

Adatkapcsolati réteg



Az adatkapcsolati réteg

- ⬡ A fizikai átviteli hibáinak elfedése a hálózati réteg előtt
- ⬡ Keretezés
 - ⬡ Adatfolyam tördelése
 - ⬡ Küldés sorrendben
 - ⬡ Nyugtázás (megbízható szolgáltatás)
 - ⬡ Kerethatárok felismerése



Az adatkapcsolati réteg

- ⬡ Forgalomszabályozás
 - ⬡ Elárasztás elleni védelem (felsőbb rétegekben is)
 - ⬡ Adó tájékoztatása a vevő szabad puffereiről
- ⬡ Összeköttetés iránya (szimplex, fél duplex, duplex)
- ⬡ Osztott csatornához való hozzáférés szabályozása

Tervezési szempontok

- ⬡ Jól definiált interfész biztosítása a hálózati rétegnek
- ⬡ Az átviteli hibák kezelése
- ⬡ Az adatforgalom szabályozása
- ⬡ Keretezés



Szolgáltatások: Hálózati réteg

- ⬡ Nyugtázatlan összeköttetés nélküli szolgáltatás
- ⬡ Nyugtázott összeköttetés nélküli szolgáltatás
- ⬡ Nyugtázott összeköttetés-alapú szolgáltatás

Nyugtázatlan összeköttetés mentes szolgálat

- ◻ Egymástól független keretek küldése a forrástól a célig
- ◻ Nincs előzetes kapcsolatépítés és bontás
- ◻ Az elveszett kereteket nem állítja helyre a réteg
- ◻ Alkalmazhatóság
 - ◻ Alacsony hibaarány esetén (javítás a felsőbb rétegekben)
 - ◻ Valós idejű adatforgalom esetén (hangátvitel)
 - ◻ Ethernet

Nyugtázatlan összeköttetés mentes szolgálat

- ◡ Egymástól független keretek küldése a forrástól a célig
- ◡ Nincs előzetes kapcsolatépítés és bontás
- ◡ Az elveszett kereteket nem állítja helyre a réteg
- ◡ Alkalmazhatóság
 - ◡ Alacsony hibaarány esetén (javítás a felsőbb rétegekben)
 - ◡ Valós idejű adatforgalom esetén (hangátvitel)
 - ◡ Ethernet

Nyugtázott összeköttetés mentes szolgálat

- ⬡ Szintén nincs előzetes kapcsolatépítés és bontás
- ⬡ Minden egyes keret megérkezését nyugtázza a cél állomás
- ⬡ Alkalmazhatóság
 - ⬡ Pl.: vezeték nélküli kapcsolatoknál

Nyugtázott összeköttetés-alapú szolgálat

- ◡ Összeköttetés felépítése az adatátvitel előtt
- ◡ Sorszámozott keretek
 - ◡ Minden keret garantáltan megérkezik
 - ◡ Minden keret garantáltan egyszer érkezik meg
 - ◡ Minden keret a megfelelő sorrendben érkezik meg
- ◡ Alkalmazhatóság:
 - ◡ Műhold
 - ◡ Nagy távolságú telefonvezeték

Keretezés

- ⬡ A fizikai réteg nem garantál hibamentes átvitelt
- ⬡ Zajos csatorna
- ⬡ A réteg feladata ezen hibák jelzése, javítása
- ⬡ Keretezés
 - ⬡ Az adatfolyam feldarabolása kisebb részekre (keret)
 - ⬡ Ellenőrző összegek számítása minden kerethez

Keretezés

- ◡ Az átvitel ellenőrzése
 - ◡ A fogadott adatok alapján az ellenőrző összeg kiszámítása
 - ◡ A fogadott és az újrászámolt összeg összehasonlítása
- ◡ A kerethatárok jelölésének problémái
 - ◡ Szünetek beszúrása az egyes keretek közé?

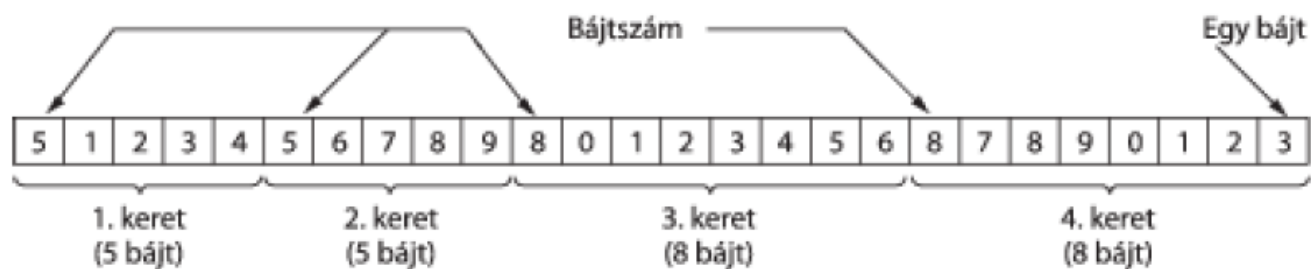
Keretezési módszerek

- ⬡ Karakterszámlálás
- ⬡ Kezdő- és végkarakterek használata bájtbeszúrással.
- ⬡ Kezdő- és végkarakterek használata bitbeszúrással.
- ⬡ Fizikai rétegbeli kódolás megsértése

Karakterszámlálás

- ⬡ Minden keret fejrészámban megadásra kerül a keret karaktereinek száma
- ⬡ A vevő a fejrészből tudja, hány karakter tartozik a kerethez
- ⬡ Illetve, hogy hol lesz a keret vége (a következő eleje)

Karakterszámlálás



(a)



(b)

Karakterszámlálás problémái

- ⬡ A karakterszám mezőt is érheti átviteli hiba
- ⬡ A vevő kiesik a szinkronból, nem találja a következő keretet
- ⬡ Hibás ellenőrző összeg \rightarrow hibás keret
 - ⬡ de hol kezdődik a következő keret?
- ⬡ Az újraküldés sem megoldás

Kezdő- és végkarakterek karakterbeszúrással

- ⬡ A kerethatárok jelzése egy különleges karakterrel
 - ⬡ Régebben eltérő protokollok eltérő bájtot használtak
 - ⬡ Manapság egységes "jelző bájtt" (flag byte) jellemző
- ⬡ Ha a vevő kiesik a szinkronból csak megvárja a következő flag byte-ot
 - ⬡ Az első flag byte az adott keret vége
 - ⬡ A második flag byte a következő keret eleje

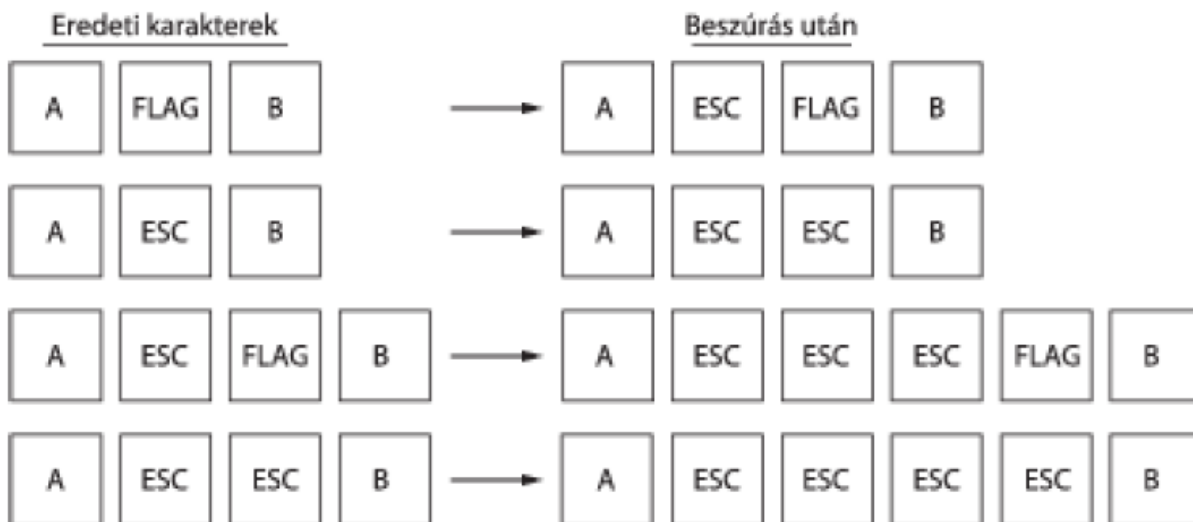
Karakterbeszúrás problémája

- Bináris adatátvitelnél belekerülhet a jelzőbájt mintája
- Adatkapcsolati réteg -> kivételbájt beszúrás (ESC)
- Mi történik ha a flag byte-nak megfelelő karaktert akarunk átvinni? (bináris átvitel)
 - Kivétel bájt (escape byte) beszúrása
- És ha escape byte-ot kell átvinni?

Karakterbeszúrás

FLAG	Fejrész	Adatmező	Farok-rész	FLAG
------	---------	----------	------------	------

(a)



(b)

Ketdő- és végjelek bitbeszúrással

- ⬡ A karakterkódok hossza ne legyen része a keretezésnek
- ⬡ Tetszőleges számú bit legyen átvihető egy keretben
- ⬡ Flag byte: 01111110 (0x7E - ~)
- ⬡ Az adó minden ötödik 1-es után beszúr egy 0-t
- ⬡ A vevő minden ötödik 1-es után törli a követő 0-t

Ketdő- és végjelek bitbeszúrással

- ⬡ A jelző bájt is probléma nélkül átvihető
- ⬡ Adási oldal: 01111110 → 011111**0**10
- ⬡ Vételi oldal: 011111**0**10 → 01111110
- ⬡ Transzparens átvitel mindkét irányban

Ketdő- és végjelek bitbeszúrással

- Transzparens átvitel mindkét irányban
- Adatfolyam az adó oldalán: 01101111101111111111010101
- Átviteli vonal: 01101111101011111011111011010101
- Adatfolyam a vevő oldalán: 011011111011111111111111010101

Fizikai rétegbeli kódolássértés

- ⬡ Alkalmazható ha a fizikai réteg kódolása redundáns
- ⬡ 4B/5B
- ⬡ Például ahol egy bit kódolása is jelváltással történik
- ⬡ A vevő könnyen felismeri a bithatárokat
- ⬡ Nincsenek szinkronizációs problémák
- ⬡ A gyakorlatban több megoldás kombinációja használt

Forgalom szabályozás

- ⬡ A lassabb (terhelt) gép védelme elárasztás ellen
- ⬡ A túlterhelés keretek elvesztéséhez vezet
- ⬡ Visszacsatolás alapú forgalomszabályozás
- ⬡ A vevő tájékoztatja az adót a pillanatnyi állapotáról

Forgalom szabályozás

- ⬡ A vevő engedélyezi az adónak a további küldést
- ⬡ Az adatkapcsolati réteg jellemző megoldása
- ⬡ Sebesség alapú forgalomszabályozás
- ⬡ A protokoll tartalmazza a sebességkorlátozást
- ⬡ Ez minden adóra nézve kötelező érvényű

Hibakezelés

- ◊ Lehetséges problémák:
- ◊ Megérkezett-e minden keret?
- ◊ Minden keret csak egyszer érkezett-e meg?
- ◊ A keretek megfelelő sorrendben érkeztek-e meg?

Hibakezelés

- ⬡ Visszacsatolás szükséges a vevő felől
- ⬡ Pozitív és negatív nyugták
- ⬡ Ha nem érkezik meg a keret?
- ⬡ Időzítők alkalmazása
- ⬡ Keretek újraküldése
- ⬡ Többször megérkezhet ugyanaz a keret
- ⬡ Kimenő keretek sorszámozása



Fizikai rétegbeli kódolássértés

- ⬡ Alkalmazható ha a fizikai réteg kódolása redundáns
- ⬡ 4B/5B
- ⬡ Például ahol egy bit kódolása is jelváltással történik
- ⬡ A vevő könnyen felismeri a bithatárokat
- ⬡ Nincsenek szinkronizációs problémák
- ⬡ A gyakorlatban több megoldás kombinációja használt