stale je miera zabezpecenia nizka.

11. Jak se používá CRC pro detekci chyb při přenosech?

CRC = Cyclic Redundancy Check

Postupnost bitu, tvoriaca blok dat, je interpretovana ako polynom, polynom nad telesom charakteristiky 2, kde bity su jeho koeficienty.

$$\dots + 1x^{14} + 0x^{13} + 0x^{12} + 1x^{11} + \dots$$

... 1 0 1 0 0 1 0 0 ...

Tento polynom je vydeleny inym polynomom, vysledkom je podiel a zvsok, v roli zabezpecenia sa pouzije zvsok po deleni charakteristickeho polynomom.

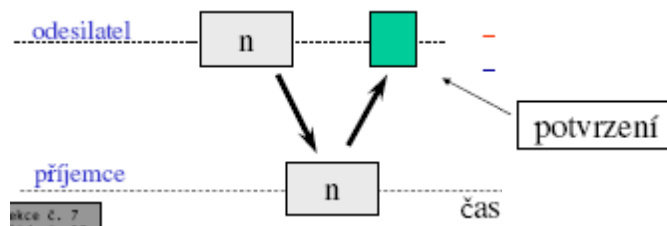
Schopnosti detekcie su vynikajuce: vsetky shluky chyb s lichym pocetom bitov, vsetky shluky chyb do velkosti n bitu, kde n je stupen charakteristickeho polynomu, vsetky shluky chyb velkosit > n+1 s pravdepodobnostou 99,999% (CRC 32)



Spolahlivost CRC kodov sa opiera o vysledky z algebry, samotny vypocet je velmi jednoduchy a moze byt lahko implementovany v HW, pomocou XOR hradiel a posuvnych registrov.

12. Jaký je princip potvrzování? Jak funguje jednotlivé a kontinuální potvrzování?

Ide o obecnejsi mechanizmus, ktory sluzi viac ucelom sucasne: 1. zaistenie spoehlivosti (umozuje, aby si prijemca vyziadal opakovane zaseialanie poskodeneho ramca), 2. riadenie toku (aby prijemca mohol regulovat tempo, akym odosielatel posielala data).



Sposoby:

- kladne a zaporne potvrdzovanie: potvrdzujú sa spravy, resp chybné prijaté bloky
- jednotlivé a kontinuálne potvrdzovania: podľa toho, či odosielateľ vždy čaká na potvrdenie alebo odosiela do fronty.
- Samostatne a nesamostatne potvrdzovanie: či potvrdenie cestuje ako samostatný rámec/paket, alebo je vnorené do datového paketu.
- Metóda okienka

Jednotlivé potvrdzovanie: Stop&Wait ARQ

- ide o samostatné jednotlivé potvrdzovanie, potvrdenie je prenášané ako samostatný (riadiaci) blok, potvrdzovaný je každý jeden paket (kladne, zaporne, timeout)
- príbeh: odosielateľ odosle datový rámec a čaká na jeho potvrdenie (kladne, zaporne), ďalší rámec neodosiela, prijemca odosle potvrdenie, podľa druhu potvrdenia odosielateľ buď odosle ďalší rámec alebo opakuje prenos, timeout interpretuje ako zápornú odpoveď
- jednoduchá a priamociarne interpretácia, charakter prenosu čisto poloduplexný, napr. protokoly IPX/SPX
- má zmysel v LAN (kratka odozva), nie WAN (zpozdění veľké)

Kontinuálne potvrdzovanie: continuous ARQ

- **idea:** odosielateľ bude vysielat datové rámce dopredu, a príslušné potvrdenia prijímať priebežne, s určitým zpozdění
- ak dostane zápornú odpoveď: **1. selektívne opakovanie:** odosle iba rámec, ktorý sa poškodil (prijímateľ musí ukladať do bufferu, narádne hospodárenie s pamäťou), **2. návrat späť:** rieši

13. Jak funguje kontinuální potvrzování s návratem a se selektivním opakováním?

Selektivne opakovanie: odosle iba rámec, ktorý sa poškodil (prijímateľ musí ukladať do bufferu, narádne hospodárenie s pamäťou), ďalšie rámce, ktoré sa podarilo preniesť sa neprenesajú, setri prenosovú kapacitu.

Návrat späť: alternatíva k selektívnemu, rieši situáciu, keď príde záporné potvrdenie so zpozdění, keď boli odoslané ďalšie rámce, zahodí sa, odosielateľ odosle poškodený rámec a tak posiela, nasledujúce.

Nevýhody: plytvánie prenosovou kapacitou

Výhody: prijemca čaká len ak poškodený rámec a tak pokračovať v prijímaní ostatných.

Lepšie znáša väčšie prenosové zpozdění, používa sa v protokoloch TCP/IP

14. Jaký je rozdíl mezi samostatným a nesamostatným potvrzováním? Jak funguje piggybacking?

Samostatne: potvrdenie je prenášané ako samostatný rámec špeciálneho typu, spojené s relatívne vysokou reziou, samostatne potvrdenie je malé, obale je veľký

Nesamostatne: potvrdenie je zasielane ako súčasť datových rámcov, prenášaných v opačnom smere, ktoré sú potvrdzované, tzv. **piggybacking**

15. Jak funguje a k čemu slouží metoda okénka?

Idea: spojit potvrzovanie s riadenim toku na úrovni ramcov.

Odosielateľ si udrzuje vysielacie okienko, veľkosť udáva, koľko ramcov smie vyslať dopredu bez potvrdenia, napr TCP.

Veľkosť okienka určuje:

- odosielateľ podľa správania a vlastností siete (stanoví max veľkosť)
- prijemca podľa svojich možností (dostupnosť bufferu)

16. Jaký problém řeší řízení toku? Na jaké úrovni (vrstvě) může řízení toku fungovat?

Podstata: rýchlosť (vypočetná sila) dvoch komunikujúcich strán môže byť výrazne odlišná, je nutné sa vyvarovať toho, aby prijemca nestihal (kvôli svojej rýchlosti, nedostatku bufferu...) a musel kvôli tomu zahadzovať prenesené pakety/ramce.

Riadiť tempo podľa možnosti prijímateľa.

Riesiť na rôznych úrovniach:

- na úrovni jednotlivých znakov: tzv hardwarový alebo softwarový handshake
- na úrovni celých ramcov: prijímateľ reguluje, či chce alebo nechce poslať ďalšie ramce