1. Charakterizujte simplexní, duplexní a poloduplexní přenos

Týka sa prenosu v oboch smeroch
Duplexny prenos: je možný v oboch smeroch, a to súčasne.
Poloduplexný prenos: je možný v obou smrech, ale nikoli souasne
Simplexný prenos: je mozny len v jednom smere
priklady: opticke vlakno bez WDM
digitalne TV vysielanie systemom DVB-T
obecne R a TV vysielanie, jednosmerne satelitne prenosy
Tyka sa komunikacie vseobecne: neide iba o to, co umoznuje prenosove medium, napr. nad plne duplexnou prenosovou cestou je mozne komunikovat len poloduplexne (stylom otazka - odpoved)

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

Kazdy bit je prenasany v ramci urciteho bitoveho intervalu

- tj. prenos bitu nie je okamzity, ale trva urcitu dobu (bitovy interval)
- prenasana data reprezentuje stav signalu behom bitoveho intervalu

Prijemca vyhodnocuje stav prenasaneho signalu niekde v ramci bitoveho intervalu

- rozhodujuci je okamzik vyhodnotenia signalu
- na zaklade vyhodnotenia okamziteho stavu signalu potom usudzuje na to, ake data su prenasane

Problem synchronizace: prijemca sa musi trafit do spravneho bitoveho intervalu (inak prijme nezmyselne data)

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

Asynchronny prenos: jeden z moznych sposobov zaistenia synchronizacie, uplne chyba chyba synchronizacia, tj. Bitovy interval nema konstatnu dlzku, zaciatok a koniec kazdeho bitoveho intervalu musi byt explicitne vyznaceny (potrebna aspon trojhodnotova logika)
Terminologicky problem: ked sa povie asynchronny, nemysli sa tym tato varianta, ale to, co je spravne oznacovane ako arytmicky, tato varianta sa dnes prakticky nepouziva
Arytmicky prenos:

- snazi sa prenasat cele skupiny bitov, tvoriace znaky (4 az 8 bitov, dnes skor 8),
- na zaciatku kazdeho znaku je tzv. start-bit: sluzi k tomu, aby si prijemca zrovnal svoje hodinky, predpoklad: po zrovnani na zaciatku kazdeho znaku budu hodinky prijemca tikat po celu dobu trvania daneho znaku, tj. Prijemca bude spravne vyhodnocovat jednotlive bity v ramci znaku
- casove prodlevy medzi jednotlivymi znakmi mozu byt rozne velke: preto a-rytmicky prenos (chyba mu rytmus prenosu jednotlivych znakov), behom prodlevy medzi znakmi sa hodinky prijemcu mozu lubovlne rozist, na zaciatku dalsieho znaku budu znovu zosynchronizovane pomocou start-bitu

4. Jaké jsou možnosti udržování trvalé synchronizace?

Bloky su lubovolne dlhe = uz nie je mozne spoliehat na to, ze hodinky prijemcu vydrzia a nerozidu sa behom prenosu.
Synchronizaciu udrzovat priebezne: priebezne zrovnavat hodinky

- skrze samostatny synchronizacny (casovy) signal : prenasa tikanie hodiniek odosielatela, prilis nakladne, samostany signal nie je k dispozicii
- skrze redundantne kodovanie: zahrnutie casoveho signalu do kodovanie jednotlivych bitov, priklad kodovanie Manchaster

- synchronizacia z dat

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

To zn. Prebiehajuce v rovnakom case, vhodne pre multimedialne prenosy (obraz, zvuk), moze by turcite prenosove spozdeni (500 ms), ale je pozadovana vysoka pravidelnost (prenosove spozdeni je konstatne a nemeni sa).
Data maju zarucene, za ako dlho sa dostanu k cielu, nemusi byt hned ale je to pravidelne.
Idu asynchronne data, vkladane do synchronneho prenosoveho mechanizmu (napr do casovyh slotov), medzi castami dat su vzdy celistve nasobky prazdnych slotov intervalov
Prepojovanie okruhov je isochronne.
Casovy multiplex (TDM) zachovava isochronny charakter.
Staticky multiplex a prepojovanie paketov nie su isochronne.

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

Bitovy prud je telekomunikacna sluzba: synchronny prenos bitov medzi dvoma lokalitami, ma konstatnu prenosovu rychlost a prenosove spozdeni.
Mozne chapat ako sluzbu fyzickej vrsty: sluzbu charatkeru odosli/prijmi bit so synchronnym sposobom fungovania.
Je vhodnym podlozim pre prenosove sluzby vyssich urovni, nad bitovym prudom je mozne realizovat prenos (linkovych) ramcov, rozne druhy prenosov (paketovy/best effort, isochronny, s QoS)

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

Framing: synchronizacia na urovni ramcov, ide o spravne rozpoznanie linkoveho ramca.
Znakovo orientovane: prenasaju data clenene na znaky, pre vyznacenie zaciatku a konca pouzivaju specialne znaky ASCII sady.
Bitovo orientovane: prenasaju data ako postupnost bitov (neclenia ich na znaky), pre vyznacenie zaciatku a konca vyuzivaju specialne bitove postupnosti (tzv. kridlove znacky)

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

Problem: ako vzdy spolahlivo spoznat, ktore data to su: riadiace (hlavicky, paticky, prikazy...) a maju byt interpretovane alebo uzitocne data a nemaju byt nijak interpretovane.

- Samostatne prenosove kanaly pre riadiacie prikazy a pre data (nie vzdy mozne)
- slucenie prikazov a dat do jedneho prenosoveho kanalu (castejsie, nutne mat schopnost rozpoznat, kedy ide o uzitocne data)

Znakovo orintovanie protokoly: Prefixacia specialnym ESCAPE znakom:

- pred kazdy znak, ktory ma mat vyznam riadiaceho znaku sa umiestni specialny escape znak (napr DLE data link escape)
- pripadny vyskyt specialnych escape znakov v uzitocnych datach sa riesi jeho zdvojenim
- tzv. charakter stuffing

Bitovo orientovane protokoly: specialna bitova postupnost (kridlova znacka)

- pripadny vyskyt specialnej bitovej postupnosti v uzitocnych datach sa riesi pomocou bit-stuffingu

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

Moze byt realizovane na ktorejkolvek vrstve okrem fyzickej. Princip a sposob realizacie je v zasade rovnaky na vsetkych vrstvach.
Podmienkou je rozpoznat, ze doslo k nejakej chybe pri prenose.

Pri nespolahlivom prenose, sa neda robit nic.
Pri spolahlivom sa postarat o napravu. **Moznosti:**

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

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

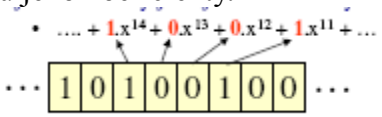
Parita:

- **Paritny bit:** bit pridany naviac 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 jednotlivych bytoch/slovach, informacie o tom, ktory byte je poskodeny je nadbytocna, aj tak sa posiela rovno cely blok (ramec, pakec).
- **Podelna parita:** parita zo vsetkych rovnolahlych bitov vsetkych bytov/slov

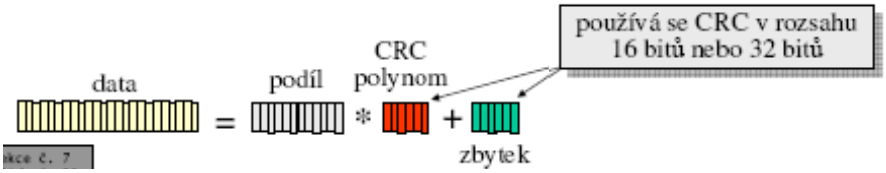
Kontrolny sucet: jednotlive byty/slova/dvojslova tvoriace prenasany blok sa interpretuju ako cisla a scitaju sa, vysledny sucet sa pouzije ako zabezpecovaci udaj (obvykle sa pouzije iba cast suctu, napr nizsi byte alebo nizsie slovo.
Alternativa: miesto suctu sa pocita XOR jednotlivych bitov.
Je ucinnejsi 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.



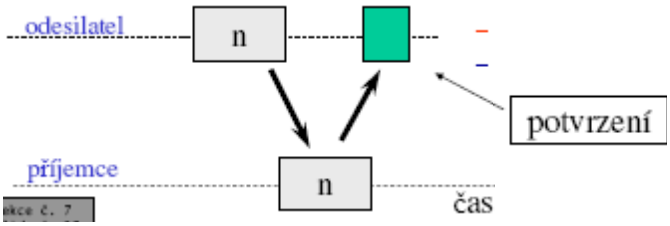
Tento polynom je vydeleny inym polynomom, vysledkom je podiel a zvysok, v roli zabezpecenia sa pouzije zvysok po deleni charakteristickym polynomom.
Schopnosti detekcie su vynikajuce: vsetky shluky chyb s lichym poctom bitov, vsetky shluky chyb do velkosti n bitu, kde n je stupen charakteristickeho polynomu, 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 spolehlivosti (umoznuje, aby si prijemca vyziadal opakovane zaseialanie poskodeneho ramca), 2. riadenie toku (aby prijemca mohol regulovat tempo, akym odosielatel posiela data.



Sposoby:

- kladne a zaporne potvrdzovanie: potvrdzuju sa spravy, resp chybne prijate bloky
- jednotlive a kontinualne potvrdzovania: podla toho, ci odosielatel vzdy caka na potvrdenie alebo odosiela do fronty.
- Samostatne a nesamostatne potvrdzovanie: ci potvrdenie cestuje ako samostatny ramec/paket, alebo je vnorene do datoveho paketu.
- Metoda okienka

Jednotlive potvrdzovanie: Stop&Wait ARQ

- ide o samostatne jednotlive potvrdzovanie, potvrdenie je prenasane ako samostatny (riadiaci) blok, potvrdzovany je kazdy jeden paket (kladne, zaporne, timeout)
- pribeh: odosielatel odosle datovy ramec a caka na jeho potvrdenie (kladne, zaporne), dalsi ramec neodosiela, prijemca odosle potvrdenie, podla druhu potvrdenia odosielatel bud odosle dalsi ramec alebo opakuje prenos, timeout interpretuje ako zapornu odpoved
- jednoducha a priamociara interpretacia, charatker prenosu cisto poloduplexny, napr. protokoly IPX/SPX
- ma zmysel v LAN (kratka odozva), nie WAN (zpozdeni velke)

Kontinualne potvrdzovanie: continuous ARQ

- **idea:** odosielatel bude vysielat datove ramce dopredu, a prislusne potvrdenia prijimat priebezne, s urcitym zpozdenim
- ak dostane zapornu odpoved: **1. selektivne opakovanie**: odosle iba ramec, ktory sa poskodil (prijimatel musi ukladat do bufferu, narocne hospodarenie s pamatou), **2. navrat spat:** riesi

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

Selektivne opakovanie: odosle iba ramec, ktory sa poskodil (prijimatel musi ukladat do bufferu, narocne hospodarenie s pamatou), dalsie ramce, ktore sa podarilo preniest sa neprenasaju, setri prenosovu kapacitu.
Navrat spat: alternativa k seletivnemu, riesi situaciu, ked pride zaporne potvrdenie so zpozdenim, ked boli odoslane dalsie ramce, zahodia sa, odosielatel odosle poskodeny ramec a tak posiela, nasledujuce.

Nevyhody: plytvanie prenosovou kapacitou
Vyhody: prijemca caka len an poskodeny ramec a tak pokracovat v prijimani ostatnych.
Lepsie znasa vetsie prenosove zpozdeni, pouziva sa v protokoloch TCP/IP

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

Samostatne: potvrdenie je prenasane ako samostatny ramec specialneho typu, spojene s relativne vysokou reziou, samostatne potvrdenie je male, obale je velky
Nesamostatne: potvrdenie je zasielane ako sucast datovych ramcov, prenasanych v opacnom smere, ktore su potvrdzovane, tzv. **piggybacking**

15. Jak funguje a k čemu slouží metoda okénka?

Idea: spojit potvrdzovanie s riadenim toku na urovni ramcov.
Odosielatel si udrzuje vysielacie okienko, velkost udava, kolko ramcov smie vyslat dopredu bez potvrdenia, napr TCP.
Velkost okienka urcuje:

- odosielatel podla spravania a vlastnosti siete (stanovi max velkost)
- prijemca podla svojich moznosti (dostupnost bufferu)

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

Podstata: rychlost (vypocetna sila) dvoch komunikujucich stran moze byt vyzrazne odlisna, je nutne sa vyvarovat toho, aby prijemca nestihal (kvoli svojej rychlosti, nedostatku bufferu...) a musel kvoil tomu zahadzovat prenesene pakety/ramce.
Riadit tempo podla moznosti prijimatela.
Riesit na roznych urovniach:

- na urovni jednotlivych znakov: tzv hardwarovy alebo softwarovy handshake
- na urovni celych ramcov: prijimatel reguluje, ci chce alebo nechce poslat dalsie ramce