ADATKAPCSOLATI RÉTEG

Az **adatkapcsolati réteg** az OSI hivatkozási modell második rétege. Itt a csatorna adategységei a keretek. A réteg alapvető feladata a hibamentes átvitel biztosítása a szomszéd gépek között, vagyis a hibás, zavart, tetszőlegesen kezdetleges átviteli vonalat hibamentessé transzformálja az összeköttetés fennállása alatt. Az adatokat adatkeretekké (data frame) tördeli, továbbítja, a nyugtát fogadja, hibajavítást és forgalomszabályozást végez. Két pont között a kommunikációs áramkörök hibáznak, véges az adatátviteli sebességük és késleltetést is okoznak.

Röviden:

- hálózati rétegnek nyújtott szolgáltatás
- nyugtázatlan összekötés nélküli szolgálat
- nyugtázott összekötés nélküli szolgálat
- nyugtázott összekötés alapú szolgálat
- keretezés (kezdet, vég)
- karakterszámlálás
- kezdő és végkarakterek
- kezdő és végbitek
- fizikai rétegbeli kódolásértés
- hibavédelem (error control)
- pontosan egyszeri megérkezés (időzítők, számlálók kezelése), ismétléssel javítás
- forgalom szabályozás (flow control)
- adó gyors, vevő lassú

Hibajelzés és javítás

Hibajavító kódolás: n = m + r (adat és ellenőrző bitek) Kódszavak Hamming távolság, d távolság (különbözőségek száma), d egybites hiba kell az egymásba való átmenethez: d hibát jelezni d+1 távolságú kód kell, d hibát javítani 2d+1 kód kell. 1 bites javító minta: (n+1) x 2m = 2n, n = m + r, (m+r+1) <= 2r, m=7, r=? 11<= 24 11. bitet az 1,2,8 bit ellenőriz Ellenőrző bitek: 1, 2, 4, 8 pozícióban, 3=1+2 , 5=1+4 , 11=1+2+8 ellenőrzőbit páros

Hibajelző kódok:

- Paritás bit, kereszt és hossz paritás bitek
- Polinom-kód (cyclic redundancy code, CRC)

M (X), rn G(X) generátor polinom foka, m+r, T(x)=M(x)+ Or(x) T(x) / G(x) = 0 8 bithez CRC-16 felismer minden egybites és kétbites hibát, minden páratlan számú hibás bitet tartalmazó hibát, valamint minden 16 vagy kevesebb bitnyi csoportos hibát, a 17 bites csoportok 99,997, a 18 vagy több bitesek 99,998 százalékát.

A hálózatok tervezői két alapvető stratégiát dolgoztak ki a hibák kezelésére. Az egyik módszer az, hogy minden elküldött adatblokkhoz annyi redundáns információt mellékelünk, amennyiből a vevő ki tudja következtetni, hogy mik voltak az eredetileg elküldött adatok. A másik módszerben csak annyi redundanciát iktatunk az adatok közé, amennyi a vevőnek lehetővé teszi, hogy a hiba tényét kikövetkeztesse. A vevő ebben az esetben nem tudja, milyen hiba történt, ezért újraküldést kér. Az előbbi stratégia hibajavító kódokat (errorcorrecting codes) használ, míg a másik hibajelző kódokat (error-detecting code). A hibajavító kódok használatát gyakran megelőző hibajavításnak (forward error correction) is nevezik.

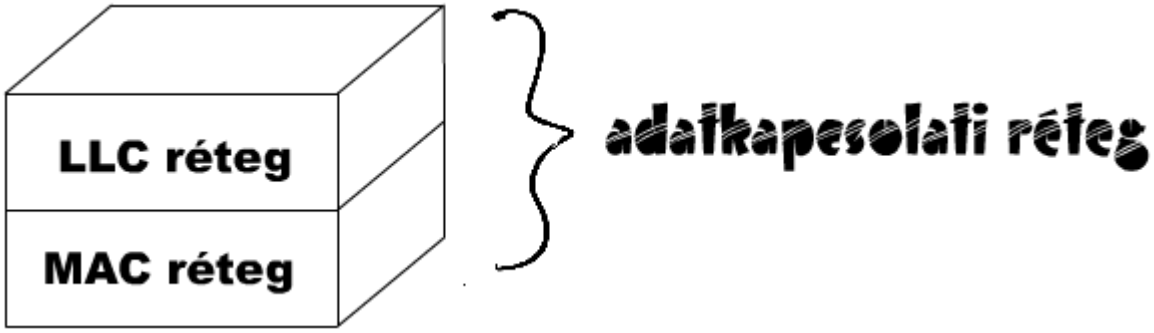
Mindkét módszernek megvan a saját alkalmazási területe. A fényvezető szálakon és más, nagymértékben megbízható csatornákon olcsóbb, ha hibajelző kódot használunk, és egyszerűen újraküldjük a ritkán előforduló hibás blokkokat. Ezzel szemben a vezeték nélküli összeköttetéseken, melyek sokat hibáznak, jobb, ha minden blokkba annyi redundanciát építünk, amennyiből a vevő már ki tudja találni, hogy mi volt az eredeti blokk. Az újraküldés ebben az esetben nem jó megoldás, mert az maga is hibás lehet.

Egyszerű példaként vegyünk egy kódot, melyben egy paritásbitet (parity bit) fűzünk az adatok végéhez. A paritásbitet úgy választjuk meg, hogy a bináris kódszóban levő 1-ek száma páros vagy páratlan legyen. Például, ha az 10110101-et egy bit hozzáfűzésével, páros paritással továbbítjuk, a kódja 101101011 lesz, míg az 10110001-ből 101100010 lesz páros paritással. Az ilyen kód egy bitnyi hiba jelzésére alkalmas.

Egyszerre megjelenő több bitnyi hiba jelzésére szolgál a polinom-kód (polynomial code), más néven ciklikus redundancia kód (cyclic redundancy code - CRC). Amikor ezt a polinom-kódok

alkalmazzuk, az adónak és a vevőnek előre meg kell egyeznie egy generátor polinomban, mely kigenerálja azt a redundáns részt, melyet hozzáfűzünk az eredeti kódszóhoz.

Az adatkapcsolati réteget szokásos két független alrétegre bontani.



Az alsót közeg hozzáférési (**Medium Access Control) alréteg**nek nevezzük. A MAC alréteg feladata a közeghez való hozzáférés, a kereteknek a kábelre való juttatása (az adási jog megszerzése és az adás).

A felsőt közeg a kapcsolatvezérlési (**Logical Link Control) alréteg**. Az LLC ellenőrzi a vett keretek épségét, kéri és végzi az újraküldést és szervezi a kapcsolatot. Mindezt természetesen a MAC réteg szolgáltatásainak (keret adása és vétele) felhasználásával.

Előzetesen összefoglalva, az adatkapcsolati réteg feladata abban áll, hogy biztosítsa azt, hogy az adó oldali adatok a vevő oldalra is adatként jussanak el, és ne legyen belőle értelmetlen jelek sorozata. Ezt úgy valósítja meg, hogy az adatokat egyértelműen azonosítható adatkeretbe tördeli szét, ellátja a szükséges vezérlőbitekkel, majd sorrendben továbbítja azokat. A vevő oldal pedig a kapott kereteket megfelelő sorrendben összeállítja. Az adó oldal ezenkívül még a vevő által küldött nyugtázásokat is feldolgozza. Mivel a fizikai réteg a biteket értelmezés nélkül továbbítja, ezért az adatkapcsolati réteg feladata, hogy meghatározza illetve felismerje a keretek határait. Így a felette elhelyezkedő réteg már hibáktól mentes adatokat kap (vétel esetén – adás esetén nyilván hibátlan adatokat ad...) Másik fontos feladata az, hogy a kétirányú átvitel esetén az esetleges ütközésekből adódó problémákat megoldja, és hogy forgalomszabályozást végezzen – tájékoztassa az adót a vevő fogadási szándékáról.

Ha kiragadjuk a Hálózati–Adatkapcsolati–Fizikai rétegeket, és csak ezek működésére koncentrálunk, akkor a működés a következő féleképpen modellezhető. Az adó oldal a hálózati rétegből kapott bitfolyamot az adatkapcsolati réteg diszkrét keretekké alakítja, melyeket ellenőrző

összegekkel lát el. Ezeket a kereteket alakítja bitfolyamból jelfolyammá a fizikai réteg, majd továbbítja a célállomás fizikai rétegének. A vevő oldal fizikai rétege fogadja, majd jelfolyamból bitfolyammá alakítja az adatokat. A vevő oldal adatkapcsolati rétege a keretek behatárolása és az ellenőrző összegek visszaellenőrzése után az így keletkezett bitfolyamot továbbítja a hálózati rétegnek. Ahhoz, hogy az adatkapcsolati réteg szolgáltatást nyújthasson a hálózati rétegnek, a fizikai réteg szolgáltatásait kell igénybe vennie.

Az adatkapcsolati réteg legjellemzőbb feladatainak felsorolása:

1. hálózati rétegnek nyújtott szolgáltatás
    a. nyugtázatlan összekötés nélküli szolgálat ilyen a megbízható csatorna alaphelyzetben az Ethernet vagy a kéretlen levél (nincs előzetes kapcsolat felépítés, kapcsolat lebontás)
    b. nyugtázott összekötés nélküli szolgálat ilyen a megbízhatatlan csatorna, a WLAN vagy a szöveges üzenetküldés (nincs előzetes kapcsolatépítés, de mivel minden keret nyugtázva van, az elveszett kereteket meg lehet ismételni)
    c. nyugtázott összekötés alapú szolgálat ez a legmegbízhatóbb átvitel, például a fájl átvitel) (a forrás és a cél az első fázisban felépíti az összeköttetést – inicializálódnak a megfelelő változók, számlálók, a második fázisban történik a keretek átvitele, a harmadik fázisban pedig az összeköttetés lebontása – és az erőforrások felszabadítása)
2. keretezés (kezdet, vég)
    a. karakterszámlálás, bájtszámlálás
    b. kezdő és végkarakterek beszúrása
    c. kezdő és végbitek beszúrása
    d. fizikai rétegbeli kódolás megsértése
3. hibakezelés, hibavédelem (Error Control)
    a. pontosan egyszeri megérkezés (időzítők, számlálók kezelése)
    b. ismétléssel történő javítás
4. forgalom szabályozás (Flow Control) gyors adó, lassú vevő helyzet kezelése

Keretezés

Az adatkapcsolati réteg legtipikusabb feladata a keretezés (nyilván adó oldalon a keretekbe tördelés, vevő oldalon a keretek eltávolítása – jellemzően az adó oldali funkciókat tárgyaljuk részletesen). A réteg alapegysége az adat keret (Data Frame). Önmagában a keretekre tördelés nem megoldás, szükség van (például) egy ellenőrző összegre, ami megmutatja, hogy az adott keret sérülésmentesen érkezett-e meg a vevő oldalra. Amennyiben az ellenőrző összegben eltérés van, akkor az egész keretet meg kell ismételni. A négy legáltalánosabban használt keretezési módszer:

1. karakterszámlálás, bájtszámlálás
2. kezdő és végkarakterek beszúrása
3. kezdő és végbitek beszúrása
4. fizikai rétegbeli kódolás megsértése

Karakterszámlálás, bájtszámlálás

Ez a keretezési módszer a keret hosszának, azaz a benne foglalt bájtok számának megfelelő adatot írja bele egy fejlécbe, az úgynevezett bájtszámmezőbe. Amikor a vevő oldal adatkapcsolati rétege megkapja az így képzett keretet, a bájtszámmezőből kiolvassa a keret hosszát (azaz bájtok számát). Ennek az algoritmusnak az a (potenciális) hibája, hogy az átviteli hiba szerencsétlen esetben pont a bájtszám mezőt érintheti, azaz ronthatja el. Így az átvitel kiesik a szinkronból, képtelenség lesz megtalálni a következő keret elejét (illetve végét). Ez a módszer ma már jellemzően nincs használatban.

Kezdő és végkarakterek beszúrása

Az előző megoldás szinkronizációs hibáját például elkerülhetjük olyan módon, hogy a keret elejét és a végét is egy-egy különleges jelzőbájttal (Flag) látjuk el. Létezett olyan megoldás, ahol a keret elejét, illetve végét különböző jelzőbájttal látták el, de a gyakorlatban az egyforma jelzőbájtok használata gyakoribb. Amennyiben az átvitel bármely okból is kiesne a szinkronból, akkor csak meg kell keresnünk a jelzőbájtot, és máris megtaláljuk az éppen átvitel alatt álló keret végét. Problémát az jelentheti, hogy hiába választunk speciális karaktert, bizonyos tartalmak esetében (pl. bináris átvitel, MP3, MKV, ZIP) a jelzésre választott speciális karakter bitmintája szerepelhet az átvitt adatban is. Egyik megoldás az, hogy egy extra ESC (Escape – kivétel bájt) bájtot használunk, szúrunk be pluszba a jelzőbájtunk elé. Ez a bájtbeszúrás (Byte Stuffing) módszere. Mi a megoldás

arra, hogy ha az ESC az átvitt adatfolyamban is szerepel? Be kell elé szúrni még egy ESC-t, így biztosak lehetünk abban, hogy a két ESC valójában egy ESC tartalommal bír. Ilyen módon abban is biztosak lehetünk, hogy önmagában csak a jelzőbájtunk előtt fog az ESC szerepelni. Az adatfolyamban így csak páros számú egymás utáni ESC bájtok fordulhatnak elő. Így egyértelműen – bár többszörös bájtbeszúrás árán, mert a vevő oldalon majd vissza kell állítani eredeti tartalmat – képesek vagyunk jelezni a keret határait, a keret hosszától függetlenül.

Kezdő és végbitek beszúrása

A bájtbeszúrás hasznos és működő megoldás, de nyilvánvaló, hogy sok „felesleges” adattal terheli az átviteli csatornát. Ezt a problémát orvosolja az eredetileg a HDLC (Highlevel Data Link Control / Magas Szintű Adatkapcsolati Vezérlés) protokollhoz kifejlesztett bitbeszúrás. Ez a módszer lehetővé teszi, hogy tetszőleges számú bit legyen a keretben, sőt, hogy a karakterkódok is tetszőleges számú bitből (ne csak 8 bitből) álljanak. A megoldás a következő módon épül fel. Minden keret egy speciális bitmintával indul, amit szintén jelzőbájtnak (Flag) nevezünk. Tartalmilag így néz ki: 01111110 azaz 6db egymást követő 1-es. Amikor az adó oldalon az adatkapcsolati réteg 5db egymást követő 1-est talál az adatok között, akkor automatikusan beszúr ezek után egy 0-át. Azaz 6db 1-es csakis a Flag-ben fordulhat elő az átalakítás után. Átvitelre már az így átalakított adat kerül. A vevő oldalon ugyanez történik csak fordítva, azaz minden egymást követő 5db 1- es után az adatkapcsolati réteg töröl 1db 0-át. A bitbeszúrásos módszerrel egyértelműen felismerhetők a kerethatárok. A szinkron elvesztése esetén meg kell keresni a 6db egymást követő 1-est, és így meg is találjuk a kerethatárokat, hiszen a 6db 1-es csak ott fordulhat elő. Ezt az eljárást használja az USB technológia is. Ezzel a módszerrel a „feleslegesen” átvitt adatok mennyisége jelentősen csökken, de nyilván számottevő mértékben megmarad.

Fizikai rétegbeli kódolás megsértése

Ez a megoldás a fizikai réteg kódolásainak tulajdonságait használja ki. Történetesen arról van szó, hogy bizonyos kódolások esetén léteznek olyan jelsorozatok, jelváltások, amelyek az adatátvitel során objektíve nem fordulhatnak elő. Így ezek felhasználhatóak jelzésekre, mivel az átvitt adattal semmiképpen sem téveszthetőek össze. Ilyen lehetőséget nyújt például a 4B/5B kódolás. Ez a megoldás 4 bites adatcsoportokat kódol 5 bites adatcsoportokba, NRZI kódolás mellett. A 32 lehetséges jelsorozatból viszont csak 16 jelsorozat van használatban az adatok kódolására, azaz a másik 16 jelsorozat felhasználható például a keret elejének illetve végének a

jelzésére. Az eljárást azért hívjuk kódsértésnek, mert olyan kód is átvitelre kerül, ami normál körülmények között nem kerülhetne átvitelre, nem használatosak kódolásra. Ezek a kódok, azaz jelsorozatok könnyen azonosíthatóak, és így a keretek beazonosításához nincs szükség az átvitt adatok módosítására (majd visszaalakítására). Összességében ez az eljárás terheli meg a legkevesebb „feleslegesen” átvitt adattal az adatátvitelt. Az legtöbb adatkapcsolati protokoll a gyakorlatban, a nagyobb biztonság érdekében (még ha ezzel jelentősen meg is növeli az átvitt adatok mennyiségét) a fenti módszerek valamely kombinációját alkalmazza. Például a keret az IEEE802.11 protokollban egy 72 bites, az IEEE802.3 protokollban egy 56 bites előtaggal (Preamble) kezdődik. Ez kellően hosszú ahhoz, hogy a vevő oldal fel tudjon készülni az adatok fogadására. Ezt követi még a fejlécben egy hosszúságot jelző kód (azaz a bájtszám). Így a keret tulajdonságai (kezdete és hossza) is többszörösen biztosítva van az átvitel során.

CSMA/CD

Ütközést jelző vivőérzékeléses többszörös hozzáférés (CSMA/CD)

A módszer angol elnevezése: Carrier Sense Multiple Access with Collision Detection = CSMA/CD. Ennél a módszernél, mielőtt egy állomás adatokat küldene, először "belehallgat" a csatornába, hogy megtudja, hogy van-e éppen olyan állomás amelyik használja a csatornát. Ha a csatorna "csendes", azaz egyik állomás sem használja, a "hallgatódzó" állomás elküldi az üzenetét. A vivőérzékelés (carrier sense) jelenti azt hogy az állomás adás előtt belehallgat a csatornába. Az állomás által küldött üzenet a csatornán keresztül minden állomáshoz eljut, és véve az üzenetet a bennfoglalt cím alapján eldöntheti, hogy az neki szólt (és ilyenkor feldolgozza), vagy pedig nem (és akkor eldobja).

Ennél a módszernél természetesen előfordulhat olyan eset, amikor egyszerre két vagy több állomás akarja használni a közeget. Az adás közben — mivel közben a csatornán lévő üzenetet veszi — el tudja dönteni, hogy az adott és a vett üzenetfolyam egyforma-e. Ha ezek különbözők, akkor azt jelenti, hogy valaki más is "beszél", azaz a küldött üzenet hibás, sérült. Ezt ütközésnek hívják, és ilyenkor az állomás megszakítja az üzenetküldést.

Az ütközés miatt kudarcot vallott állomások mindegyike az újabb adási kísérlet előtt bizonyos, véletlenszerűen megválasztott ideig várakozik. Ezek az idők a véletlenszerűség miatt eltérők, és a versengő állomások következő hozzáférési kísérlete során egy, a legrövidebb várakozási idejû fog

tudni adni, mivel a többiek a várakozási idejük leteltével adás előtt a csatornába belehallgatva azt már foglaltnak fogják érzékelni. Az e protokoll szerint működő állomások a következő három állapot valamelyikében lehetnek: versengés, átvitel, és tétlen állapot. Végiggondolva az eljárást, nyilvánvaló, hogy gyér forgalom esetén a közeghozzáférés nagyon gyors, mivel kevés állomás kíván a csatornán adni. Nagy hálózati forgalom esetén az átvitel lelassul, mivel a nagy csatorna terhelés miatt gyakoriak lesznek az ütközések. A széles körben elterjedt Ethernet hálózat ezt a módszert használja.

CSMA/CD – Carrier Sense Multiple Access/Collision Detection másképp

1.Az eszköz amelyik keretet szeretne küldeni, figyeli és várakozik arra hogy a média szabad legyen, vagyis senki ne adjon adatot rajta.

2.Amikor a média szabad, elkezdi az adást.

3.Ezzel egy időben figyeli, hogy esetlegesen történt-e ütközés a médián, mert épp egy másik eszköz is adni kezdett.

4.Ha ütközés történt, az összes éppen adatküldést kezdeményező készülék egy úgynevezett torlódás jelző jelet (jam signal) küld, hogy minden eszköz tudjon az ütközés tényéről.

5.Ezután minden eszköz mely adatot akar küldeni, egy véletlenszerűen kiválasztott ideig hallgat, mielőtt megkísérli újraküldeni az adatot.

6.Amikor ez az időzítője lejár, az 1. lépéstől kezdi a folyamatot.

CSMA/CD nem szünteti meg az ütközéseket, azonban biztosítja, hogy az Ethernet hálózat megfelelően működjön ütközések esetén is. Természetesen a CSMA/CD algoritmusnak teljesítménybeli vonzatai vannak. Minél nagyobb a hálózatunk, annál több ütközés fog történni, annál több ideig várnak az eszközök az ismétlésre, és az újabb ütközés lehetőségének az esélye is nagyobb. Minél több ütközés van a hálózatban, annál több véletlenszerűen kiválasztott ideig kell az eszközöknek várniuk, vagyis a hálózatot lassúnak fogjuk érzékelni. A CSMA/CD segít elkerülni az ütközéseket, de nem szünteti meg őket. Ahhoz hogy ütközés ne forduljon elő, egyszerre maximum egy eszköz adhat a szegmensen. Ennek következménye képen, az ugyanarra a HUB-ra csatlakoztatott eszközök osztoznak a HUB-on keresztül elérhető sávszélességen.

A logika, hogy az eszközöknek meg kell várniuk adás előtt hogy a média szabad legyen, a half-duplex mód. Ahhoz hogy az eszköz egyszerre tudjon fogadni is és adni is, egy másik eszköznek abban a pillanatban is adnia kell a szegmensen, ebben az esetben azonban ütközés történik.

Más szavakkal a half-duplex módot jellemezhetjük úgy is hogy:

- az eszköz nem képes egyszerre adni és fogadni
- az eszköznek meg kell várnia még a többi eszköz befejezi az adást és a média szabad lesz

Ütközés esetén az eszközök egy véletlenszerű ideig várakoznak, majd utána kísérlik meg az adást újra. Ütközés alatt az adatok elvesznek, illetve ezalatt hasznos adat nem közölhető a vezetéken.
Lényegesebb teljesítményromlást már 30%-os hálózat kihasználtság esetén is érezni lehet.

Ennek a problémának a megoldása, hogy HUB-jainkat Switchek-re cseréljük.

A switchek úgynevezett mikroáramköröket hoznak létre a az eszközök között, melyet úgy képzelhetünk el, mint ideiglenesen létrehozott, dedikált privát folyosókat két lakás között.

Mivel ez esetben a adás-vétel két külön kábelen folyik, ütközés nem fordulhat elő. Az egyik kábelen a készülék küldeni tudja az adatot, a másikon pedig venni. A mikroáramkörök gyakorlatilag pedig csak két eszköz közt jönnek létre.

CSMA/CA – Carrier Sense Multiple Access/Collision Avoidance

CSMA/CA hasonló a testvéréhez a CSMA/CD-hez. Ezt a technológiát azonban olyan helyeken szokták alkalmazni, ahol a média nem alkalmas arra hogy egyszerre folytasson a készülék adatátvitelt és fogadást is, illetve ahol a készülékek nem rendelkeznek teljes lefedettséggel egymást illetően (A és B, B és C látja egymást, de A és C túl messze van egymástól hogy érzékeljék ha a másik adást folytat).Leggyakoribb alkalmazási területe a WiFi hálózatok.

A CSMA/CA kommunikáció az alábbi lépésekből áll:

1. Az eszköz amelyik keretet szeretne küldeni, figyeli és várakozik arra hogy a média szabad legyen, vagyis senki ne adjon adatot rajta.
2. Amikor szabad, egy véletlenszerűen kiválasztott ideig vár, hogy csökkentse az esélyét annak hogy két vagy több eszköz egyszerre kezd el adni.

3.Ismételten belehallgat a médiába, és amennyiben az még mindig szabad, elkezdi az adást.

4.Miután a adatot elküldte, nyugtázásra vár.

5.Ha a nyugtázás nem érkezik meg, az adatokat újra küldi az egyes lépéstől.

Az Ethernet (802.3) család

Az Ethernet egy állomása a közvetítő közeggel (kábel) való állandó kapcsolatot kihasználva bele tud hallgatni a csatornába, így ki tudja várni, amíg a csatorna felszabadul, és a saját üzenetét leadhatja anélkül, hogy ezzel más üzenet sérüljön, tehát a torlódás elkerülhető. A csatornát az állomások folyamatosan figyelik, ha ütközést tapasztalnak, akkor zavarni kezdik a csatornát, hogy figyelmeztessék a küldőket, ezután véletlen ideig várnak, majd adni kezdenek. Ha ezek után további ütközések történnek, az eljárás ugyanez, de a véletlenszerű várakozás idejét kétszeresére növelik, így időben szétszórják a versenyhelyzeteket, esélyt adva arra, hogy valaki adni tudjon.

A DIX Ethernet és a 802.3 Ethernet adatkereteket küld a kábelen. Az adatkeretek kicsit eltérő formátumúak a két Ethernet esetében, de mindkét Ethernet a Manchester kódolást használja. A token ring (802.5) az adatkeretek kódolásánál a manchester kódolás egy változatát, a differenciál Manchester kódolást használja.

A klasszikus Ethernet

A megnevezés első száma az átviteli sebességet jelöli, az ezt követő Base az alapsávú átvitelre utal. A következő szám, koaxiális kábel esetén a kábel hosszát adja meg 100 méteres egységekre kerekítve. A klasszikus Ethernet kábelek leggyakoribb típusai:

Megnevezés	Kábel	Max. szegmenshossz	Csomópont/szegmens	Megjegyzés
10Base5	vastag koaxiális	500 m	100	Eredeti kábel, mára idejétmúlta
10Base2	vékony koaxiális	185 m	30	Nincs szükség elosztóra
10Base-T	sodrott érpár	100 m	1024	A legolcsóbb rendszer
10Base-F	optikai	2000 m	1024	Épületek között

A 10Base5

A vastag Ethernet (thick Ethernet) esetében a kábel egy sárga kerti öntözőcsőre emlékeztet, amelyen a lehetséges csatlakozási pontok 2,5 méterenként meg vannak jelölve. (A szabvány a sárga színt nem írja elő, de javasolja.)

Egy állomás csatlakoztatása úgynevezett vámpír csatlakozóval történik, ahol egy vékony tüskét nyomnak a kábelbe, amíg a koaxiális kábel központi vezetékét el nem érik. A vámpír csatlakozó közvetlenül kapcsolódik egy adó-vevő egységhez, ami egy speciális kábelen keresztül csatlakozik a számítógépben lévő csatolókártyához.

A forgalmazás 10 Mb/s-os alapsávú (baseband) jelekkel történik. Létezett a széles-sávú változat, a 10Board36, de idővel eltűnt a piacról.

A 10Base2

A vékony Ethernet (thin Ethernet) estében a kábel jobban hajlítható, vékonyabb, és gyári BNC csatlakozókat és T elosztókat használ. A BNC csatlakozókról a jel egy koaxiális kábelen keresztül jut a számítógép csatolókártyájához – a kábel hossza korlátozott – a csatolókártya tartalmazza a szükséges adó-vevő áramköröket is.

A 10Base-T

Az irodai környezetben szokásos, csavart érpárokat használó megoldás, a számítógépek közvetlenül egy elosztóhoz csatlakoznak.

A 10Base-F

Optikai csatolás van az egységek között, ezért mind biztonsági szempontból, mind zavarvédelmi szempontból kedvezőbb az előbbieknél, viszont jóval drágább. Tipikusan épületek közötti kapcsolat kiépítéséhez használatos.

A gyors Ethernet (802.3u)

1992-ben összehívták a bizottságot, hogy készítsenek egy új szabványt egy gyorsabb LAN-ra, megtartva a 802.3 minden egyéb előírását. Egy másik elképzelés szerint viszont teljesen át kell mindent alakítani, biztosítani kell a valós idejű forgalmat, valamint a digitális hangátvitelt. A nevet – üzleti okokból – meg akarták tartani. A bizottság az első változatot fogadta el, és elkészítette a 802.3u-t. Az el nem fogadott javaslat hívei elkészítették a saját szabványukat, a 802.12-t, ami nem terjedt el. A gyors Ethernet eredeti kábelezése:

Megnevezés	Kábel	Max. szegmenshossz	Megjegyzés
100Base-T4	sodrott érpár	100 m	3-as kategóriájú UTP
100Base-TX	sodrott érpár	100 m	Duplex 100Mb/s (5.kat. UTP)
100Base-FX	fényvezető szál	2000 m	Nagy távolságra, duplex 100Mb/s

•

A gigabites Ethernet (802.3.z)

A gyors Ethernet szabványt követően 1995-ben a 802-es bizottság egy még gyorsabb Ethernet tervein kezdett dolgozni. A célkitűzések a következők voltak: 10x gyorsabb sebesség, kompatibilitás az eddigi Ethernetekkel. A végső szabvány, a 802.3z eleget tett a feltételeknek.

A gigabites Ethernet – eltérően a klasszikus Ethernet-től – pont-pont felépítésű. A legegyszerűbb topológiánál a két számítógép van gigabites Ethernettel összekapcsolva. Gyakoribb az a megoldás, amikor egy kapcsoló vagy elosztó köt össze több számítógépet, vagy további elosztókat vagy további kapcsolókat. Minden esetben egy Ethernet kábel végén pontosan egy-egy eszköz található csak.

A gigabites Ethernet két működési módot támogat: a duplex és félduplex működést. "Normális" esetnek a duplex módot tekintik, a forgalom mindkét irányban egy időben folyhat. Ezt akkor használják, ha egy központi kapcsolót vagy a periférián lévő gépekkel, vagy más kapcsolókkal kötnek össze. Ekkor minden adatot pufferelnek, így bármelyik gép és kapcsoló tetszés szerinti időben küldheti el az adatait (kereteit). Az adónak nem kell figyelnie a csatorna forgalmát, mert a versengés kizárt. Mivel a kábel egy gépet és egy kapcsolót köt össze, csak ez a gép adhat a kapcsoló felé, a duplex megoldás miatt az esetleges ellenirányú adatküldés biztosan sikeres lesz. Nincs tehát versengés, a Csma/cd protokoll használata felesleges, a maximális kábelhosszt a jel erőssége

határozza meg, nem pedig egy zajlöket adóhoz való visszajutás ideje. A kapcsolóknak módjukban áll keverni és egyeztetni a sebességeket, és automatikusan konfigurálhatják a hálózatot is, hasonlóan a gyors Ethernethez.

Ha a számítógépek nem kapcsolóhoz, hanem elosztóhoz kapcsolódnak, akkor a félduplex módot használják. Az elosztó nem puffereli a beérkező kereteket. A kapcsoló belül villamosan összeköti az összes vonalat, hasonlóan a klasszikus Ethernetnél alkalmazott megoldáshoz. Az ütközések nem kizárhatók, tehát szükséges a Csma/cd protokoll használata. Mivel a legrövidebb keretet (64 byte) 100-szor gyorsabban lehet elküldeni, a maximális távolság is 100-szor kisebb, azaz 25 méteres lesz, hogy megmaradjon az a sajátosság, hogy az adás még a legrosszabb esetben is addig tart, amíg a zajlöket visszaér az adóhoz. Egy 2500 méter hosszú kábel esetében, 1 Gb/s sebesség mellett az adó már régen végzett egy 64 byte-os keret adásával, amikor a keret a kábel hosszának tizedét sem tette még meg (a visszaútról nem is beszélve).

A bizottság – érthetően – elfogadhatatlannak tartotta a 25 méteres távolságot, ezért bevezette a vivőkiterjesztés (carrier extension) és a keretfűzés (frame bursting) funkciókat. Így a kábelhossz 200 méterre kiterjeszthető.

Ellentmondásos, hogy egy szervezet, amelyik a gigabites Ethernet megvalósítása mellett dönt, de elosztókkal köti össze a gépeket, ezzel tulajdonképen a klasszikus Ethernetet szimulálja. Tény, hogy az elosztók valamivel olcsóbbak a kapcsolóknál, és a gigabites Ethernet illesztőkártyák elég drágák. Ha ezt a relatív drágaságot az olcsóbb elosztókkal akarja a szervezet ellensúlyozni, akkor ezzel egyben a hálózata teljesítményét drasztikusan csökkenti. A bizottságnak viszont a kitűzött visszafelé kompatibilitás előírása – és szokás – miatt a 802.3z szabványban meg kellett engednie ezt a lehetőséget.

A gigabites Ethernet kábelezése:

Megnevezés	Kábel	Max. szegmenshossz	Megjegyzés
1000Base-SX	fényvezető szál	550 m	Többmódusú fényvezető szál (50 vagy 62,5 mikron)
1000Base-LX	fényvezető szál	5000 m	Egy- vagy többmódusú fényvezető szál (50 vagy 62,5 mikron)

1000Base-CX	2 pár STP	25 m	Árnyékolt, sodrott érpár
1000Base-T	4 pár UTP	100 m	Szabványos 5-ös kategóriájú UTP

Klasszikus Ethernet keret felépítése

1. Az előtag (preamble)váltakozva tartalmaz egyeseket és nullákat. 7 darab 10101010 tartalmú bájtból álló sorozat. A 10 Mbit/s-os és kisebb sebességű Ethernet-megvalósításoknál az órajel szinkronizálása ennek a mezőnek a segítségével történik. Az Ethernet gyorsabb változatai szinkron működésűek, ezeknél időzítési információkra nincs szükség; ennek ellenére, a kompatibilitás érdekében a mező megmaradt.

2. Az előtagot egy egyoktetből álló mező a keretkezdő (start frame delimiter) követi, amely az időzítési információk végét, a keret tényleges kezdetét jelzi. Tartalma az 10101011 bitsorozat.

3. Ezután a cél (destination) és küldő (source) állomás 48-bites címei következnek. Az Ethernet hálózaton minden állomást egy egyedi, 48-bites (6 bájtos) ún. MAC (Media Access Control) cím azonosít. Ezen címek kiosztását az IEEE kontrollálja.

4. A hossz/típus mezőt kétféle célra lehet használni. Ha értéke a decimális 1536-nál, vagyis a hexadecimális 0×600-nál kisebb, akkor a benne szereplő érték hosszt ad meg, egyébként típus értékként azt adja meg, hogy az Ethernet folyamatainak lezárulása után melyik felsőbb rétegbeli protokoll fogja kapni az adatokat. A hossz a mezőt követő adatrészben található bájtok számát adja meg.

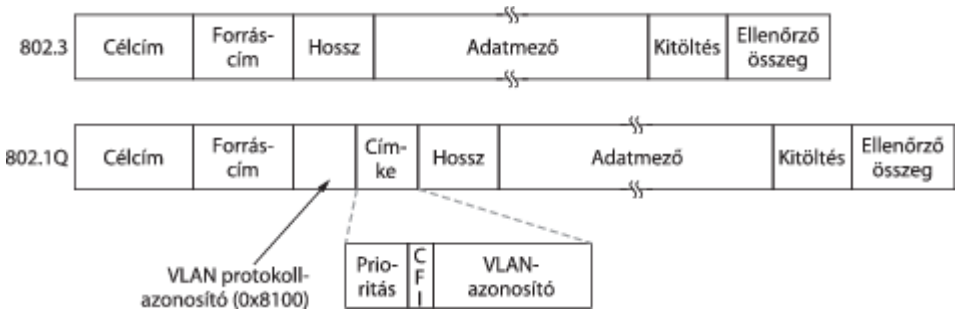
5. Az adat mező és a szükség szerinti kitöltés hossza tetszőleges lehet, azonban a keret mérete nem haladhatja meg a felső mérethatárt. A maximális átviteli egység (maximum transmission unit, MTU) az Ethernet esetében 1500 oktett, az adatok mérete tehát ezt nem haladhatja meg. A mező tartalma nincs meghatározva. Ha nincs elég felhasználói adat ahhoz, hogy a keret mérete elérje a minimális kerethosszt, akkor előre meg nem határozott mennyiségű adat kerül beillesztésre, közvetlenül a felhasználói adatok mögé. Ezt a többletadatot nevezzük kitöltésnek. Az Ethernet keretek hosszának 64 és 1518 oktett között kell lennie.

6. A keret végén szereplő FCS (Frame Check Sequence - Keret Ellenőrző Sorozat) mezőben egy 4 bájton CRC ellenőrző összeg helyezkedik el. Ha a vevő által számolt és a keretben lévő összeg nem egyezik, a keret eldobásra kerül.

8BYTE	6BYTE	6BYTE	2BYTE	46-1500BYTE	4BYTE
Előtag / Preamble	CÉL MAC CÍME	FORRÁS MAC CÍME	Típus/ Hossz	Adatok	CRC

VLAN-os Ethernet keret 802.1q

A keretben az egyetlen változást két darab 2 bájtos mező hozzáadása jelenti. Az első a ***VLAN protokollazonosító (VLAN protocol ID)***, melynek értéke mindig 0x8100. Ez a szám nagyobb 1500-nál, ezért az Ethernet-kártyák nem hosszként, hanem típusként értelmezik. Az, hogy mit tesz egy hagyományos kártya egy ilyen kerettel, bizonytalan, mivel az ilyen keretek elküldését a hagyományos kártyák nem támogatják.



A (hagyományos) 802.3 és a 802.1Q keretformátumok

A második 2 bájtos mező három almezőt tartalmaz. Ezek közül a legfontosabb a ***VLAN-azonosító (VLAN identifier)***, ami az alsó 12 bitet foglalja el. Erről szól az egész történet: ez adja meg a VLAN színét, amelyikhez a keret tartozik. A 3 bites ***Prioritás (Priority)*** mezőnek semmi köze a VLAN-okhoz, de mivel egy évtizedben úgyis csak egyszer módosítják az Ethernet-fejrészt, és ahhoz is három év és száz ember kell, akkor, ha már egyszer nekiálltak, miért ne raknának bele még valami jó dolgot. Ez a mező lehetővé teszi a szigorú és kevésbé szigorú követelményeket támasztó valós idejű, valamint a nem időérzékeny forgalmak megkülönböztetését, hogy jobb szolgáltatásminőséget lehessen elérni az Etherneten. Erre az Etherneten keresztül történő

beszédátvitelnél van szükség (bár az igazat megvallva, az IP-nek is van egy hasonló mezője immár negyed százada, és azt sem használta soha senki).

Az utolsó mezőt igazából nem is ***CFI***-nek ***(Canonical Format Indicator – kanonikus formátumjelző)***, hanem ***CEI***-nek ***(Corporate Ego Indicator – vállalati érdekérvényesítés-jelző)*** kellene nevezni. A bitet eredetileg arra szánták, hogy megkülönböztessék vele a bitek sorrendjét a MAC-címekben (alsóvég vagy felsővég kódolású), de ez a használat valahogy elsikkadt a viták során. Jelenléte ma már csak arra utal, hogy az adatmező egy nem módosítható 802.5-ös keretet tartalmaz, ami reményei szerint egy másik 802.5-ös LAN-t talál a céljánál, miközben a két hálózat között egy Etherneten halad át. Természetesen ennek az egész elrendezésnek semmi köze nincs a VLAN-okhoz. De hát a szabványosítási bizottságokban is csak úgy működik a politika, mint máshol: ha te megszavazod az én bitemet, én is megszavazom a tiédet.

Mint már említettük, ha egy címkézett keret érkezik egy VLAN-képes kapcsolóhoz, akkor a kapcsoló a VLAN-azonosító alapján keresi ki a táblázatból, hogy melyik porton kell a keretet kiküldeni. De honnan jön a táblázat? Ha kézzel kell összeállítani, mint annak idején a kézi konfigurációs hidakat, akkor ott vagyunk, ahol a part szakad. Az átlátszó hidak szépsége éppen abban van, hogy csatlakoztatás után rögtön működnek, és nem igényelnek semmilyen kézi beállítást. Nagy szégyen lenne elveszíteni ezt a tulajdonságot. Szerencsére a VLAN-képes hidak is képesek automatikusan konfigurálni magukat az elhaladó címkék megfigyelése alapján. Ha például egy 4-es VLAN-címkét hordozó keret a 3-as porton érkezik be, akkor a 3-as porton lévő egyik gép nyilvánvalóan a 4-es VLAN tagja. A 802.1Q szabvány, mely többnyire a 802.1D szabvány megfelelő részeire hivatkozik, kifejti, hogyan kell dinamikusan felépíteni a táblázatokat.

Mielőtt elhagynánk a VLAN-ok útválasztásának témáját, érdemes egy utolsó megfigyelést tennünk. Az internet és az Ethernet világában sokan fanatikus hívei az összeköttetés nélküli hálózatoknak, és hevesen elleneznek mindent, ami egy kicsit is emlékeztet az összeköttetésekre az adatkapcsolati vagy a hálózati rétegekben. A VLAN mégis valami olyasmit vezetett be, ami meglepően hasonlít az összeköttetésekhez. Egy helyesen működő VLAN-ban ugyanis minden keret egy új, speciális azonosítót hordoz, amit a kapcsoló táblázatában arra használnak, hogy kikeressék, hová kell küldeni a keretet. Pontosan ez az, ami az összeköttetés-alapú hálózatokban is történik. Az összeköttetés nélküli hálózatokban a célcím alapján végzik az útválasztást, nem

pedig valamiféle összeköttetés-azonosító alapján. Az effajta, mélyben meglapuló összeköttetés-elvűségről az 5. fejezetben még többet is olvashatunk.

Felhasznált források:

1. 10. fejezet – Adatkapcsolati réteg <online> <http://rs1.sze.hu/~paalda/oktat/szg_hal/Dave/10_Az%20adatkapcsolati%20r%E9teg.pdf> <2016. 12.11.>
2. CSMA CD <online><http://juhaasztamaas.uw.hu/12F1_prelm/halo_CSMA_CD.html><2016. 12.11.>
3. CSMA CD ésCA <online><http://www.ciscoworld.hu/csmacd-es-csmaca/><2016.12.11.>
4. Ethernet wiki <online><https://hu.wikipedia.org/wiki/Ethernet><2016. 12.11.>
5. Az IEEE 802.1q szabvány <online><https://gyires.inf.unideb.hu/GyBITT/30/ch04s09.html><2016. 12.11.>