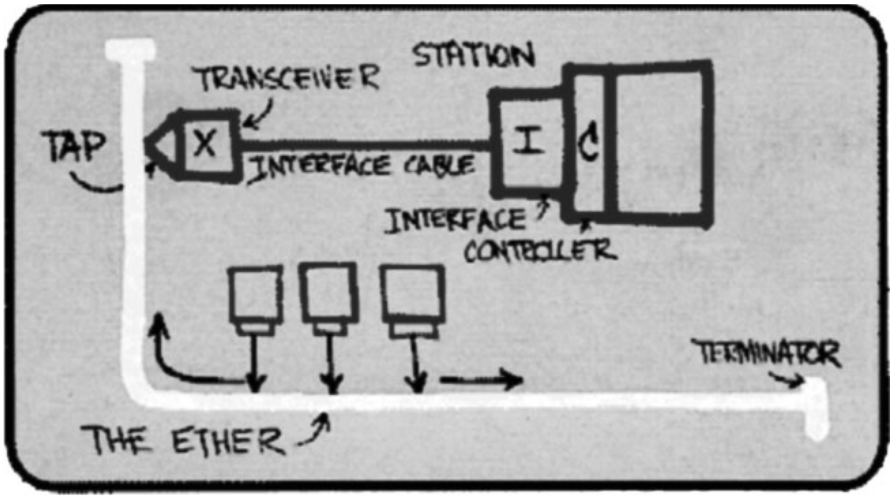
ÚT AZ IPARI ETHERNETIG

Dr. Ferenczi István, villamosmérnök, Nyíregyházi Egyetem,
e-mail: ferenczi.istvan@nye.hu

1. Az Ethernet kialakulása és fejlődése

Már lassan négy évtizede annak, hogy 1976 nyarán a Xerox cég mérnöke, Robert Metcalf egy számítógépes konferencián (National Computer Conference) bemutatta az első Ethernet hálózatra vonatkozó elképzelésének vázlatát (1. ábra). A hetvenes évek elején a Xerox cég mérnökei egy vezetékes hálózat kifejlesztését tűzték ki célul. Azt szerették volna megvalósítani, hogy a Xerox Alto számítógépeket összekapcsolják egymással, a szerverrel és a lézernyomtatóval. Így lehetségessé válhatott, hogy egy nyomtatóra, több, akár nagyobb távolságra lévő számítógépről is lehetett nyomtatni. Az átviteli közegként a két végén impedancia illesztéssel lezárt vastag koaxiális kábelt használtak. Mindegyik állomást hálózati interfész vezérlővel láttak el, amely az illesztőn (transceiver) keresztül kapcsolódott a közeghez. Az interfész órajelét az Alto órajeléből származtatták, amely már egy igen jelentős 2,94 Mb/s-os bitsebességet tudott biztosítani.



1. ábra. A Metcalf által bemutatott Ethernet vázlat
(forrás: http://www.ieee802.org/3/ethernet_diag.html)

Az Alto Aloha Network nevet viselő kísérleti hálózat jól vizsgázott, így 1973-ban Metcalf el is nevezte a rendszert **Ethernet**nek (éterhálónak), arról az anyagról, amelyet az elektromágneses hullámok terjedési közegének véltek korábban.

A Xerox féle Ethernet olyan sikeres volt, hogy 1978-ban szabadalmaztatták. Röviddel ezután Robert Metcalf kilépett a Xeroxtól, új céget alapított és 1979-ben sikerült meggyőznie három másik nagy céget (Digital Equipment Corporation, Intel és Xerox) a szabadalom támogatására, hogy elfogadtathassák általános szabványként. Így született meg 1982-ben az első igazi, 10 Mbit/s-os Ethernet szabvány az Ethernet II vagy más néven DIX V2.0. Kisebb

változtatások után a szabványt az IEEE (Institute of Electrical and Electronics Engineering) is elfogadta és 1983-ban kidolgozta a máig is használatos Ethernet hálózati szabvány alapját az IEEE 802.3-at [1].

A szabványnak azóta számos változata jelent meg. Az eredeti 10Base5 nevű, vastag koaxiális kábeles szabványt (Thicknet) rövidesen (1985-ben) követte a 10Base2, ugyancsak 10 Mbit/s-os vékony koaxiális kábeles (Thinnet) szabvány, amely a kábel viszonylag olcsó ára miatt nagyon gyorsan elterjedt és vált népszerűvé főleg az irodai hálózatok kialakításában. Az áttörést az 1990-ben megjelenő csavart érpáros 10Base-T szabvány jelentette, amely már jóval előnyösebb tulajdonságokkal rendelkezett mint a koaxiális kábeles elődje.

A Fast Ethernet története 1993 júniusában kezdődött, amikor több mint 50 fejlesztő és szolgáltató fogott össze egy közös cél érdekében, hogy kidolgozzák a 100 Mbit/s-os Ethernetet. Az eredmény 1995-ben vált teljessé, amikor megjelent az IEEE 802.3u 100Base-TX CAT5-ös csavart érpáros szabvány, amelyet még napjainkban is igen elterjedten használnak szinte minden területen. A következő jelentősebb lépés 1999-ben történt az IEEE 802.3ab Gigabit Ethernetes szabvány megjelenésével, majd ezt követi 2003-ban 10 Gigabit Ethernet. És a fejlődés nem áll meg…

A történelem itt is megismétli önmagát. A 90-es évek elején kezdtek megjelenni az első vezeték nélküli adatgyűjtő terminálok. Ezek a rendszerek nagyrészt egyediek voltak, mindegyik gyártó a saját elgondolása szerint valósította meg a kapcsolatot. Ez azt jelentette, hogy egy mobil adatgyűjtő csak a saját gyártójának eszközeivel kiépített hálózatában működött, a másikében nem. Ennek áthidalására fejlesztette ki az IEEE munkacsoport az évtized közepére az első szabványos WLAN protokollt, az IEEE 802.11-es családot, amelyet 1997-ben véglegesítettek és ratifikáltak. A 802.11 protokoll logikája és az eszközök együttműködését garantáló Wi-Fi a nagysikerű és szinte kizárólagossá vált 802.3 Ethernet protokoll működésén alapult. Az első 802.11 szabvány három fizikai réteget definiált: egyet az infravörös tartományban, kettőt pedig a szabad 2.4 GHz-es tartományban. A szabadon felhasználható frekvencia tartományban az adási teljesítmény nem haladhatja meg a 100 mW értéket. Az adatátviteli sebesség induláskor 1 Mb/s volt, ami később 2 Mb/s-ra nőtt. Mindössze két év telt el, és 1999-ben két idevonatkozó szabvány is megjelent. A 802.11b szabvány az előző továbbfejlesztett változata volt. Az adatátviteli sebesség egy új modulációs technikának köszönhetően 11 Mb/s-ra növekedett. A másik megoldás az 5 GHz-es tartományban működő 802.11a protokoll volt, amely már ekkor 54 Mb/s-os sebességet produkált. Ez azonban, az 5 GHz-es korlátozott hozzáférésű (főleg katonai radarok által használt) frekvencia tartomány miatt nem tudott igazán elterjedni.

2003 folyamán megjelent a 802.11g szabvány, amely a 802.11a modulációs eljárását alkalmazva megsokszorozta a 802.11b rendszerek sebességértékét, elérve ebben a frekvenciasávban is az 54 Mb/s-os értéket. Az új szabvány megtartotta kompatibilitását az előzővel, vagyis a 802.11g eszközök automatikusan felismerik és hozzákapcsolódnak a 802.11b elérési pontokhoz, és fordítva, a régi 802.11b eszközöket felismerik az új 802.11g cellák.

Az adattovábbítás sebessége az évek során tovább nőtt. A 2009 szeptemberében ratifikált 802.11n rendszer 20 MHz-es sávszélesség mellett, akár 300 Mb/s sebességet tud biztosítani az éter hullámain. A vezeték nélküli világ egyre nagyobb teret hódít az ipari kommunikációs rendszerek területén is. Lehetővé vált olyan helyeken is érzékelőket és távadókat elhelyezni, ahol egyébként a vezetékek használata miatt ez eddig lehetetlen volt vagy csak nagyon nehezen és költséges megoldások mellett lehetett megvalósítani.

2. Adattovábbítás Etherneten keresztül

Ahhoz, hogy az Ethernet hálózaton keresztül az adatok egyik állomástól a másikig zavartalanul és biztonságosan eljussanak, számos bonyolult részfeladatot kell megoldania a küldőnek és a fogadónak is egyaránt. Ráadásul ezeknek a részfeladatoknak a lebonyolítása során, sok esetben együttműködés is szükséges. Azért, hogy ezeket a feladatokat a különböző gyártók által készített interfészek mindegyike kezelni tudja, szabványra volt szükség.

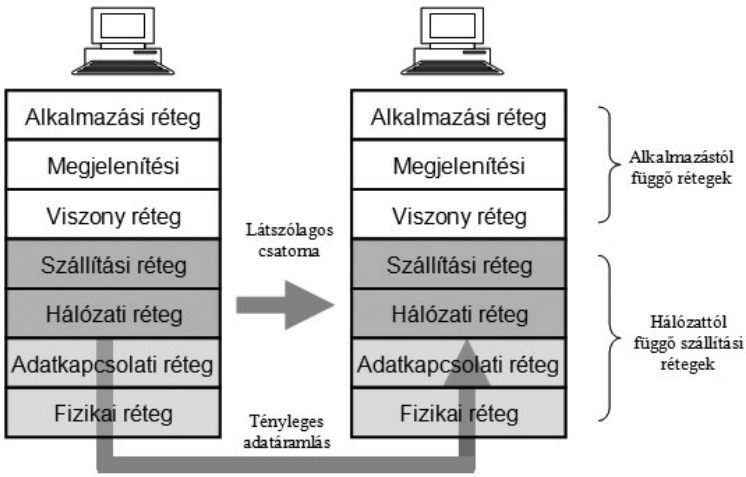
2.1. Az OSI modell

A Nemzetközi Szabványügyi Szervezet (ISO) volt az, amely vállalkozott erre a feladatra. Olyan elképzelést dolgoztak ki, mely szerint a hálózati kommunikációt rétegekre bontják, ezen belül pontosan definiálják az egyes rétegek feladatát, illetve a rétegek mentén zajló kommunikációt. A szabványt kicsit megkésve, 1984-ben bocsátották ki és ISO OSI (Open System Interconnection) modellnek nevezték el, mivel a nyílt rendszerek kommunikációjával és összekapcsolásának részleteivel foglalkozik.

Az OSI modell a különböző protokollok által nyújtott funkciókat egymásra épülő rétegekbe sorolja. Minden réteg csak és kizárólag az alsóbb rétegek által nyújtott funkciókra támaszkodhat, és az általa megvalósított funkciókat pedig csak a felette lévő réteg számára nyújthatja. Ez a modell alapjaiban meghatározó volt a hálózatokkal foglalkozó ipar számára. A kezdeményezés legfontosabb eredménye az volt, hogy meghatározták azokat a specifikációkat, amelyek pontosan leírták, hogyan léphet egy réteg kapcsolatba egy másik, azonos szintű réteggel. Ez gyakorlatilag azt jelenti, hogy egy adott gyártó által készített réteg programja együtt tud működni egy másik gyártó által készített programmal. A végeredmény egy olyan rendszer lett, mely két alapelvből állt össze:

- kialakult a 7 rétegből álló struktúra, (2. ábra),
- meghatározták az egyes résztevékenységeket ellátó protokollokat.

A modellt az IEEE is elfogadta, beépítette és erre alapozta az IEEE 802.3 szabványsorozatát [1].



2. ábra. Az információ OSI modell szerinti látszólagos és tényleges áramlási útja

A 2. ábrán látható rétegek funkcióiról csak röviden tennék említést, ugyanis az ipari Ethernetes rendszereknél, csak néhány, főleg az alsóbb rétegek játszanak meghatározó szerepet. Fentről lefelé haladva ezek a következők:

7. Alkalmazási réteg (Application layer): a felhasználói programok számára biztosítja a hálózati kommunikációt. Ide tartoznak azok a szabványosított protokollok is, mint például a HTTP, FTP, Telnet, SMTP.

6. Megjelenítési réteg (Presentation layer): foglalkozik az adatok szintaktikájával és szemantikájával, azért, hogy a különböző adatábrázolási kódokat alkalmazó rendszerek is kommunikálni tudjanak egymással. Az adatokat átalakítja olyan formátumokká, amelyeket az alkalmazások már közvetlenül tudnak értelmezni (MIME, XML, stb.) Az adatkonverziókon túl, ez a réteg képes az adatok tömörítésére és esetleges titkosítására is.

5. Viszony réteg (Session layer): Lehetővé teszi a kapcsolati viszony létesítését két számítógép között. Vagyis, hogy két állomás között kiépül a kommunikációs kapcsolat, adatok jutnak el egyik géptől a másikig (szállítási réteg feladata), majd az adatáramlás befejezésével a kapcsolat megszűnik. Háromféle kommunikációs módot tesz lehetővé: szimplex (csak egyirányú), half-duplex (kétirányú, de egy időben csak egy irány aktív) és full-duplex (kétirányú, egyidejűleg történő kommunikáció).

4. Szállítási réteg (Transport layer): Adatokat fogad a felső rétegtől, kisebb darabokra vágja szét, átadja a hálózati rétegnek, és gondoskodik róla, hogy az adatcsomagok megbízhatóan és hibamentesen eljussanak az egyik állomástól a másikig. Ezt úgy valósítja meg, hogy kapcsolatot teremt a hálózati eszközök között, nyugtázza a csomagok kézbesítését és újra elküldi az elveszett vagy sérült csomagokat. Küldéskor a szállítási réteg a nagyméretű adatcsomagokat kisebb csomagokká darabolja a hatékonyabb szállítás érdekében. A fogadó számítógép ezeket a csomagokat összeilleszti, és megvizsgálja, hogy minden adatcsomag megérkezett-e? Ezen az elven működik a TCP protokoll [5]. A TCP összeköttetés elsődleges célja a biztonságos szállítás. Feladata, hogy a kapott szelvényeket megfelelő sorrendben rakja össze az eredeti üzenetnek megfelelően. Küldéskor, az adatátvitel hibátlanságának biztosítása érdekében kézfogásos (handshaking) módszert, úgynevezett pozitív nyugtát használ. Az adó elküldi az adatot, közben elindít egy időzítőt is és várakozik a nyugtázásra. Ha nem kap választ az időzítés lejárta előtt, akkor újraküldi a szelvényt. A TCP kapcsolat kétirányú adatáramlást biztosít, pont-pont összeköttetésben, így nem támogatja a többesadást és a szórtadást [2].

Bizonyos esetekben a sebesség és a hatékonyság fontosabb, mint a megbízhatóság. Ilyenkor a küldő fél nem foglalkozik az adatok nyugtázásával, csupán elküldi a csomagokat. Ez az alap filozófiája az UDP protokollnak. A folyamatos nyugtázás, az elveszett szegmensek újraküldése, a torlódásvédelem lassítja az adatátvitelt. Ott ahol minél gyorsabb átvitelre van szükségünk (pl. médiafájlok), vagy ahol a felhasználói adatmennyiség csekély (pl. PLC vezérlési adatok) célszerűbb az egyszerűbb felépítésű UDP protokollt alkalmazni. Ha az üzenetek rövidek, nincs szükség az adatok sorba rendezésére sem, mert azok egy szegmensen belül elférnek. Az UDP leírását az RFC 768-as szabvány tartalmazza, amelyet 1980-ban bocsátottak ki [6].

3. Hálózati réteg (Network layer): Feladata az előző réteg által előkészített csomagok eljuttatása a forrástól a célig, mialatt a csomagok sok esetben több közbeeső csomóponton (router) is áthaladnak. Ha a forrás és a cél eltérő típusú hálózatokban van, ez a réteg gondoskodik a hálózatok közti különbségből fakadó feladatok elvégzésére is. A leggyakrabban alkalmazott hálózati protokollokat az 1. táblázat foglalja össze [2].

1. táblázat. Fontosabb hálózati protokollok

Hálózati protokoll	Gyártó/szabvány
DDP (**D**atagram **D**elivery **P**rotocol)	Apple Computer
IDP (**I**nternet **D**atagram, **P**rotocol)	Xerox Network System
IPv4 (**I**nternet **p**rotokoll 4-es változat)	IETF szabvány
IPv6 (**I**nternet **p**rotokoll 6-es változat)	IETF szabvány
IPX (**I**nternetwork **P**acket **E**xchange)	Novell
NetBIOS (**N**etwork **B**asic **I**nput/**O**utput **S**ystem)	Microsoft
SNA (**S**ystems **N**etwork **A**rchitecture)	IBM

A hálózati réteg legismertebb és talán a legelterjedtebben használt protokollja az IPv4. Ebben a protokollban minden egyes csomag külön-külön címzést kap, amely egyértelműen azonosítja azt a két csomópontot, amely között felépül az éppen aktuális kapcsolat. Tehát nem feltétlenül azonos a forrás és a cél címével. A címzéshez 4 bájtot (32 bit) használ, ezeket rendszerint decimális formában, a bájtok között ponttal elválasztva, IP címként ismerjük [2]. Az IPv4 elterjedtségét mi sem bizonyítja jobban, mint az, hogy már az IPv6, 16 bájtos változatot is használják, mert a 4 bájtos címzés adta lehetőségek nemrég kimerültek.

2. Adatkapcsolati réteg (Data Link layer): Az adatkapcsolati réteg elsődleges feladata, hogy a hálózat csomópontjai között hibamentes átvitelt biztosítson, illetve az adatkeretek kezelése. Az adó kapcsolati rétege ezeket a kereteket bitenként továbbítja a fizikai réteg felé. Közben megfelelő bitmintákkal ellátja a kerethatárokat, azért, hogy a vételnél, a fizikai réteg felől kapott biteket, a vevő adatkapcsolati rétege vissza tudja alakítani keretekké. Továbbá a küldő gép adatkapcsolati rétege a keret tartalmát felhasználva elvégez egy számítást is az adatkereten, melynek eredményét hozzácsatolja a kerethez. A fogadó gép adatkapcsolati rétege megismétli ezt a műveletet és a kapott eredményt összehasonlítja a kerethez csatolt értékkel, ezzel biztosítva a keretek sérülésmentességének ellenőrzését. Ha a keret sérülésmentes, akkor a fogadó gép adatkapcsolati rétege küld egy nyugtajelet (acknowledgement frame) a küldő gép adatkapcsolati rétegének.

Mivel egy hálózatban az adott fizikai csatornát többen is akarják használni, a hozzáférést az adatkapcsolati réteg egyik különleges alrétege, a *Közegelérés-vezérlési alréteg (Medium Access Control, MAC)* kezeli. Akárcsak a hálózati réteg, a MAC szintű kapcsolat is használ címzést az eszközök azonosítására. Itt az eszköz valóságos fizikai címe kerül felhasználásra. Az IEEE szabványban határozta meg, hogy mindegyik Ethernet hálózati eszköznek rendelkeznie kell egy jól meghatározott egyedi, 48 bites fizikai címmel. Ezt nevezik MAC címnek. Az első 24 bit a gyártót azonosítja, a fennmaradó 24 bit pedig a hálózati eszköz sorozatszámát tartalmazza. Az adatkapcsolati rétegen működő ARP (Address Resolution Protocol) az adónál az előző rétegtől kapott IP címek alapján határozza meg a MAC címeket. A vevőnél ennek a fordítottja történik, a RARP (Revers Address Resolution Protocol) visszaállítja az IP címeket a MAC címek alapján [9].

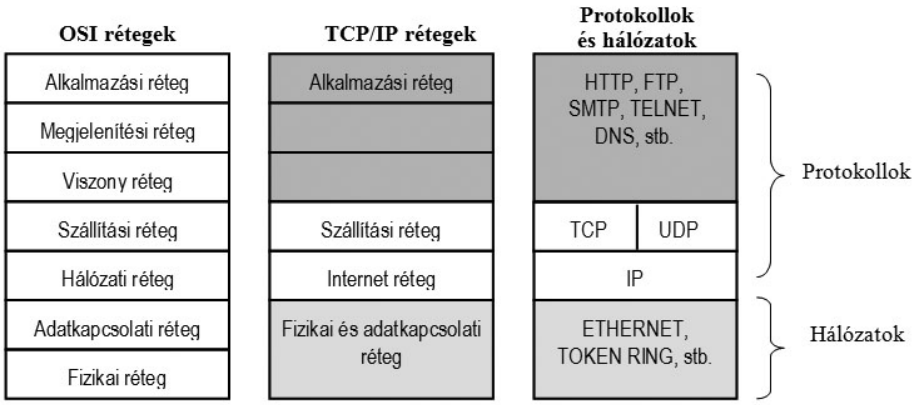
1. Fizikai réteg (Physical layer): Ezen a rétegen zajlik a tényleges, jelszintű adattovábbítás. A rétegnek biztosítani kell azt, hogy az adótól elküldött 1-es bit, a másik oldalon, a vevőnél ugyancsak 1-esként jelenjen meg. Ennek érdekében, az adónál a fizikai réteg bitértékekhez feszültségszinteket rendel, vagyis kódolja az információt, a vevőnél pedig a feszültségszintekből visszaállítja a bináris információt. Az alkalmazott kódolási technikát szabvány írja elő. Pl. a 10 Mb/s-es Ethernet a Manchester-kódolást, a 100Base-TX pedig a 4B/5B kódolást használja [3].

2.2. A TCP/IP hivatkozási modell

Megjelenése után az OSI modellt igencsak sok kritika érte. Egyrészt azért, mert túl bonyolultnak tartották, másrészt azért, mert protokolljai nem teljesen tökéletesek. A viszony réteget alig használja néhány alkalmazás, a megjelenítési réteg pedig szinte teljesen üres. Igaz, hogy eredetileg csak öt réteg volt, de mivel a nagy befolyású IBM-nek már volt egy 7 rétegű modellje ezért 7 rétegűre módosították [10].

Az időzítés sem volt túl szerencsés. Egy szabvány megjelentetésének időpontja rendkívül erősen befolyásolhatja annak sikerét. A szabványosításnak a kutatások befejezése után és a beruházások megkezdése előtt kell megtörténnie. Ez azért fontos, hogy a kellő tapasztalat birtokában egységes szabványt hozhassunk létre. Mire az OSI protokollok megjelentek, addigra a Berkeley-féle UNIX TCP/IP protokolljai már széles körben elterjedtek a kutatóegyetemeken [4].

A TCP/IP modell lényegében az OSI modellnek egy egyszerűsített változata, amely csak négy réteget tartalmaz, de megvalósítja mindazokat a funkciókat, amelyek szükségesek a biztonságos kommunikációhoz. A két modell közötti különbséget és az alkalmazott protokollokat a 3. ábra mutatja.

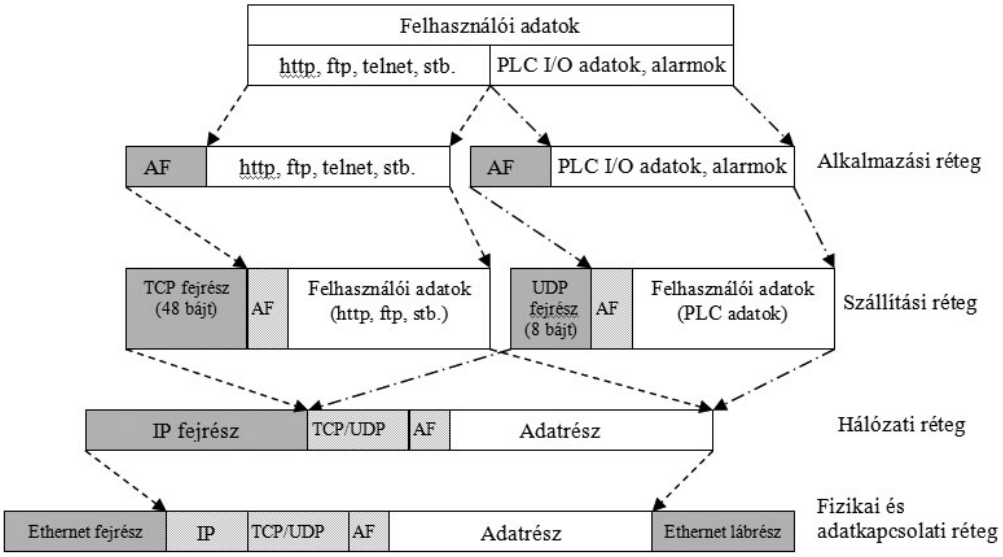


3. ábra. Az OSI és a TCP/IP hivatkozási modell és protokolljai

2.3. Adattovábbítási stratégia a TCP/IP rétegeken keresztül

Az előző fejezetben bemutatott TCP/IP rétegmodell egyaránt alkalmas általános felhasználói és az ipari Ethernetet alkalmazó rendszerek tárgyalására is. Ezért a továbbiakban ennek működési stratégiáját szeretném bemutatni, kiemelve azokat a sajátosságokat, amelyek leginkább az ipari Ethernetre jellemzőek.

Tételezzük fel, hogy a felhasználói adatok, legyenek azok általános internet információk, vagy speciális ipari alkalmazások adatai, a forrástól a célig jól meghatározott sorrendben végighaladnak a különböző rétegeken és közben járulékos információk hozzáadásával, egymásba ágyazódva (4. ábra), a fizikai közegen keresztül érik el a célállomás rétegeit. Ezeken áthaladva rendre lebomlanak a járulékos információk, így a célhoz csak az eredeti üzenet érkezik.



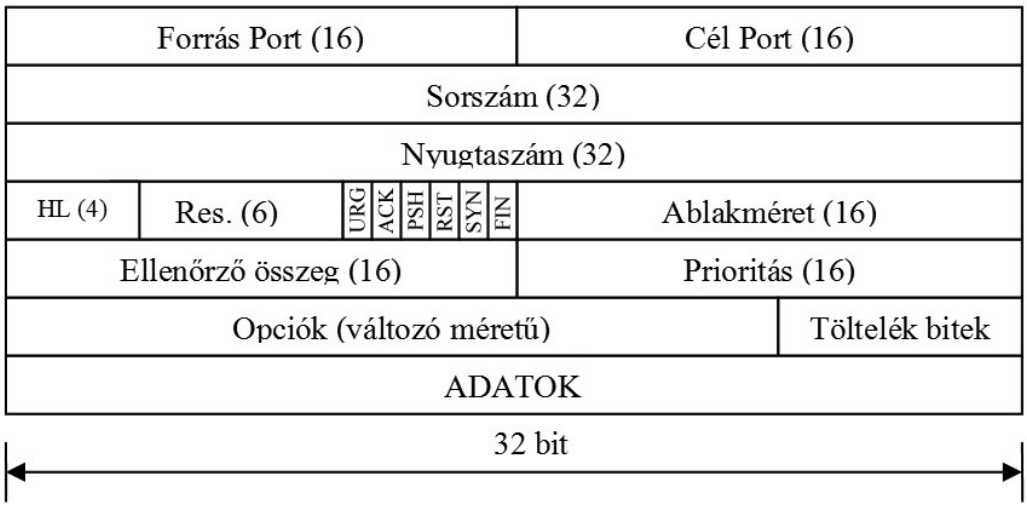
4. ábra. Adatáramlás a rétegeken keresztül

A szállítási réteg az alkalmazástól kapott tetszőleges hosszúságú üzenetekből álló adatfolyamból adatszegmenseket készít. A szegmens méretét az MTU (Maximum Transfer Unit) határozza meg, amely minden hálózat meghatározó jellemzője. A gyakorlatban ez néhány ezer bájtot jelent (a legtöbb esetben 1500 bájt), de maximum 64 kilobájt lehet, azért, hogy a fejléccel együtt elférjen a hálózati réteg IP adatmezőjében. Ezeket a szegmenseket a hálózati réteg IP csomagonként továbbítja. Amikor a szelvények elérik a célállomást, az ottani szállítási réteg azokat ismét adatfolyammá rakja össze és továbbítja a felhasználói folyamatnak.

2.4. A TCP protokoll működése

Az Ethernet hálózatokon a legelterjedtebben használt szállítási protokoll a TCP (Transmission Control Protocol). Egyes mérések szerint a teljes Internetes forgalom mintegy 95%-a a TCP protokollt használja. A TCP összeköttetés elsődleges célja a biztonságos szállítás. Feladata, hogy a kapott szelvényeket megfelelő sorrendben rakja össze az eredeti üzenetnek megfelelően. Küldéskor, az adatátvitel hibátlanságának biztosítása érdekében kézfogásos (handshaking) módszert, un. pozitív nyugtát használ. Az adó elküldi az adatot, közben elindít egy időzítőt is és várakozik a nyugtázásra. Ha nem kap választ az időzítés lejárta előtt, akkor újraküldi a szelvényt. A TCP kapcsolat kétirányú adatáramlást biztosít, pont-pont összeköttetésben, így nem támogatja a többesadást és a szórtadást [2].

Minden TCP szegmens egy fix hosszúságú, 20 bájtos fejléccel kezdődik, de a HL fejrész hossztól függően akár 24 bájt is lehet. A TCP fejrész, a forrás és a cél IP címét nem tartalmazza, csak a portok számát, amelyeken keresztül kapcsolatot tart az alkalmazási réteggel. Különböző alkalmazások más és más portokat használnak az internetes kommunikáció során. Célszerű a fontosabb alkalmazások számára lefoglalni un, „jól ismert” (well-know) portokat. (Pl. az FTP a 21-es portot, a Telnet a 23-ast, a HTTP a 80-ast használhatja.) A küldő és a célállomásra vonatkozó információk az előző réteg fejrésze (AF) tartalmazza, amelyet majd az IP réteg hozzárendeli a port számához. Így jön létre a 48 bites egyedi szállítási cím. A következő ábra a TCP szelvényformátumát mutatja.



5. ábra. A TCP szelvényformátuma (a zárójelben a bitek száma látható)

A fejrészben szereplő elemek közül leginkább a szabályozási jelzőbiteket emelném ki, amelyek főleg a biztonságos adatátvitelért felelősek:

- URG (Urgent): - ha 1-esre van állítva sürgősségi mutatóként használják.
- ACK (Acknowledgement): - a SYN jelzőbittel együtt használatos. 1-es értéknél az összeköttetés során a válasz nyugtázását jelenti. Ilyenkor a nyugtaszám mező a vevő által a soron következő adatbájt sorszámát tartalmazza.
- PSH (Push): - ha 1-es, a TCP protokoll érkezéskor azonnal továbbítja az adatokat az alkalmazásnak és nem tárolja az ütközési tárban [2].
- RST (Reset): - törli a kapcsolatot, ha zavarossá vált, és az adatátvitel már nem biztonságos.
- SYN (Synchronize): - az összeköttetés létesítésére használják. SYN =1 és ACK = 0, a nyugtaszám mező nincs használva.
- FIN (Finished sending data): - az adás végét jelzi, vagyis az összeköttetés bontására használják.

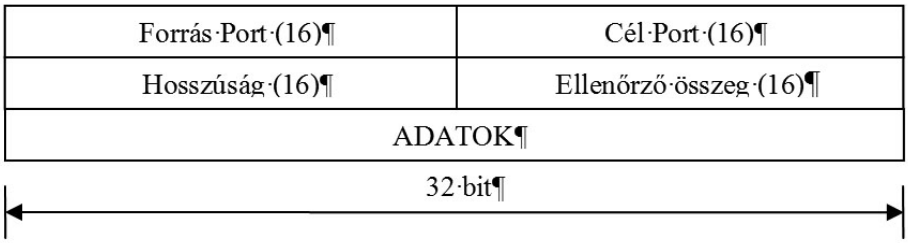
A sorszám (sequence number) és a nyugtaszám (acknowledgement number) mezők a bájtok sorszámát mutatják, azzal a különbséggel, hogy az utóbbi a várt bájt sorszámát tartalmazza, nem az utolsó rendben beérkezett bájtét, mert a küldési sorrend nem feltétlenül egyezik meg a bájtok sorrendjével. A HL fejrészhossz (header lenght) mondja meg, hogy hány 32 bites szóból áll a TCP fejrész, azaz használják-e az opciók mezőt.

Érdekes lehet még az ablakméret jelentése. A nyugtakeretben mindig fel van tüntetve az utolsó fogadott bájt sorszáma, és az, hogy ettől kezdve hány bájtot képes fogadni a vevő. Ez az érték akár nulla is lehet, ami azt jelenti, hogy a vevő puffere megtelt, nem tud több adatot fogadni, mert az alkalmazás felé lelassult az adatok átadása. (Gyors adó, lassú vevő esete.)

2.5. Az UDP protokoll

A TCP szállítási protokollt azok az alkalmazások használják, amelyek igénylik az adatcsomagok sorrendhelyes és megbízható adatátvitelt. Ennek azonban ára van. A folyamatos nyugtázás, az elveszett szegmensek újraküldése, a torlódásvédelem lassítja az adatátvitelt. Ott ahol minél gyorsabb átvitelre van szükségünk (pl. médiafájlok), vagy ahol a felhasználói adatmennyiség csekély (pl. PLC vezérlési adatok) célszerűbb az egyszerűbb felépítésű UDP protokollt alkalmazni. Ha az üzenetek rövidek, nincs szükség az adatok sorba rendezésére sem, mert azok egy szegmensen belül elférnek. Az UDP leírását az RFC 768-as szabvány tartalmazza, amelyet 1980-ban bocsátottak ki [6].

Az UDP fejrész mindösszesen 8 bájt hosszú. Csak a forrás- és a célállomás portjainak számát tartalmazza, valamint a csomag hosszára vonatkozó információt és egy ellenőrző összeget. (6. ábra)



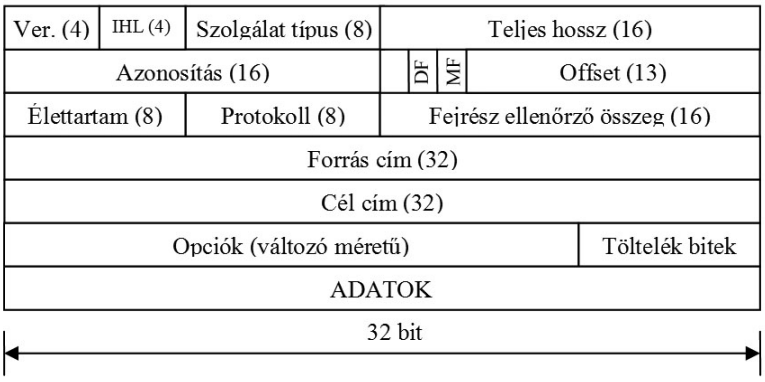
6. ábra. Az UDP datagram felépítése

Az ellenőrző összeg az egyedüli biztonsági információ az adat tartalmára vonatkozóan. Ha ez hibára utal, a vevő egyszerűen eldobja a csomagot és figyelmen kívül hagyja. Ebből következik, hogy adatvesztés vagy adatduplikáció bizony előfordulhat az UDP-re épülő kapcsolatban, ezért alkalmazás szinten kell megoldani ezeket a kérdéseket. Nagy előnye viszont a gyorsasága. Összehasonlítva a TCP protokollal, mintegy háromszor gyorsabb adatátviteli sebességet biztosít. Gyorsaságának köszönhetően számos valós idejű ipari Ethernet rendszer használja.

2.6. Az IP protokoll

A hálózati réteg legfontosabb protokollja. Az a feladata, hogy optimális útvonalat biztosítson a szállítási réteg által előkészített datagramok számára a forrásállomástól a célig, függetlenül attól, hogy ezek azonos vagy különböző hálózatban vannak. (Ez azt jelenti, hogy az IP protokoll segítségével tudunk kapcsolatot teremteni a helyi hálózatok között is.) A szállítási rétegtől kapott datagramokat adatként kezeli és hozzáfűzi saját fejlécét, amely tartalmazza az összes olyan információt, amely szükséges a pontos célba éréshez. Az így összeállított csomagot továbbítja az adatkapcsolati rétegnek. A 7. ábrán az IPv4 csomag formátumát láthatjuk.

A fejrésznek van egy 20 bájtos rögzített, és egy változó hosszúságú opcionális része. A továbbítás a legtöbb esetben a *Verzió mező* legmagasabb helyértékű bitjével kezdődik. Egyébként ez a mező tartja nyilván, hogy a protokoll melyik verziójához tartozik a datagram. Így lehetséges az, hogy a verziók közötti átmeneti időszakban is működhet a kommunikáció.



7. ábra. Az IP keretformátuma

Mivel a fejrész hossza nem feltétlenül állandó, az *IHL* 4 bites mező adja meg, hogy hány darab 32 bites részből áll a fejrész. A legkisebb érték 5, azaz 20 bájt, amikor az *Opciók*

rész hiányzik. A maximális értéknél (15), még 40 bájtnyi bejegyzés fűzhető a datagramhoz, hogy ne kelljen olyan információk számára is fejlécbiteket lefoglalni, amelyekre csak ritkán van szükség. Összesen 5 opciót lehet definiálni. Mindegyik egy egybájtos kóddal kezdődik, amelyet egy szintén egybájtos hosszmező követ, majd egy vagy több adatbájt, amelyet szükség esetén töltelék bitekkel (Pad) egészítenek ki 4 bájtra vagy ennek többszörösére. Ilyen opciók például a *Biztonság*, amely azt mutatja, hogy mennyire titkos az információ, hogy ezáltal kikerülhetőek legyenek olyan routerek, amelyek nem biztonságosak. Ugyancsak használatos még az *Útvonal feljegyzés* opció, amikor mindegyik router, amelyen áthalad a csomag hozzáfűzi az IP címét, azért, hogy a csomag útja utólag ellenőrizhető legyen, vagy az *Időbélyeg opció*, amely az IP cím mellé, még egy 32 bites időbélyeget is felír.

A fejléc mezői közül a legfontosabb és nélkülözhetetlen információt a *Forrás cím* és a *Cél cím* mezők tartalmazzák, amelyek a küldő és a fogadó állomások IP címét mutatják.

A további mezők jelentése:

- *Szolgálat típus (ToS)*: - Prioritás megadása, 0-tól 7-ig,
  - D (Delay) késleltetés,
  - T (Throughput) átbocsátás,
  - R (Reliability) megbízhatóság.
- *Teljes hossz*: - a csomag hossza a fejléccel együtt.
- *Azonosítás*: - a több darabból álló datagramokat azonosítja. Adott datagramhoz tartozó darabok azonosítója ugyanaz.
- *DF (Don't Fragment)*: - jelzi a routernek, hogy a datagram nem darabolható, mert a vevő nem képes visszaállítani.
- *MF (More Fragment)*: - az utolsó kivételével, mindegyik darabnál be kell állítani, hogy tudjuk, hogy minden darab megérkezett-e.
- *Offset*: - megmutatja, hogy a darab hová tartozik a datagramon belül. A 13 bit maximum 8192 darabot tesz lehetővé datagramonként.
- *Élettartam (Time to live, TTL)*: - egy számláló, amelynek csökken a tartalma, amikor a csomag áthalad a routeren. Ha eléri a nullát, vagy a routerekben beállított minimális értéket, a csomagot eldobják, ezáltal megakadályozható, hogy rossz irányba továbbított csomagok „céltalanul bolyongjanak" a hálózatban.
- *Protokoll*: - megmutatja, hogy milyen szállítási protokolltól származik a datagram. (TCP, UDP, stb.)
- *Fejrész ellenőrző összeg*: - egy számlálási algoritmus szerint, ha az értéke nulla, a fejrész sértetlen. Minden egyes ugrásnál újra kell számolni a TTL változása miatt.

2.7. IP címek rendszere.

Az Ethernet hálózatban jól meghatározott egyedi címmel kell rendelkezzenek mindazok a hosztok, amelyek adatokat szeretnének fogadni vagy továbbítani a helyi vagy távoli állomásoknak. Ennek megfelelően az ipari Ethernet hálózat szereplői is (pl. a PLC-k és IO eszközök) a legtöbb esetben rendelkeznek IP címmel. Ennek köszönhetően ezek az eszközök, megfelelő jogosultságok birtokában, akár nagy távolságról is elérhetők, lekérdezhetők, konfigurálhatók esetleg újraprogramozhatók.

Mindegyik IPv4-es cím 32 bites és két részből áll: az első rész a hálózatot, a második pedig a hosztot azonosítja. Összességében a teljes kioszthető címtartomány öt osztályra tagolódik, attól függően, hogy nagy, közepes vagy kis hálózatokhoz tartoznak, illetve többesadás, valamint fenntartott címek (2. táblázat). Ezeket rendszerint 0-255-ig terjedő, egymástól ponttal elválasztott számcsoportokkal jelöljük.

2. táblázat. IPv4 címosztályok

Név	Perfix	1. bájt	2. bájt	3. bájt	4. bájt	Címtartomány	Alhálózati maszk
A osztály	0	Hálózat 1 - 126	Host 0.0.0 – 255.255.255			1.0.0.0 – 126.255.255.255	255.0.0.0
B osztály	10	Hálózat 128.0 – 191.255		Hoszt 0.0 – 255.255		128.0.0.0 – 191.255.255.255	255.255.0.0
C osztály	110	Hálózat 192.0.0 – 223.255.255			Hoszt 0 - 255	192.0.0.0 – 223.255.255.255	255.255.255 .0
D osztály	1110	Többesadás				224.0.0.0 – 247.255.255.255	
E osztály	11110	Fenntartott címek				240.0.0.0 – 255.255.255.255	

A legtöbb hálózat az A, B és C címosztály valamelyikét használja. A táblázatból láthatjuk, hogy a C címosztályban lehet a legtöbb (2 097 152), egy mástól független helyi hálózatot létrehozni. Mindegyik hálózat legfeljebb 254 állomást tartalmazhat. (Az x.y.z.0 cím nem osztható ki, mert az mindig az alhálózatot azonosítja, az x.y.z.255 pedig az üzenetszórás címe.) Az ipari Ethernetes hálózatok eszközei számára is a legtöbb esetben valamilyen C osztályú címtartományon belüli címeket szokás kiosztani (akár DHCP-vel). Nagyon gyakran, fölég, ha az Internetről történő elérés szükségtelen, a saját ipari Ethernetes hálózat számára szokás az un. *privát címtartomány*beli címeket használni (3. táblázat). Ezeket a címeket Internetes kapcsolatokra nem használják, a routerek pedig nem továbbítják az ilyen címekkel ellátott csomagokat [8].

3. táblázat. Privát címtartományok

Címosztály	Címtartomány	Lehetséges hostok száma
A	10.0.0.0 – 10.255.255.255	16 646 144
B	172.16.0.0 – 172.31.255.255	1 040 384
C	192.168.0.0 – 192.168.255.255	65 024

Érdemes még megemlíteni a 127.x.y.z *visszahurkolási címek*et, amelyeket úgyszintén nem lehet használni hálózati címként, mert ezek a címek az eszköz saját címét jelentik. Az erre küldött csomagok nem hagyják el az adott eszközt, hanem úgy szerepelnek, mint beérkező csomagok. Főleg hálózati eszközök tesztelése és hibafeltárása alkalmával használhatók eredményesen.

2.8. A fizikai és az adatkapcsolati réteg faladatai és működése.

Az ipari kommunikációs rendszerekben egyrészt a gép-gép közötti kommunikáció, másrészt az egyszerű protokollok alkalmazása, valamint a valós idejű működés miatt számos esetben, főleg a szigorúan valós idejű (Hard Real-Time) kommunikáció során, csak az 1. 2. és 7. OSI rétegeket használjuk. Ezek közül is kitűntetett szerepet játszik az adatkapcsolati réteg. Ezért ennek a rétegnek a feladataival, illetve működésével kicsit bővebben szeretnék foglalkozni. Ennek a rétegnek a feladata az előző réteg által előkészített csomagoknak az átvétele, elkülönítése egymástól, a forrás- és a célállomás fizikai címének leképzése az IP címből (ha van ilyen), a hibafelismerés biztosítása, valamint az így létrehozott keret továbbítása a fizikai rétegnek. Vételi oldalon pedig a kapott bitekből összerakni a kereteket, ellenőrizni az esetleges hibákat és a hibátlan kereteket a felette lévő réteg számára megfelelő

formátumúvá tenni. Akárcsak a többi rétegek esetében itt is egy látszólagos kapcsolat épül fel az adó és vevő között.

Az IEEE802-es szabvány az adatkapcsolati rétegen belül két, jól meghatározott funkciókat ellátó alréteget definiál: - közegelérés-vezérlési alréteg (MAC),

… ….- logikai kapcsolatvezérlési alréteg (LLC).

2.9. A közegelérés-vezérlési (MAC) alréteg feladatai

Az Ethernet hálózat eszközei a legtöbb esetben egy közös fizikai csatornát használnak a kommunikáció megvalósításához. Az volna jó, ha egyszerre, egy időben csak két állomás használná a vonalat. Egyik legfontosabb feladata a MAC alrétegnek, hogy olyan hozzáférést biztosítson az eszközök számára, amely az adatátvitel szempontjából a legmegfelelőbb és a legbiztonságosabb. E nélkül az adatátvitel során az információk egymásra íródnának vagy megsemmisítenék egymást a sínkonfliktus következtében. A közegelérés-vezérlési eljárások azokat a szabályokat tartalmazzák, amelyek megadják, mikor és mennyi ideig használhatja egy állomás az adatcsatornát. A továbbiakban ezek közül az eljárások közül fogok néhányat bemutatni, amelyeket főleg az ipari Ethernetes kommunikációban használnak.

3. A CSMA/CD protokoll működése

Az IEEE 802.3 szabvány szerinti Ethernet hálózatba kapcsolt állomások mindegyike egyenlő jogokkal rendelkezik a közeg használatát illetően. A hozzáférés ellenőrzése és vezérlése a CSMA/CD (Carrier Sense Multiple Access with Collision Detection) közegelérési módszer szerint történik, amelynek röviden a lényege a következő.

Mindegyik állomás állandóan figyeli a hálózati forgalmat. Ha talál olyan adatcsomagot, amelyet neki címeztek leveszi, a többit figyelmen kívül hagyja. Az adatküldésre készülő állomás is hallja a közegen zajló forgalmat, és ha azt érzékeli, hogy valaki már adásban van, azaz vivőt érzékel (Carrier Sense), akkor várakozik az adás megszűnéséig. Amikor elcsendesül a közeg, megkezdi az adást, amelyet egyedül csak a megcímzett állomás fog átvenni. Minél több állomás van az adott hálózati szegmensre kapcsolva, annál nagyobb az esélye annak, hogy egy időben egyszerre két vagy több állomás is küldeni akar. Ilyenkor a közegen a villamos jelek összeadódhatnak, az adatok sérülnek, az információ használhatatlanná válik. Ezt az állapotot érzékelni kell (Collision Detection), sőt figyelmeztetni kell rá a többi állomást is, hogy ütközés történt. Az ütközést az adó érzékeli, amikor visszaolvasáskor nem ugyanazt az adatot kapja. Azért, hogy minden állomás értesüljön az ütközésről, az ütközést észlelő egy rövid ideig egy 32 bit hosszúságú un. JAM jelsorozatot helyez a hálózatra, amelynek hatására mindegyik állomás azonnal beszűnteti az adást, majd véletlenszerű késleltetési idő után újból próbálkozik.

A véletlen időzítésnek köszönhetően valamelyik adó elsőként lefoglalja és megkezdheti az adását, ezt a többiek érzékelni fogják, mielőtt megkezdenék adásukat. A véletlen várakozási idő N ütközés után maximum a következő lehet:

$$t_w = t_s \cdot (2^N - 1) \text{; ahol: } 1 \leq N \leq 16 \qquad (1)$$

A fenti relációban szereplő t_s résidő (slot time), vagy körbejárási késleltetés, az az idő mialatt a keret a szegmens két legtávolabbi pontja között körbejár ($t_s = 5{,}12\ \mu s$ Fast Ethernet hálózatban). Ennyi idő szükséges ugyanis ahhoz, hogy mindegyik állomás biztonsággal érzékelje az ütközést.

A várakozási idő egy algoritmus szerint (Binary Exponential Backoff) kerül meghatározásra. Az első ütközés után 0 vagy 1 a szorzó, a második után 0, 1, 2, 3, a harmadik után 0, 1, ..., 7 és így tovább. A 10-edik ütközés után (1023) a szorzó nem nő tovább, a 16.-odik ütközés után az eszköz tovább már nem próbálkozik és hibaüzenettel jelzi az átvitel sikertelenségét.

Az (1) összefüggésben szereplő résidő függ a csatorna terjedési sebességétől (v) és a leghosszabb hálózati szegmens hosszától (L). Ennek megfelelően a résidőhöz tartozó minimális keretméret (d_F), ismerve a bitsebességet (b) a következő összefüggéssel határozható meg:

$$d_F = \frac{2 \cdot L}{v} \cdot b \tag{2}$$

Az IEEE 802.3 szabvány annak idején a 10 Mb/s-os Ethernet hálózatban, amelynek a hossza négy ismétlővel maximálisan 2500 m lehetett, ezt a keretméretet felkerekítve, 512 bitben határozta meg, ami azt jelenti, hogy a minimális keretméret 64 bájt nagyságú kell legyen. Vagyis ahhoz, hogy az ütközést elkerüljük a jelenleg használatos nagyobb sebességű hálózatokban vagy csökkentenünk kell a szegmensek hosszát, vagy növelnünk kell a minimális keretméretet [11].

4. Az IEEE 802.3 Ethernet keret felépítése

A MAC alréteg egy másik fontos funkciója a keretek létrehozása és ezeknek jól meghatározott elkülönítése egymástól, valamint a hibaellenőrzés. Az IEEE 802.3 keretstruktúráját a 8. ábrán láthatjuk.

Előtag	Keret-határoló	Célcím	Forráscím	Adat-hossz	**Adatok**	Ellenőrző összeg
7 bájt	1 bájt	6 bájt	6 bájt	2 bájt	**46-1500 bájt**	4 bájt

8. ábra. Az IEEE 802.3 keretformátuma

Az előtag és a kerethatároló igazából nem része a keretnek. A MAC alréteg azért fűzi a keret elejére, hogy biztosítsa a vevő hardverjének ráhangolódását az adó órájára. Tulajdonképpen egy 10101010 mintájú jel 7 bájton keresztül, amelynek a Manchester kódja egy 10 Mhz-es, 5,6 μs időtartamú négyszögjelnek felel meg 10 Mb/s bitsebességnél. Ezt követi a kerethatároló bájt, amely hasonló, kivéve az utolsó bitet, amely ugyancsak 1-es. Ez jelzi, hogy itt kezdődik a keret. A célcím és forráscím az adó illetve a vevő MAC címét tartalmazza, amelyet az adó alrétegének ARP protokollja képez le az IP címből. A vevőnél pedig a RARP éppen az ellenkezőjét teszi, a MAC címből visszaállítja az IP címet a hálózati réteg számára.

A kétbájtos adathossz mező, az adatmezőben található bájtok számát adja meg, amely 0 és 1500 közötti érték lehet. Ha 46 bájtnál kisebb ez a szám, a keret kiegészül töltelékbitekkel, azért, hogy a 64 bájtos minimális keretméret meglegyen. Ismervén az adatmező hosszát, a vevő a vételnél leválasztja ezeket.

Az ellenőrző összeg vagy FCS (Frame Check Sequence) tulajdonképpen egy CRC (Cyclic Redundancy Check) algoritmus szerint kapott osztási művelet 32 bites maradéka, amelyet a keret végéhez fűz hozzá. A vételnél a teljes keretet a CRC-vel együtt, ha elosztják ugyanazzal az értékkel (a generátor polinommal) akkor, ha hibátlan a keret maradék nélküli

értéket kapunk. Akár már egyetlen egy bit megváltozását is érzékeli, de 99,9%-ban alkalmas csoportos bithibák felderítésére is.

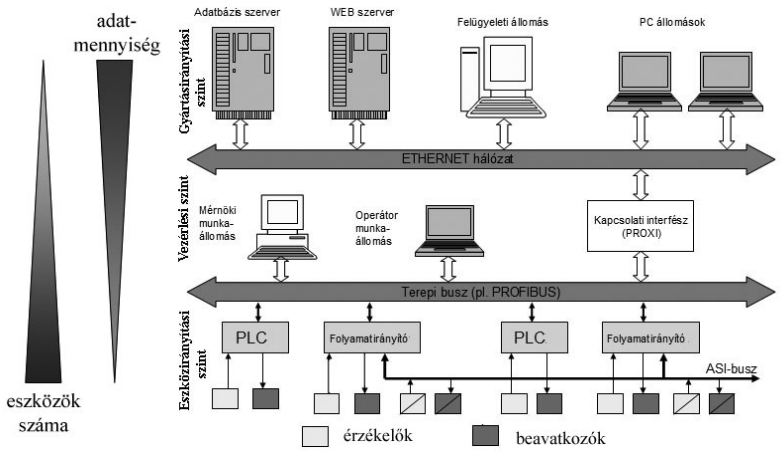
A keret tartalmaz még az adatmezőbe beágyazva három bájtnyi LLC (Logical Link Control) alréteg információt. Ebből kettő, a DSAP (Destination Service Access Point) és az SSAP (Source Service Access Point) a címzett illetve a forrás protokoll azonosítója, a CONTROL blokk pedig az LLC szolgáltatás módját mutatja [10].

Ezek után pedig újra azzal szembesülünk, hogy van egy szabvány szerinti keretünk, amit gyakorlatilag senki sem használ. Még a Windows alapú operációs rendszerek is az eredeti Ethernet II elnevezésű, természetesen egyszerűbb keretformátumot használják alapértelmezésben. Ebből következik, hogy az ipari Ethernet rendszerek is az utóbbit használják. Az Ethernet II keret formátuma annyiban tér el a 802.3 formátumtól, hogy nem az LLC mezők azonosítják a protokollt, hanem az adathossz mező helyett, az ugyancsak 2 bájtos Ethertype mezőt használják.

Az előzőekben bemutatott, igencsak nagyfokú bizonytalanságot tükröző közeg-hozzáférési algoritmus ismeretében jogosan feltehetnénk magunknak a kérdést, egyáltalán hol használnak ilyen kiszámíthatatlannak tűnő kommunikációs kapcsolatot? A válasz nagyon egyszerű: mindenütt, kivéve ahol nincsenek időhöz kötött, determinisztikus folyamatok, ott ahol egy esetleges ütközés után kialakult véletlenszerű várakozási idő a valós idejű kapcsolatokat kedvezőtlenül befolyásolná. Az Ethernet hálózat alkalmazása viszont számos előnnyel rendelkezik. Világméretű kiterjedése, az Internet közvetlen elérhetősége, a hardvereszközök és szoftverek viszonylag alacsony ára, a nagy átviteli sebesség (100 Mb/s, 1 Gb/s, stb.) mind olyan tényezők, amelyek az ipari automatizálás fejlesztői számára célul tűzték ki azt a feladatot, hogy hogyan lehetne az Ethernet kommunikációt determinisztikussá tenni?

5. Az ipari Ethernet kialakulásának körülményei

Az ipari irányítóberendezések közötti kommunikációs kapcsolat alapfeltétele egy korszerű, biztonságos decentralizált folyamatirányító rendszer kialakításának. A technológia közvetlen közelébe telepített ipari vezérlők egy-egy részfolyamat közvetlen irányítását látják el. Az innen érkező adatokat továbbítani kell a rendszer adatbázisába, hogy feldolgozható, kiértékelhető esetleg megjeleníthető vagy archiválható legyen. A 70-es évektől kezdődően a vezérlési feladatokat egyre inkább a programozható logikai vezérlők (PLC) látják el. A mikroprocesszorok szinte minden területet átfogó elterjedése, az analóg jeltovábbítást felváltó, egyre több információt hordozó digitális jelközlés lehetővé tette az új irányítási megoldások használatát. Az intelligens távadók és beavatkozók megjelenésével az adatfeldolgozás osztott jellege kiszélesedik. Ezek a készülékek csak villamos segédenergiával működnek, digitális jelekkel dolgoznak és alkalmasak a hozzájuk tartozó irányítórendszerrel akár kétirányú kommunikációra. Az erőforrások megosztása miatt a kommunikációt biztosító buszrendszerek szerepe egyre jobban felértékelődik. Az egykor háromszintű folyamatirányítási hierarchia (9. ábra) a fizikai hálózat szempontjából egyre inkább egy, legfeljebb két szintre korlátozódik. A közöttük lévő különféle kommunikációs kapcsolatok pedig az ipari Ethernet irányába fejlődnek.



9. ábra. Klasszikus háromszintű folyamatirányítás

Az egyre intelligensebb eszközvezérlők (iDevice) kezdenek egyenrangúakká válni az őket irányító PLC-kkel. Sok esetben képesek önálló kommunikációs kapcsolatot is teremteni egymással. Az egyre fejlettebb rendszerek, egyre nagyobb adatmennyiséget kell továbbítsanak. A klasszikus terepi buszok az alacsony adatátviteli sebességük mellett már nem képesek kiszolgálni ezeket a fejlett rendszereket.

Akárcsak a PLC-k, a hozzájuk tartozó terepi buszrendszerek is gyártó-specifikusan alakultak ki. Ez megnehezítette bizonyos terepi eszközök cseresabatosságát. A rendszerek bővítése vagy esetleges módosítása csak ugyanattól a gyártótól beszerezhető eszközökkel volt lehetséges. Mindezek nagyban hozzájárultak ahhoz, hogy egy egységes kommunikációs rendszert használjanak az ipari automatizálás területén is. Ez pedig nem más, mint az ipari Ethernet, amelynek alkalmazása számos előnyt jelent a hagyományos terepi buszokkal szemben:

- a tömeggyártás miatt olcsó eszközök használata,
- a különböző gyártmányú eszközök könnyen integrálhatók a rendszerbe,
- rendszerint többféle hálózati topológiát támogatnak (van, amelyik mindegyiket),
- azonos vagy kompatibilis protokollok használhatók az adatfeldolgozási és a technológiai szinten egyaránt, ezáltal a termelési adatok irodai szinten közvetlenül elérhetők,
- nagy átviteli sebesség és nagyobb sávszélesség érhető el, amely lehetővé tesz olyan technikák alkalmazását, amelyek a képinformáció továbbításán alapulnak, mint például a gépi látás,
- könnyű integrálni az Internettel, és akár minden szinten kihasználhatók a web alapú programozás előnyei,
- egységes és ezáltal olcsóbb üzemeltetés és karbantartás,
- vezeték nélküli kapcsolatok alkalmazása olyan helyeken, ahol másfajta kommunikáció nem, vagy csak nagyon nehezen valósítható meg,
- távolról történő beavatkozási, esetleg konfigurálási lehetőség meghibásodás esetén.

Igazi technikai kihívásnak számított még a 90-es évek elején, amikor egy távoli, akár több ezer kilométerre lévő PLC-s irányító rendszer meghibásodásakor a fejlesztők diagnosztizálás céljából modemes kapcsolaton keresztül tudták elérni. Ez ma már az ipari Ethernet hálózatban, megfelelő jogosultságok birtokában bármikor megtehető.

Mindezen előnyök mellett természetesen feltehetjük a kérdést: Vajon jó dolog-e kitennünk a teljes gyártási folyamatot az Internetre? Természetesen ezt csak akkor tehetjük meg, ha biztosítani tudjuk a rendszer hatékony védelmét a támadások ellen. Az Ethernet hálózatok biztonságának kérdései mára már külön ágazattá fejlődtek. Számos olyan fejlett tűzfal-kialakítási, beléptetési (azonosítási és jogosultsági) és titkosítási eljárás létezik,

amelyek segítségével megakadályozható az illetéktelen behatolás és szinte teljes biztonságban tudhatjuk rendszereinket.

6. Az irodai Ethernet determinisztikussá tételének néhány elvi megoldása

Ahhoz, hogy determinisztikussá tegyük az Ethernet hálózatot, több szempontot is figyelembe kell vegyünk. Elsőként azt célszerű elérni, hogy valamilyen módon megkülönböztessük a valós idejű csomagokat a nem valós idejűektől. Ezen belül a determinisztikus csomagokat besorolni valamilyen prioritási szintbe, attól függően, hogy milyen valósidejű osztályt képviselnek. Számos elméleti és gyakorlat megoldás is született az idők során, amelyhez nagyban hozzájárult az utóbbi idők Ethernetes kommunikációnak nagyarányú műszaki fejlődése. Az alkalmazott megoldások két nagy csoportra oszthatók.

1. Az első csoportba azokat az eljárásokat sorolnám, amelyek az *ütközési valószínűség csökkentését* tűzték ki célul, vagyis lecsökkenteni az ütközési tartományt, korlátozni a backoff algoritmust, illetve prioritással ellátni a valós idejű csomagokat. Sokat segített a kapcsolt Ethernet kifejlesztése, azaz a hubok helyett a switchek használata, melynek következtében az ütközési tartomány mindössze két eszközre korlátozódik, ugyanakkor puffereiben a nem determinisztikus csomagok tárolhatók, amíg a valós idejűek továbbítása be nem fejeződik
2. A második csoportba azokat a megvalósításokat helyezem, amelyek valamilyen jól meghatározott algoritmus szerint vezérlik a buszhasználatot. Ilyenek például a ciklikus hozzáférési technikákat alkalmazó master-slave, vagy provider-consumer eljárások, vagy a token passing, illetve a token ring módszerek. Ugyancsak ide sorolnám az időosztásos megoldásokat is, amelyeket főleg a szigorúan valós idejű, szinkronizált kommunikációs kapcsolatoknál használnak. Itt is találunk szoftveres, illetve ASIC-t (Application Specific Integrated Circuit) használó hardveres megoldásokat.

6.1. Az ütközési valószínűséget csökkentő eljárások

Itt említeném meg az ORTE-t (Ocera Real-Time Ethernet), amely forgalommenedzselő alkalmazással igyekszik ütközésmentessé tenni a kommunikációt, vagy a forgalomfinomítási (Traffic Smooting) eljárást, amely a lyukas vödör elvét használja [14, 15].

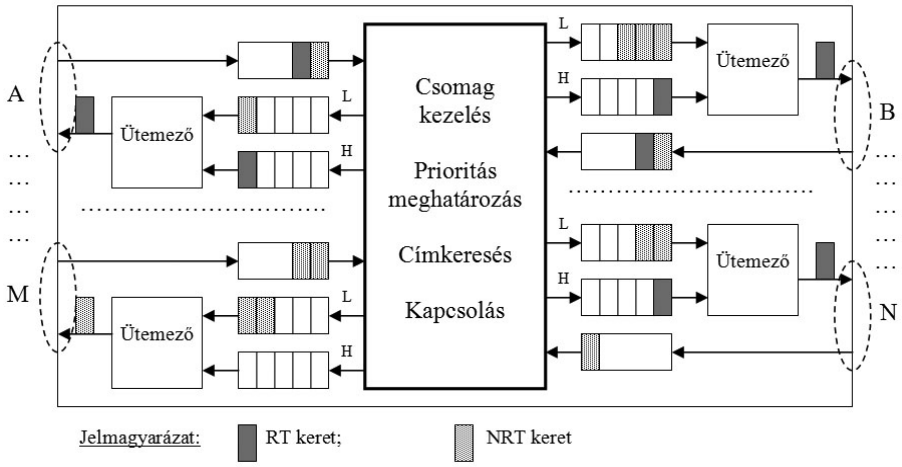
Lo Bello és társai [16] egy olyan megoldást fejlesztettek ki a dinamikus megközelítés kapcsán, mely a hálózati forgalmat az ütközések számával és az áteresztéssel jellemzi egy-egy adott intervallumon belül. A paramétereket ezután egy fuzzy szabályozóra küldik a hálózati bemeneti korlát beállítása végett. A módszerrel nagyon nagy hatékonyságot sikerült elérni a forgalomfinomítás terén.

Más megoldások a versengési backoff algoritmus befolyásolását helyezik előtérbe. Egy átlapoló mechanizmussal egészíti ki a CSMA/CD protokollt, még pedig olyan formában, hogy ütközéskor a valós idejű keretek nem egy 32 bájtos JAM-mel jelzik ezt, hanem egy hosszabb és a várakozási idővel arányos hosszúságú, de felső korláttal rendelkező úgynevezett *fekete burst*-öt (black burst) küldenek. A módszer kidolgozása J. L. Sobrinho és A. S. Krishnakumar nevéhez fűződik [17], akik igazolták, hogy ez determinisztikus működést tesz lehetővé osztott Ethernet hálózatokon. A módszer alkalmazásával megvalósítható, hogy ugyanazon a hálózaton a valós idejű keretek előnyt élvezzenek a nem valós idejűekkel szemben, sőt egymáshoz viszonyítva is attól függően, hogy melyik érkezett hamarabb. Azaz FCFS (First Come First Served) elsőnek érkezett elsőnek kiszolgált hozzáférést biztosít

6.2. Kapcsolt Ethernet

A kapcsolt Ethernet igénye akkor merült fel, amikor célul tűzték ki a globális áteresztőképesség javítását, a forgalom izolálását és az eredeti CSMA/CD választó mechanizmus nem-determinisztikus paraméterei okozta hatás csökkentését. Az Ethernet kapcsolók tulajdonképpen leszűkítik az ütközési tartományt az éppen aktuális kommunikációban résztvevő két állomásra, ha csak nem multicast vagy broadcast adás történik. Az állomások tehát nem csatlakoznak direkt módon egymáshoz. Képesek a kapcsolatot duplex módban bonyolítani, vagyis olyan mintha pont-pont kapcsolat épülne ki két állomás között. A legtöbb korszerű Ethernet kapcsoló a „tárol és továbbít" (store and forward) elvet használja, és ha ismeri az IEEE 802.1p, minőségi szolgáltatást (QoS) biztosító protokollt, akkor prioritási szintek szerint tudja pufferelni a kimeneti adatokat [18].

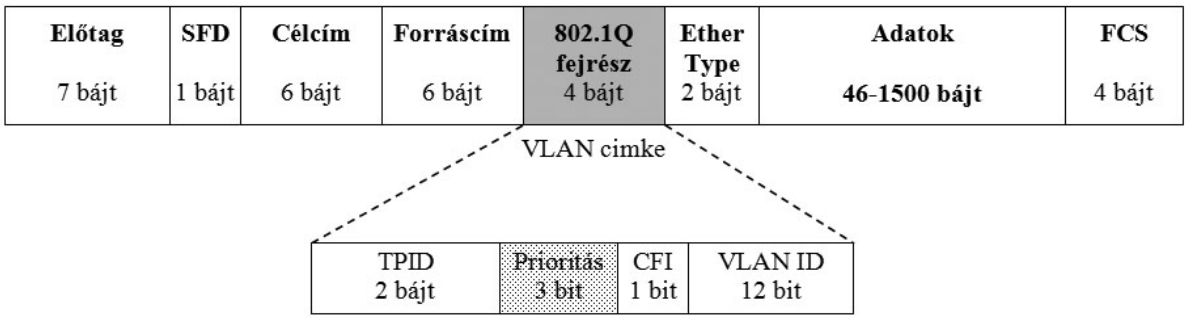
A legtöbb switch kettő, de a menedzselhető, főleg ipari Ethernet switchek akár négy prioritási szintet is tudnak kezelni. Amikor egy üzenet érkezik egy porthoz, akkor az letárolódik a bemeneti pufferbe, megállapításra kerül a prioritás és a cél, majd a prioritásnak megfelelő pufferébe továbbítódik, ahonnan az ütemező mindig a prioritási sorrendnek megfelelően helyezi a buszra. A következő ábrán (10. ábra) egy „tárol és továbbít", két prioritási szinttel rendelkező, duplex kapcsoló elvi felépítési vázlatát láthatjuk.



10. ábra. A Store and Forward elvű switch felépítési vázlata

Amikor az egyes portokon az üzenetek érkezési aránya (legyen bemeneti vagy kimeneti port) nagyobb, mint az elküldési arány, az üzenetek sorokban várakoznak. A legtöbb korszerű switch elég gyors az érkező üzenetek lekezeléséhez, így a bemeneti portokon nem alakul ki sor. A sorok a kimeneti portokon jelenhetnek meg, amikor rövid időn belül, egyszerre több port felől érkeznek üzenetek, ugyanarra a kimeneti portra. Ez esetben a sorban lévő üzenetek szekvenciálisan kerülnek továbbításra. Ezért a valós idejű ipari Ethernet hálózatban nélkülözhetetlen a párhuzamos, különböző prioritású szintekhez tartozó kimeneti, FCFS rendszerű pufferek használata.

Most már csak az a kérdés, hogy honnan ismeri fel a kapcsoló a valós idejű csomagokat? Az IEEE 802.1Q, virtuális LAN-okra vonatkozó szabványleírás szerinti adatkeretekben, az IEEE 802.1P-nek megfelelően, nyolc különböző prioritási szintet lehet definiálni.



11. ábra. A 802.1Q szabvány szerinti VLAN címkés keretformátum

Az előző ábrán (11. ábra) láthatjuk, hogy a 802.1Q keret tulajdonképpen Ethernet II keret, amely kiegészül egy 2x2 bájtos VLAN címkével, amelyben 3 bit áll rendelkezésünkre a prioritás meghatározására. A normál TCP/IP adatkeretek a legalacsonyabb 0 szintet kapják, míg a valós idejű keretek 5, 6 vagy a legmagasabb 7 szintű prioritást kapják.

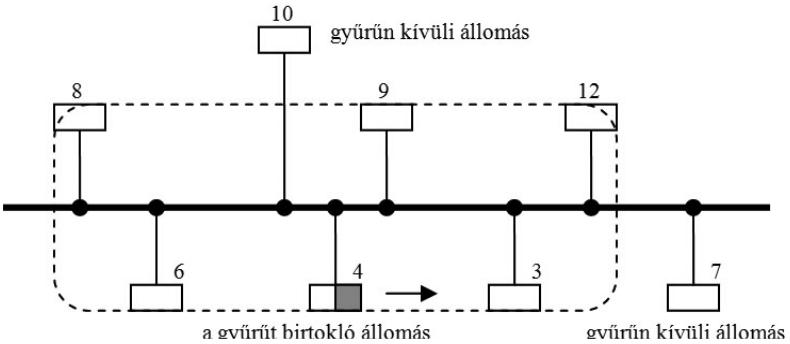
Első ránézésre azt gondolhatnánk, hogy a switch-ek alkalmazásával meg is oldottuk a valós idejű adattovábbítást az Ethernet hálózatban. Ez ugyan sokat javít a helyzeten, de korántsem elegendő. Az automatizált ipari berendezések irányítása a legtöbb esetben úgy valósul meg, hogy egy (vagy több) vezérlő nagyon sok, akár egymástól távol eső érzékelőktől kap adatokat, amelyek kiértékelése után ugyancsak ez a vezérlő továbbítja a beavatkozáshoz szükséges információt is. Az ilyen Termelő-Fogyasztó (Provider-Consumer) modellen alapuló kommunikáció esetében bizony előfordulhatnának olyan helyzetek, hogy egyszerre, egy időben több érzékelő eszköz Ethernetes illesztője elárasztja a vezérlő switchen belüli kimeneti portját. Vagyis szükséges lenne még valami, ami ütemezi ezt a feladatot.

Azt sem szabad elfelejtenünk, hogy a hálózati kapcsolók a csillag topológiát preferálják, az ipari berendezések és vezérlők pedig a legtöbbször busz, vagy gyűrű topológiát használnak. Mindezek mellett még bizonyos késleltetéssel is számolnunk kell, amely akár jelentős is lehet, főleg a szigorúan valós idejű rendszereknél.

Mindezek ellenére a hálózati kapcsolók alkalmazása nélkülözhetetlen a valós idejű ipari Ethernet világában, olyannyira, hogy nagyon sok ipari Ethernetes eszközt saját belső kapcsolóval látnak el, hogy bármilyen helyzetben alkalmas legyen további kommunikációs kapcsolat kialakítására.

6.3. Vezérjeles sín

Már a 802.3-as szabvány megjelenésének idején egyes gyártásautomatizálással foglalkozó szakemberek komoly fenntartással fogadták ezt a már ismert nem determinisztikus tulajdonságai miatt. Ezért egy egyszerű, kiszámítható legrosszabb esettel rendelkező rendszer kidolgozásába fogtak. Az elképzelés szerint az állomások gyűrűt alkotnak és egy vezérjel, a token körbejár a csomópontok között egy jól meghatározott sorrend alapján. Ezek szerint, ha n állomás vesz részt a gyűrűben, és Δt ideig (token tartózkodási idő) tart egy keret elküldése, akkor egyetlen egy állomásnak sem kell $n \cdot \Delta t$ időnél többet várakoznia egy keret elküldésére. A gyűrű topológia viszont fizikailag nem nagyon illeszkedik a gyártásautomatizálás egyenes vonalú pályájához. Ezért az ide vonatkozó 802.4-es szabványt, amelyet 1986-ban Dirvin és Miller indított útjára, úgy határozták meg, hogy az fizikailag egy lineáris, vagy fa topológiájú hálózat, amelyben az állomások vannak logikai gyűrűbe szervezve (12. ábra). A vezérjel a logikai gyűrű mentén körbe jár. Küldési joga csak a vezérjelet birtokló állomásnak van, ezért az ütközés kizárható.



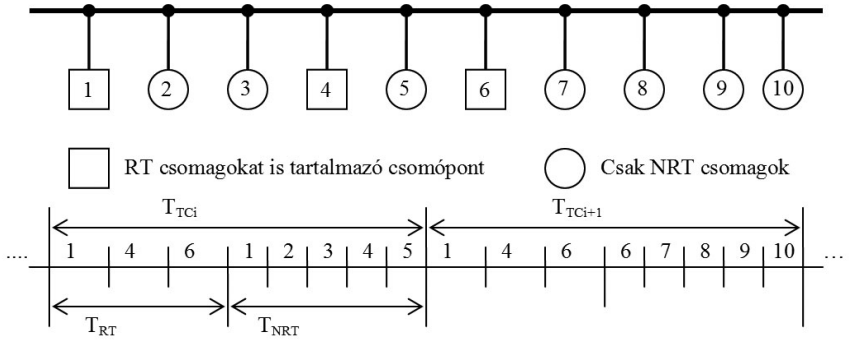
12. ábra. A vezérjeles sín elve

A gyűrűben résztvevő állomások fizikai sorrendje lényegtelen mivel adatszórásos közegben vannak. A lényeg az, hogy a tokent birtokló állomás az utána logikai sorrendben következő állomásnak a címét ismerje. Induláskor mindig a legmagasabb címmel rendelkező állomás kapja meg a vezérjelet, majd ezt követi csökkenő sorrendben egy alacsonyabb címmel rendelkező állomás. (12-9-8-6-4-3 a sorrend, a 10-es és a 7-es állomások a gyűrűn kívül vannak.) Azt, hogy melyik állomás vehet részt a gyűrűben a MAC protokoll határozza meg, sőt azt is, hogy engedélyt kap-e a kérelmező állomás a belépésre vagy a kilépésre.

A vezérjeles adattovábbítás protokollja négy prioritási szintet is enged definiálni: a 0-ást, a 2-est, a 4-est és a 6-ost. Ez virtuálisan olyan mintha minden állomás belül négy állomást tartalmazna, amelyek az adott prioritási osztályokhoz tartoznak, amelyek közül a legmagasabb prioritású, a 6-os van legfelül. A MAC alrétegbe érkező bemeneti csomagok prioritás szerint kerülnek a megfelelő szintre. Az állomáshoz érkező vezérjel először mindig a 6-os prioritási szintről veszi fel a keretet, ha van, majd a halad beljebb a 0-s szintig, amikor pedig letelt az állomáshoz rendelt idő, továbblép a következőhöz.

A token adott állomáshoz köthető tartózkodási idejét, a sávszélességen belül eloszthatjuk egyenlő arányban (szimmetrikusan) az állomások között, de aszimmetrikusan is, lehetővé téve ezáltal, hogy ugyanahhoz a csomóponthoz a token akár többször is eljusson egy körben járási ciklus alatt.

Az imént bemutatott vezérjeles továbbítási taktikát alkalmazzák a RETHER és a RTEP protokollok.



13. ábra. A RETHER protokoll kerettovábbítási stratégiája.

A RETHER esetében a token körbejárás periodikus. CSMA/CD üzemmódban van mindaddig, míg nem érkezik RT kommunikációs kérés, ekkor átvált token üzemmódba és végig megy mindazokon a csomópontokon, amelyek RT üzenetet akarnak továbbítani, majd ha marad elég ideje a cikluson belül, akkor felvesz még NRT kereteket is (13. ábra). Ha már nincs több RT kérés, akkor visszakapcsol CSMA/CD módba [19, 20].

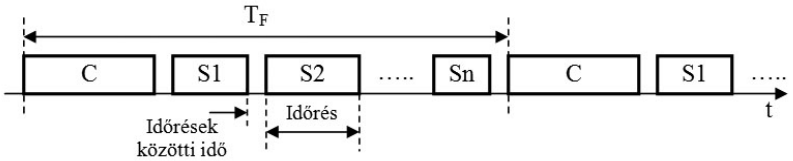
Az RTEP estében a közeghez való hozzáférés két fázisban történik: kiválasztás és üzenettovábbítás. A token először az összes csomóponton körbejár, hogy megkeresse a legnagyobb prioritású üzenetet, majd ezt követően közvetlenül ahhoz a csomóponthoz megy, amely a legnagyobb prioritású üzenetet tartalmazza. A valós idejű csomagok az Ethernet keret adatmezejében kerülnek továbbításra. Az átvitel után a csomópont egy új kiválasztási fázist indít. Ha a kiválasztási keretbe beírt prioritásnál nagyobbat talál, akkor az felülírja a keretben lévőt. A protokoll tartalmaz hibatolerancia mechanizmust is, amely segíti a sérült keretek helyreállítását vagy újraküldését [21].

6.4. Időosztásos többszörös hozzáférés (Time-Division Multiple Access, TDMA)

Egy másik jól bevált technika az osztott kommunikációs hálózaton, ha ciklikus ütemezés mellett, kizárólagos időréseket (slot time) rendelünk a különböző adatforrásokhoz. Ez a jól ismert időosztásos többszörös hozzáférés (TDMA), mely tartalmaz egy globális szinkron keretet azért, hogy az összes csomópont egyezségre jusson az adott átviteli időréssel kapcsoltba. Ez a megoldás egy ütközésmentes közeg-hozzáférési protokollt eredményez, amelyet kezdetben a nagysebességű autóipari alkalmazások protokolljaként (TTP/C, TTCAN) [29, 30], majd a későbbiekben az osztott Ethernet legmagasabb, C osztály szerinti szintjén is alkalmazzák, felülbírálva ezzel az eredeti CSMA/CD mechanizmust.

Az időrés lehet fix, vagy változó szélességű. A jobb sávszélesség kihasználása érdekében, természetesen a dinamikus, változó méretű időrések alkalmazása válik előnyössé. Minden egyes időrés alatt a csomópontokban a feladatok ütemezése megtörténik, megakadályozva ezzel a versengést a buszhozzáféréshez. A konfigurálás során lehetőség van újabb csomópontok hozzáadására vagy elvételére. Ennek alapján meghatározható a keretidő (T_F), amelyet ha fix értéken tartunk, akkor kiszámítható, pontos valós idejű működést eredményez.

Minden egyes keret egy kontroll (C) időréssel kezdődik, amely főleg szinkronizálási információkat tartalmaz, majd ezt követik a csomópontokhoz rendelt S_i időrések, amelyek között egy-egy rövid, szabad időköz van hagyva (14. ábra).



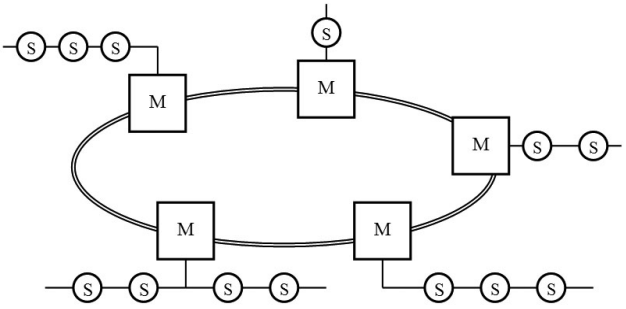
14. ábra. TDMA kertstruktúra

A módszer alkalmazásával lehetőség nyílik a nagysebességű kommunikációs kapcsolatok szinkronizálására, kiszámítható algoritmusok bevezetésére, valamint valós idejű és nem valós idejű információk azonos időben történő ütközésmentes továbbítására.

6.5. Master-Slave alapú technikák

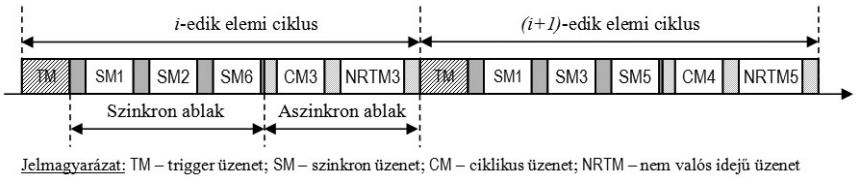
Az egyik legegyszerűbb módja a valós idejű kommunikáció megvalósításának a master-slave megoldás. A közeghozzáférést egy kitüntetett szereppel ellátott csomópont, a master vezérli. Az összes többi, hozzárendelt csomópontok a slavek. Ezek csak a master által kezdeményezett kapcsolatok során kommunikálhatnak. A master-slave kapcsolat egyfajta hálózati hierarchia szintet határoz meg. A masterek a hierarchia valamelyik magasabb szintjén egyenrangú partnerekként szerepelnek. Közöttük legtöbbször gyűrű topológia szerinti tokenes kommunikáció épül fel. Ezek alá vannak rendelve a slave egységek. Ezek lehetnek ugyanarra a buszra kötve, mint a masterek, de jellemzően inkább a felső szintűtől független

kommunikációs vonal köti a masterhez (15. ábra). A tokent birtokló master üzeneteket küldhet a slaveknek vagy fogadhat ezektől.



15. ábra. Master-Slave kommunikáció hierarchiája

Az előbbi eljárás egyik módosított változata az FTT Ethernet protokoll, amely ipari Ethernet hálózatra lett kifejlesztve. Tulajdonképpen egy módosított master-slave technikát használ, amely lecsökkenti a protokoll kommunikációs túlterheltségét. A buszhozzáférési időt fix hosszúságú, úgynevezett elemi ciklusokra osztják fel. Ezek további két részre tagolódnak: a szinkron és az aszinkron ablakra (16. ábra), melyek különböző karakterisztikával rendelkeznek [31].



16. ábra. Az elemi ciklusok szervezése

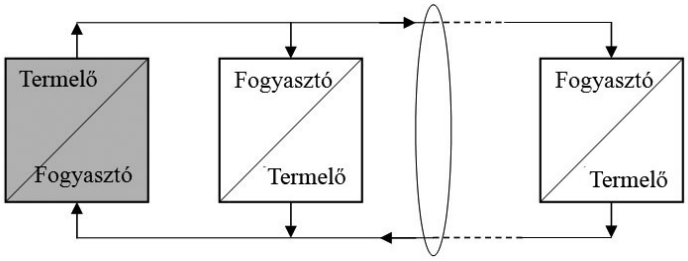
A szinkron ablak a master által ütemezett periodikus, idővezérelt forgalmat szállít. Az idővezérelt kifejezés azt jelenti, hogy a forgalom egy közös időalaphoz szinkronizált, melyet ez esetben a master szolgáltat. Az aszinkron ablak ciklikus, eseményvezérelt üzenetekhez kötődő információkat, illetve nem valós idejű forgalmat továbbítja. A két fázis között egy elég jelentős időbeli elkülönülés van jelen, hogy az eseményvezérelt forgalom ne ütközzön az idővezérelt forgalommal.

6.6. Termelő-fogyasztó (Provider-Consumer) modell

Az ipari Ethernet rendszerek talán legfejlettebb valós idejű kommunikációs megoldásának tekinthetjük. Működési elve hasonlít a master-slave kommunikációhoz, azzal a különbséggel, hogy full duplex összeköttetést feltételez és itt mindkét fél, akár egy időben is kezdeményezhet kommunikációs kapcsolatot. Az termelő rendszerint adatokat gyűjt az érzékelők felől vagy saját adatbázisából veszi elő és továbbítja a fogyasztó felé. A fogyasztó a kapott adatokat feldolgozza és/vagy továbbítja beavatkozók felé. A kapcsolatot kezdeményezheti bármelyik fél. Ebből a szempontból a felek egyenrangúaknak tekinthetők. A felek bármikor szerepet cserélhetnek, attól függően, hogy mikor milyen feladatot kell ellássanak.

Konfiguráláskor az eszközök között úgynevezett kommunikációs relációk épülnek fel, amelyeket bármikor meglehet nyitni, amikor adatátvitelt akar kezdeményezni valamelyik fél, illetve le lehet zárni, amikor az adattovábbítás befejeződik. A leggyakrabban alkalmazott megoldás az egy termelő, több fogyasztó, illetve a több termelő egy fogyasztó struktúra (14. ábra). Ezt a megoldást alkalmazzák például a Profinet IO rendszereknél [32], ahol egy PLC-

hez több IO-eszköz kapcsolódik. A PLC is és az eszközök is egyaránt lehetnek termelő illetve fogyasztó szerepben is.



17. ábra. Termelő-fogyasztó modell

Az előző alfejezetekben bemutatott elvi megoldások egy része soha sem került ipari alkalmazásra, más része már elavult (például a vezérjeles gyűrű), de jelentős szerepet játszottak a jelenlegi, korszerű Ethernet alapú ipari kommunikációs rendszerek kifejlesztésében.

Irodalomjegyzék

[1] IEEE 802.3 Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, 2008.

[2] Hosszú Gábor: Az internetes kommunikáció alapjai, ISBN 963-9442-51-8, 2005.

[3] Ajtonyi István: Ipari kommunikációs rendszerek I., Aut-Info Kft. Miskolc, 2008.

[4] Andrew S. Tanenbaum: Számítógép hálózatok, ISBN 978-963-545-529-4, 2011

[5] RFC 793, Transmission Control Protocol, 1981.

[6] RFC 768, User Datagram Protocol

[7] RFC 826, An Ethernet Address Resolution Protocol, 1982.

[8] RFC 1918, Address allocations for privat Internets, 1996.

[9] Perry S. Marshal, John Rinaldi: Industrial Ethernet, 2. edition, ISBN 1-55617-892-1, 2005.

[10] Petrényi József: TCP/IP alapok, 2009

[11] Hartványi Tamás, Kovácsházi Tamás: Számítógép hálózatok, 2001.

[12] Kovács Szilveszter: A hálózattervezés alapjai, 2007.

[13] RFC 894, Charles Hornig: A Standard for the Transmission of IP Datagrams over Ethernet Networks, 1984.

[14] Ajtonyi István: Ipari kommunikációs rendszerek III., Aut-Info Kft. Miskolc, 2009.

[15] Seok-Kyu Kweon, Kang G. Shin: Statical Real-Time communication over Ethernet for Manufactoring Automation Systems, 1999.

[16] Lucia LoBello, Giordano Kaczynski, Oratio Mirabella: Improving the Real-Time Behavior of Ethernet Networks Using Traffic Smoothing, 2005.

[17] J. L. Sobrinho, A. S. Krishnakumar: EQuB-Ethernet quality of service using black burst, 2006.

[18] Q. Zang, W. Zang: Priority Scheduling in Switched Industrial Ethernet, 2005.

[19] Chitra Venkatramani: The design Implementation and Evaluation of RETHER: Real-Time Ethernet Protocol, PhD dissertation, 1996.

[20] Chitra Venkatramani, Tzi-cker Chiueh: Design, Implementation, and Evaluation of a Software-based Real-Time Ethernet Protocol, 1995.

[21] I. L. Badiola, I. Radovanović, G. Russello, M. Chaudron: Design and implementation of a Real-Time protocol over Ethernet, 2004.

[22] H. Hoang, G. Buttazzo, M. Jonsson, S. Karlsson: Computing the Minimum EDF Feasible Deadline in Periodic Systems, RTCSA 2006.

[23] Jane W. S. Liu: Real-Time Systems, Prentice Hall, 2000

[24] Tei-Wei Kuo: Introduction to Real-time Process scheduling, National Taiwan University http://www.csie.ntu.edu.tw/~ktw/rts/ch-short-course-uni.pdf

[25] C. L. Liu, J. W. Layland: Scheduling algorithm for multiprograming in a Hard Real-time Environments, Journal of ACM, 1973.

[26] S. K. Baruah, A. K. Mok, L. E. Rosier: Preemtively scheduling Hard Real-Time sporadic task on one processor, IEEE Real-Time System Symposium, 1990.

[27] Kondorosi Károly, Szirmay-Kalos László, László Zoltán: *Objektum orientált szoftverfejlesztés,* 5. fejezet, http://www.tankonyvtar.hu/hu/tartalom/tkt/objektum-orientalt/ch05.html#id517610

[28] Ajtonyi I., Gyuricza I., Programozható irányítóberendezések és hálózatok, 2002.

[29] Ross Bannatyne: Time Triggered Protocol: TTP/C, Embended System Programming, march 1999.

[30] B. Müller, T. Führer, F. Hartwich, R. Hugel, H. Weiler, Fault tolerant TTCAN networks, http://www.canopen.org/fileadmin/cia/files/icc/8/mueller.pdf

[31] P. Pedreiras, L. Almeida, P. Gai, G. Buttazzo: FTT-Ethernet, a platform to implement the Elastic Task Model over message streams, http://www.ieeta.pt/lse/ftt/pub/wfcs02_ftt-eth_elas_task_model.pdf

[32] Raimond Pigan, Mark Metter: Automating with Profinet (2006)