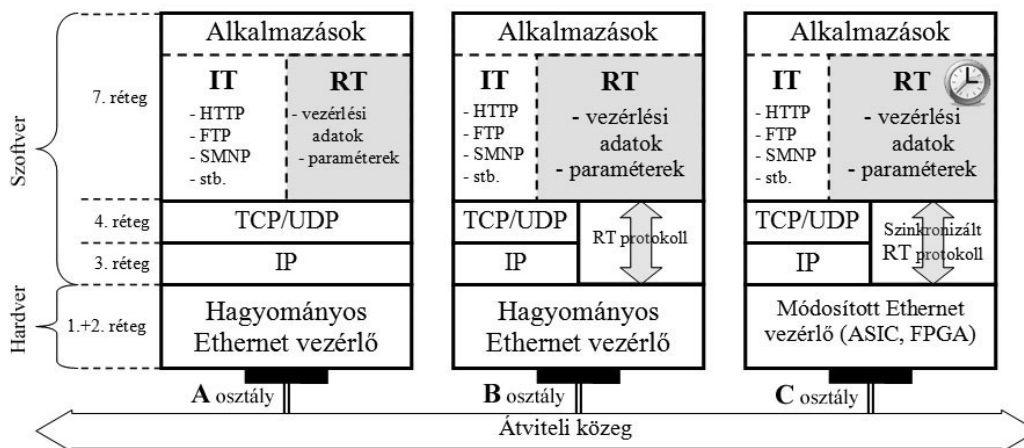


IPARI ETHERNET RENDSZEREK

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

Az ipari irányító- és kommunikációs rendszerek gyártói számos ipari Ethernet technológiára épülő megoldást fejlesztettek ki, amelyek jelentős szerepet játszanak a mai, korszerű ipari elosztott vezérlési és folyamatirányítási alkalmazásokban. A hagyományos terepi buszos kommunikáció korlátozott adatátviteli sebessége gátat szab azon korszerű technológiai megoldások alkalmazásának, amelyek nagyobb információmennyiséget és átviteli sebességet igényelnek. A közös Ethernet platform amellett, hogy kétirányú, 100 Mb/s-os sebességet képes szolgáltatni, egy lépéssel közelebb hozta egymáshoz a különböző gyártók specifikus termékeit. Ezek most már ha nem is közvetlenül, de megfelelő eszközeik segítségével implementálhatók a legtöbb rendszerbe. Ez a feladat persze nem olyan egyszerű, mert a gyártók meglehetősen kevés műszaki információt közölnek a termékeik ilyen jellegű alkalmazásaival kapcsolatban.

A jelenleg elterjedt megoldásokat a gyakorlatban három osztályba sorolhatjuk, attól függően, hogy milyen kommunikációs rétegeket használnak és milyen valós idejű követelményeket képesek teljesíteni. Az 1. ábrán ezen osztályok rétegstruktúráit láthatjuk.



1. ábra. Ipari Ethernet osztályok struktúrái.

Az *A osztályú* ipari Ethernet gyakorlatilag ugyanazt a hardver felépítést és TCP/IP vagy UDP/IP protokollkészletet használja, mint az irodai Ethernet. A valós idejű követelményeket az RT alkalmazások a TCP/IP rétegek felett valósítják meg. Az ütközések elkerülése a valós idejű alkalmazásokban szoftveres úton forgalomfinomítással, vagy kapcsolt Etherneten keresztül történik. Meglehetősen korlátozott valós idejű működést képes biztosítani, általában 50-100 ms-os kézbesítési idővel és legalább ekkora jitterrel. Alkalmazási területe: Főleg monitoring rendszerek, folyamatirányítás, épületautomatizálás, stb. Ebbe az osztályba sorolhatók például a hagyományos Ethernet/IP, a Modbus/TCP vagy a Profinet CBA. [1, 2, 3]

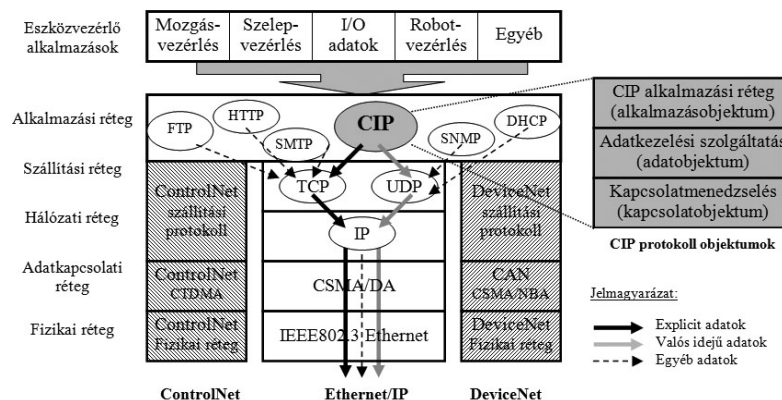
A *B osztály* ugyancsak a szabványos Ethernet hardveres felületet használja, de már jobb valós idejű támogatást biztosít, egyrészt mert a QoS szerint a prioritással ellátott RT csomagokat képes előnyben részesíteni a nem valós idejű csomagokkal szemben, másrészt az egyes RT protokollok képesek kezelni közvetlenül az alsóbb rétegek felől érkező kereteket. Ilyenkor azonban a kommunikáció csak egy szegmensen belül valósítható meg. Az ide tartozó rendszerek általában kielégítik a gyártásautomatizálás követelményeit: 5-10 ms-os kézbesítési idők, 10-20 ms-os válaszidők, 2-4 ms jitter. Főleg a PLC-s vezérlésű DCS alapú ipari automatizálás terén alkalmazzák. Az osztály egyik meghatározó megoldása a Siemens támogatottságú Profinet IO vagy a Bernecker & Rainer cég által fejlesztett Ethernet Powerlink.

A harmadik, *C osztályba* a legigényesebb valós idejű működést igénylő hajtásszabályozás és mozgásvezérlés tartozik, amely rendszerint szinkronizált, ciklikus működést feltételez. A ciklusidő 1 ms, vagy ennek tört részei (0,5 ms, 0,25 ms), a jitter pedig a legtöbb esetben kisebb, mint 1 µs. Mindenképpen 100Mb/s-os Full duplex Ethernet hálózatot feltételez. A hardverfelület módosított (nem szabványos Ethernet), speciális ASIC-t vagy FPGA-t tartalmaz, amelyben a kapcsoló a TDMA technikát alkalmazva mindig szabad utat biztosít a szigorúan valós idejű kereteknek, ugyanakkor a fennmaradó időben a nem valós idejű forgalom is biztosított. A szinkronizált működést az adatkapcsolati rétegbe implementált IEEE 1588 PTP (Precision Time Protocol) protokoll [4] biztosítja. Az eszközöket a MAC címek alapján azonosítják, routolás itt sem valósítható meg. A csoport meghatározó képviselője a Backhoff cég által fejlesztett EtherCAT, de jelentős részt foglal el a Profinet IRT és a Sercos III is.

A fejezet további részében mindegyik osztályból néhány gyakrabban alkalmazott, elterjedtebb megoldást fogok bemutatni.

1. Az Ethernet/IP

Az Ethernet/IP protokoll eredetileg a Rockwell cég által került bevezetésre 2001-ben, mint az akkori idők egyik legkorszerűbb és legjobb valós idejű adottságokkal rendelkező ipari Ethernet rendszere. Az IEC 61158 szabvány valós idejű követelményeit, az IEC 61784-1 szabvány ipari Ethernetes specifikációjának megfelelően valósítja meg. Főleg a tengerentúlon, Észak-Amerikában valamint Ázsiában terjedt el. Jelenleg az ODVA (Open DeviceNet Vendors Association) menedzseli, melynek köszönhetően a valós idejű működésének alapját képező CIP (Common Industrial Protocol) protokollja szerepel más ipari Ethernetes megoldásokba implementálva is, mint például a ControlNet vagy a DeviceNet és újabban a CompoNet. (2. ábra) [4].



2. ábra. A CIP protokollon alapuló Ethernet/IP rétegstruktúrája

1.1. A CIP protokoll

Valós idejű működés elsősorban az alkalmazási rétegbe implementált CIP protokoll objektumorientált funkcióinak köszönhető. Ezek, funkcióikat tekintve, három szintre bonthatók a következők szerint:

1. Üzenetirányítás és kapcsolatmenedzselés (kapcsolatobjektum),
2. Adatkezelési szolgáltatás (adatobjektum),
3. Objektum orientált alkalmazáskezelés (alkalmazásobjektum).

A CIP protokoll beágyazása az alkalmazási rétegbe (2. ábra) nem befolyásolja a többi, már jól ismert protokoll működését, megőrizvén ezáltal az Ethernet hálózat nyitottságát.

Az eszközvezérlő alkalmazások üzeneteit az üzenetirányító objektum a létrejövő kapcsolat rendeltetése szerint két irányba továbbítja. Ennek megfelelően a CIP protokollt használó eszközök alapvetően kétféle üzenetküldési módot használnak:

- *Nem kapcsolt kommunikáció (Unconnected Messaging)* – alacsony prioritású, nem rendszeresen előforduló kapcsolat-felvételi procedúrák kezelésére. Pl. konfigurációs beállítások során keletkező adatforgalom, vagy lekérdezések alkalmával létrejövő kapcsolatok. Ezek az üzenetek a TCP/IP csomagok adatmezőjébe ágyazódnak be és úgy kerülnek továbbításra, mint bármilyen szabványos Ethernet üzenet.
- *Kapcsolt kommunikáció (Connected Messaging)* – elsősorban valós idejű, rendszeresen (ciklikusan) továbbítandó vezérlési adatok továbbítására. A kommunikációt a Kapcsolatmenedzser irányítja, amely miután létre jött egy kapcsolat a csomópontok között, képes szétválasztani az explicit és implicit üzeneteket. Ezeket aztán az Üzenetirányító a megfelelő típusú TCP illetve UDP szállítórétegek felé irányítja.

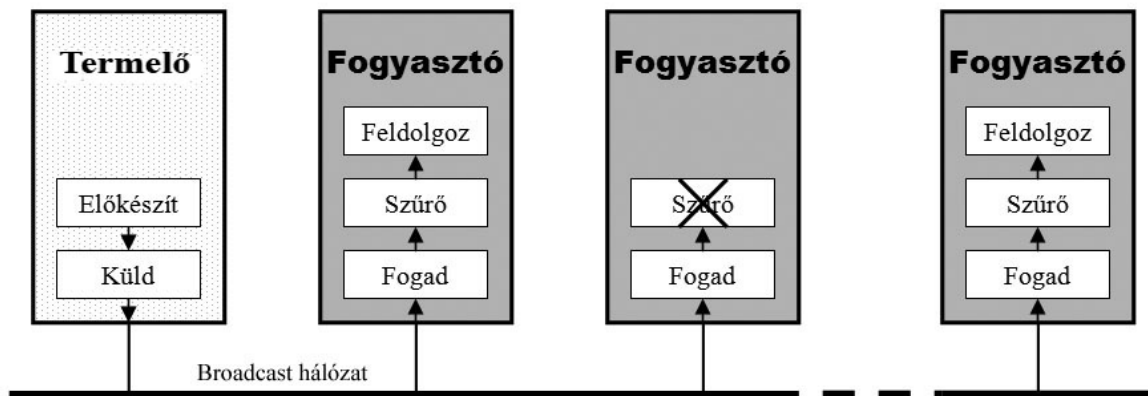
Az imént említett üzenet szétválasztásnak köszönhetően az összes Ethernet/IP hálózati kommunikáció e két üzenettípus szerinti kapcsolaton alapul:

- *Explicit üzenet kapcsolat (Explicit Messaging Connection)* – általános célú pont-pont kapcsolat két eszköz között. Ezek a kapcsolatok kivétel nélkül az Üzenetirányítón (Message Router) keresztül a TCP/IP szolgáltatás adta lehetőségeket használják az adatcsere megvalósítására. Legtöbbször „kérés-válasz” jellegűek ezek a kapcsolatok. Az explicit kérést a vevő dekódolja, értelmezi, majd ugyancsak explicit választ generál.
- *Implicit I/O adat kapcsolat (Implicit I/O data Connection)* – akkor jön létre, amikor valós idejű, alkalmazás specifikus I/O adatokat kell továbbítani az erre a célra létrehozott kapcsolat során. Ezek az adatok szabályos időközönkénti, ciklikus jellegű üzenetek. A legtöbb esetben termelő-fogyasztó modell szerinti pont-multipont kétirányú kapcsolat épül fel a felek között. Az ilyen típusú üzenetek számára mindig az UDP/IP típusú hálózati protokoll szerinti kapcsolat épül ki és fennmarad mindaddig, míg a kezdeményező el nem bontja.

1.2. Az Ethernet/IP kapcsolat modellje

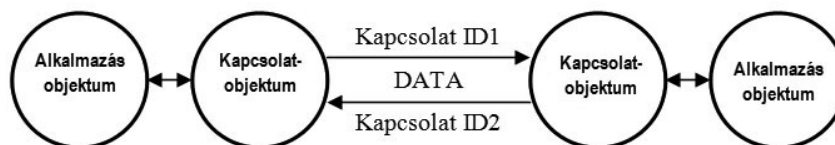
Az Ethernet/IP az eszközök közötti kommunikáció számára a sok szempontból előnyös termelő-fogyasztó modellt alkalmazza. Az esetek többségében ez úgy valósul meg, hogy egy termelő több fogyasztót szolgál ki. Hátránya ennek a megoldásnak, hogy broadcast kommunikációt feltételez, így a fogyasztónak ki kell szűrnie a számára szükségtelen üzenetet. (3. ábra).

Ha megváltoztatjuk a kommunikációs kapcsolatokat leíró attribútumokat, akkor olyan kapcsolat is létre hozható, amikor több termelő szolgál ki egy fogyasztót. Full duplex hálózatban, a két kapcsolat akár egyszerre, egy időben is megvalósítható.



3. ábra. Termelő-fogyasztó modell szerinti Ethernet/IP kapcsolat

A kommunikációs kapcsolat kiépítéséhez, a kapcsolatban résztvevő állomások két-két objektumának együttműködésére van szükség: az egyik az alkalmazásobjektum a másik pedig a kapcsolatobjektum (4. ábra). Ha az állomások egyenrangúak, az alkalmazás jellegétől függően a kapcsolatfelvételt bármelyik fél kezdeményezheti. Az alkalmazásobjektumban az eszközök azonosítására szolgáló gyártó-specifikus adatok szerepelnek, mint például a gyártó ID, az eszköz típusa, sorozatszáma, neve, stb. Ha a kapcsolatfelvétel során megszólított állomás úgy látja, hogy képes az illető eszközzel kommunikálni, akkor elfogadja a kapcsolatot. [5]



4. ábra. Kapcsolatfelvétel két eszköz között

A kapcsolatobjektum feladata meghatározni, hogy az adott alkalmazás szerint milyen típusú kapcsolatra van szükség, mert ennek megfelelően kell kiépíteni a kapcsolatot. Mindegyik kapcsolathoz hozzárendel egy jól meghatározott kapcsolatazonosítót (Connection ID). Ha kétirányú kapcsolatra van szükség, akkor kapcsolatazonosítóból is kettőt kell létrehozni. Amikor egy kapcsolat létrejött, akkor a kommunikációhoz szükséges összes erőforrás, beleértve a CIP üzenetirányítót is, lefoglalásra kerül, így stabil összeköttetés biztosítható.

Az Ethernet/IP hálózat eszközeit, rendeletetésük és betöltött funkcióik szerint három osztályba csoportosíthatjuk: üzenet- (Messaging Class), adapter- (Adapter Class) és szkennerek osztály (Scanner Class).

1. Az üzenet osztályba tartozó eszközök kizárólag csak explicit üzenetforgalmat (kapcsolt vagy nem kapcsolt kommunikáció szerint) bonyolítanak, valós idejű I/O adatforgalmat nem. Ide sorolhatók például a programozó eszközök, amelyek a PLC-k és HMI eszközök vezérlési programjait töltik fel vagy le.
2. Az adapterek tulajdonképpen távoli I/O eszközvezérlők, amelyek válaszolnak, vagy végrehajtják a szkennerek kéréseit. Nem kezdeményeznek adatkapcsolatot, csak akkor küldik vagy fogadják a valós idejű I/O adatokat, amikor a szkennerek erre felkéri őket. Explicit üzeneteket is fogadhatnak, más osztálybeli eszközöktől, például a programozó eszköztől, de ilyen jellegű kapcsolatot sem képesek kezdeményezni.

3. A szkennerek osztályba tartoznak mindazok az eszközök, amelyek a kapcsolatot kezdeményezhetik az adapterekkel, vagy más szkennerekkel, és mindkét alapfunkciót betölthetik, vagyis termelő illetve fogyasztó szerepük is lehet. A legtöbb esetben erre a célra PLC-eket, robotvezérlőket, stb. használnak.

1.3. A CIP protokollt használó Ethernet/IP legfontosabb jellemzői

A Ethernet/IP rendszereknél a kerettovábbítási kapacitás (Packet Rate Capacity – $C_{P/S}$) az egyik legfontosabb meghatározó érték, amelynek segítségével jellemezni tudjuk a hálózat képességeit. Ennek segítségével tudjuk meghatározni a csomópontok számát (N), amelyet kezelni tud az illető rendszer. Ugyancsak ettől az értéktől függ az implicit I/O adattovábbítás buszfrissítési ideje, azaz ciklusideje. Ezt az egyébként nagyon fontos alaptényezőt az Ethernet/IP rendszerben csomagküldési intervallumnak (Requested Packet Interval – RPI) nevezik. Kétirányú I/O kapcsolatnál a ciklusidő számított, minimális átlagértékét a következő összefüggéssel határozhatjuk meg:

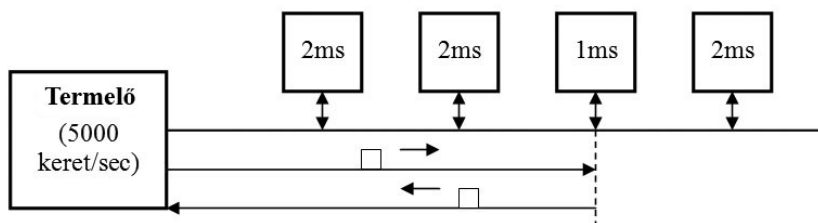
$$RPI = \frac{2 \cdot N}{C_{P/S}} \quad (1)$$

A következő táblázatban a ciklusidő értékeket láthatjuk, a csomópontok számától függően, különböző osztályú szkennerek esetében [5].

1. táblázat. Az RPI változása a csomópontok szerint

Csomópontok száma (N)	RPI [ms]		
	5000 keret/s alap szkennerek	10000 keret/s SRT szkennerek	25000 keret/s HRT szkennerek
4	1,6	0,8	0,32
8	3,2	1,6	0,64
16	6,4	3,2	1,28
32	12,8	6,4	2,56
64	25,6	12,8	5,12

A valóságban a számított ciklusidő értékeknél kisebb ciklusidő is beállítható egyes csomópontokban akkor, ha a többi csomópontok eszközei nagyobb ciklusidőt igényelnek. Például a Rockwell ControlLogix sorozat 1756-ENBT 5000 keret/sec-os szkennerek N = 4 csomóponttal konfigurálható úgy, hogy egy csomópont 1 ms-os, három csomópont pedig 2 ms-os ciklusidővel rendelkezzen (5. ábra) [4].



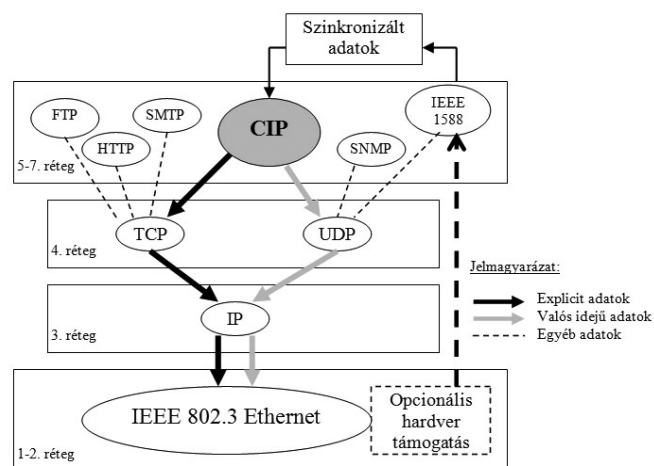
5. ábra. Különböző ciklusidővel rendelkező kapcsolatok

A három 2 ms-os ciklusidejű csomópont kiszolgálása az (1) szerint másodpercenként 3000 keretet igényel, így még marad egy 2000 keret/sec-os kapacitás, ami pontosan egy 1ms-os fogyasztót tud kiszolgálni.

Az explicit és implicit üzenetek kapcsolatainak száma is korlátozott. A legtöbb Rockwell Ethernet/IP eszköz maximum 64 eszközzel tud létesíteni TCP kapcsolatot. CIP kapcsolatot viszont ennél jóval többet, például a 1756-ENBT 128-at, a 1756-EN2T pedig 256-ot, azért mert egy TCP kapcsolat alatt több CIP kapcsolat is létrehozható [4].

1.4. A hatékonyság növelése PTP protokollal

Az Ethernet/IP rendszer valós idejű teljesítményének javítása érdekében a Rockwell bevezette a CIP Sync implementációt, amely szinkronizált adatátvitelt biztosít az eszközök között. Lényege, hogy az IEEE 1588 PTP (Precision Time Protocol) protokollt alkalmazva az eszközök órajelei szinkronizálva lesznek egy meghatározott master órához képest, amely akár az termelő órája is lehet [4]. A ciklusidőt ugyan nem csökkenti a CIP Sync, de a jittert igen, amely viszont a szinkronizálásnak köszönhetően már jelentős válaszügy csökkenést eredményezhet [6]. A CIP Sync tulajdonképpen a CIP protokoll egy módosított változata, amely a szinkronizálást az alkalmazási réteg szintjén valósítja meg, de opcionálisan az IEEE 1588 hardvertámogatottsága is felhasználható, ezáltal még jobb valós idejű teljesítmény érhető el. (6. ábra).



6. ábra. A PTP protokoll implementálása az alkalmazási rétegbe

A CIP Sync alkalmazásával az Ethernet/IP A osztályú besorolásból egyfajta C osztályba lép elő, vagyis alkalmassá válik szinkronizált hajtásvezérlésre, de még nem alkalmas szervóhajtások szabályozóinak vezérlésére, ugyanis a ciklusidő még mindig elég magas, tipikusan 5-10 ms a reakció idő pedig 15-30 ms körüli [5].

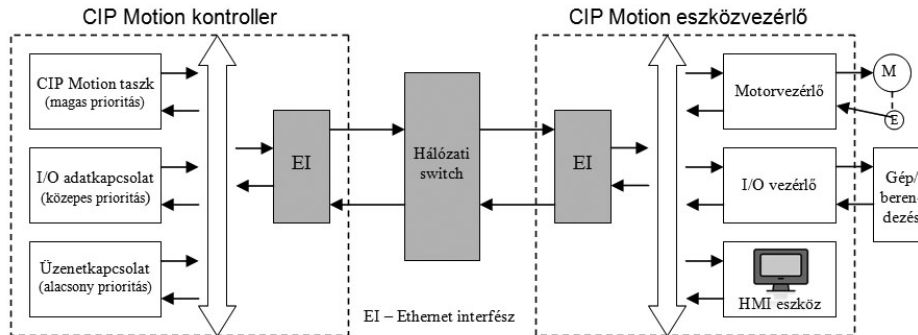
1.5. CIP mozgásvezérlő (CIP Motion) lehetőség implementálása

Az ODVA fejlesztései alapján, a Rockwell cég 2009-ben mutatta be az első olyan Ethernet/IP megoldást, amely már képes többtengelyes szervomotoros és frekvenciaváltós mozgásvezérlők ipari Ethernetes adatkapcsolatának bonyolítására. A CIP Motion kettős feladatot lát el: egyrészt biztosítja a mozgásvezérlő eszköz interfész felületéhez az eszközvezérlőt, vagyis azt, hogy a mozgásvezérlő összekapcsolható legyen a frekvenciaváltóval vagy a szervó vezérlővel, másrészt a tengelyvezérlési funkciókat támogatja.

A CIP Motion implementálásával az Ethernet/IP alkalmassá vált mindhárom ipari kommunikációs osztály valós idejű követelményeinek kielégítésére. Ezekhez az osztályokhoz prioritási szinteket rendel az alábbiak szerint:

- mozgásvezérlés (CIP Motion objektum) – magas prioritás,
- I/O adatkapcsolat – közepes prioritás,
- üzenetkapcsolat – alacsony prioritás.

Az ODVA, a prioritással kitüntetett csomagok számára a VLAN címkézett keretek helyett a DSCP-t (Differentiated Service Code Protocol) alkalmazza, amely az IP fejléc Szolgálat típus (ToS) mezőjébe írja be a prioritási szinteket. A 7. ábrán egy ilyen megvalósítás vázlatja látható.

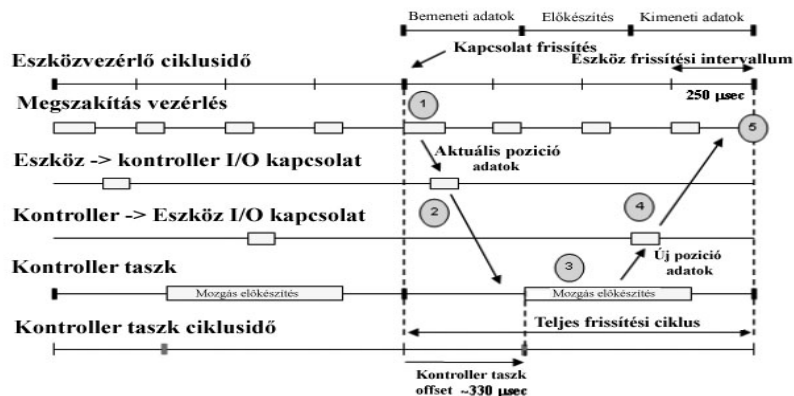


7. ábra. Teljes Ethernet/IP vezérlés

A CIP Motion controller hajtásvezérlő taszkja szinkronizált módon, ciklikusan kapja az aktuális helyzetadatokat a tengelyek pozícióról, majd kiszámolja az új értékeket és továbbítja az eszközevélő felé. Mindez az Ethernet hálózaton keresztül történik szabványos UDP/IP csomagok formájában. A magas prioritási szintnek köszönhetően ezek az adatok mindig előnyt élveznek a többi adattal szemben, így ütközés nem fordulhat elő. A teljes frissítési ciklus három részre tagolódik:

- bemeneti adatok fogadása,
- az új vezérlési adatok előkészítése,
- kimeneti adatok küldése.

Ha 1 ms-os teljes buszfrissítési ciklusidőt feltételezünk, akkor ez négy, 250 μ s-os megszakítási intervallummal biztosítható. A megfelelő szinkronizálás biztosításával (CIP Motion controller taszk offset 330 μ s) elérhető, hogy az új pozíció adatainak előkészítése még az eszközfrissítési intervallum előtt megtörténjen. (8. ábra) [1].



8. ábra. CIP Motion buszfrissítési ciklus időviszonyai.

Az előző ábrán bemutatott időviszonyok, valamint a CPU teljesítménye alapján meghatározható a CIP Motion controller legfontosabb jellemzője, mégpedig az, hogy hány tengelyt képes vezérelni 1 ms alatt. A controller teljesítőképességének növelése érdekében

optimalizálhatjuk a keretterhelést. Az Ethernet/IP típusú rendszerek ezt úgy oldják meg, hogy bizonyos adatösszetevőket, amelyek hosszú ideig nem változnak, azokat csak egyszer, általában a kapcsolatfelvétel után továbbítják. Vagyis az adattovábbítási stratégia azon alapszik, hogy csak azokat az adatmezőket továbbítják minden ciklusban, amelyek változnak. Ennek a rugalmas adattovábbításnak köszönhetően a keretterhelés közel negyedére csökkenthető, ami pedig a kiszolgált tengelyvezérlők számának a növekedését eredményezheti.

2. A Profinet

A PROFINET szabványos (IEC 61158 és IEC 61784 [7]), nyílt ipari Ethernet hálózat, amely kielégíti az automatizálási technológiák és ipari folyamatirányító rendszerek minden követelményét. A PROFINET segítségével kivitelezhetők a decentralizált terepi eszközök közötti biztonságos adatkapcsolatoktól a gyors válaszidőt igénylő motorhajtásokig minden irányítási tevékenység. A valós idejű adatátviteli lehetőségei mellett a teljesen IT technológiára épülő előnyei játszanak fontos szerepet abban, hogy egyre inkább kezd elterjedni az ipari kommunikációban. A PROFINET lehetőséget biztosít a már meglévő terepi buszrendszerek: PROFIBUS, INTERBUS, ASI megoldások kiváltására, anélkül, hogy a meglévő eszközöket lecserélnék.

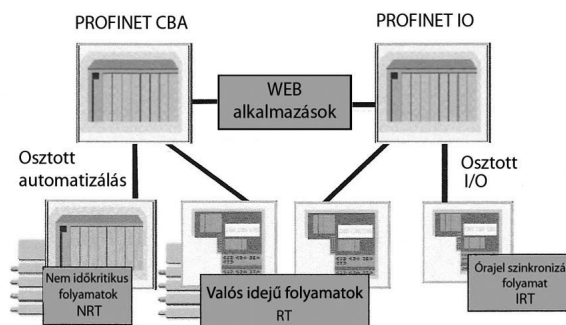
A PROFINET rendszer előnyei elsősorban a gyártásautomatizálás osztott rendszereinek (DCS) valós idejű ipari Ethernetes kommunikációjában nyilvánulnak meg. A rendszer szinte teljes egészében az információs technológia elemeire épül:

- szabványos csatlakozókat (RJ45) és hálózati elemeket (hub, switch, stb.) használ,
- gyakorlatilag az összes hálózati topológiát támogatja,
- biztosítja a hálózaton keresztül a diagnosztizálás lehetőségét,
- biztonságos kommunikációt tesz lehetővé,
- a WLAN segítségével a különleges mozgást végző gépelemek intelligens érzékelői is könnyen kapcsolódhatnak a vezérlőhöz.

A PROFINET használata egyben csökkenti az eszközök telepítési költségeit, kevesebb mérnöki és felügyeleti munkát igényel, de biztosítja a rendszerfejlesztés rugalmasságát és bővíthetőségét.

2.1. A Profinet CBA

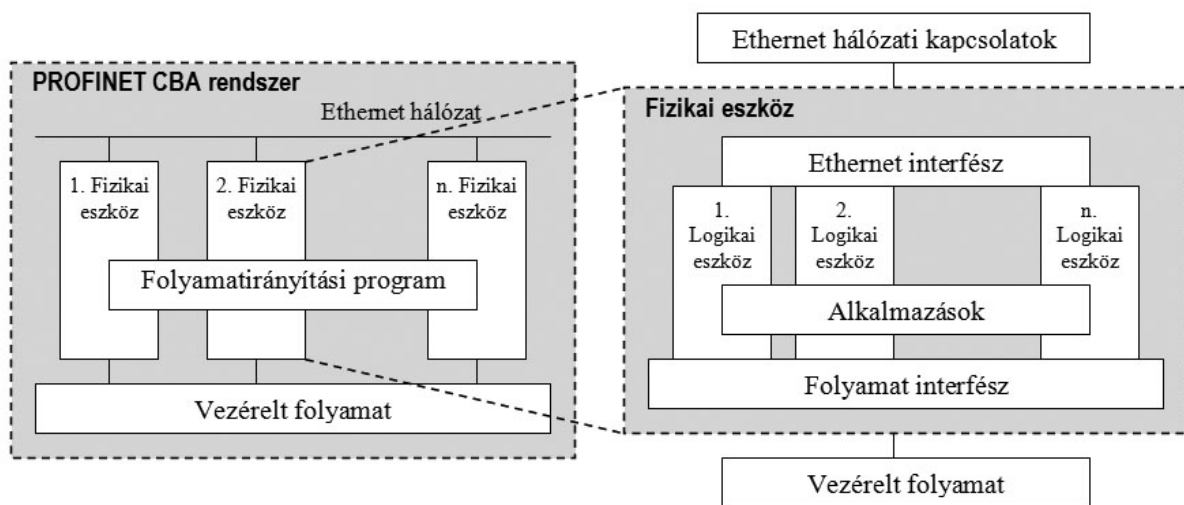
A PROFINET rendszer elemeit, funkcionalitásukat tekintve két külön kategóriába sorolhatjuk attól függően, hogy az automatizálás melyik szintjén teljesítenek szolgálatot, de természetesen közöttük is kiépíthető Ethernet kapcsolat. (11. ábra)



11. ábra. Egy teljes PROFINET rendszer kommunikációs kapcsolatai [2]

A PROFINET CBA-t komponens alapú, elsősorban programozható funkcionalitású intelligens terepi eszközök valamint kontrollerek számára alakították ki. A komponens modell alatt gépek és technológiai modulok egymástól függetlenül működő részegységeit kell érteni. A fix funkcionalitású komponensek mellett számos programozható funkcióval rendelkező komponens is részt vesz. Ez nagymértékben megkönnyíti a gépek, gépsorok valamint gyártóegységek modulrendszerű objektumainak kialakítását, amelyek jelentősen lecsökkentik a mérnöki költségeket. A komponensek leírásához a PCD-t (Profinet Component Descriptor) használja, amely XML-alapú és a gyártó komponens generátorával hozható létre.

A komponensek tekintetében a PROFINET CBA teljes mértékben alkalmazkodik a 2000-ben kibocsátott, az osztott automatizálási rendszerek IEC 61499-1 szabványához. Hierarchikus struktúra és objektumorientáltság jellemzi, melynek komponensei az eszközök, erőforrások, funkcióblokkok és az alkalmazás. A PROFINET CBA eszközei és erőforrásai között Ethernet kapcsolat van. A hierarchia csúcsán a rendszer áll, amely kapcsolatban lehet más rendszerekkel. A 12. ábrán egy ilyen rendszer- illetve eszközmodellt láthatunk [8].



12. ábra. PROFINET CBA IEC 61499-1 szerinti rendszermodell

A fenti modellben a fizikai eszközök a rendszert alkotó hardverobjektumokat jelentik, amelyek Ethernet hálózaton keresztül kapcsolódnak egymáshoz, illetve a hierarchia tetejét képező PROFINET vezérlőhöz. Mindegyik eszköz saját MAC címmel rendelkezik, és a konfiguráció során IP címet is kapnak. A logikai eszközök tulajdonképpen az erőforrásokat jelentik, amelyek a fizikai objektumok eszközvezérlő szoftverei. Az ezekhez beprogramozott funkcióblokkok összessége meghatározza az eszközhöz hozzárendelt alkalmazás objektumot. Ez egy futtatható program, amely saját adatterülettel rendelkezik és biztosítja az eszköz technológiai funkcióját. Ezeknek az alkalmazásoknak az összessége alkotja egy adott PROFINET CBA rendszer folyamatirányítási programját, azaz a futási objektumot (Runtime Automation Object).

2.2 A Profinet IO

Az Ethernetes kommunikáció szempontjából a PROFINET IO a legkritikusabb. Itt lehet ugyanis megvalósítani az órajel szinkronizált, mozgásvezérlőknél alkalmazott, meglehetősen időkritikus PROFINET IRT kommunikációt is. Architektúrája a jól ismert Profibus rendszer filozófiáját követi azzal a meghatározó különbséggel, hogy az eszközök közötti kommunikáció az ipari Ethernet nyújtotta lehetőségeken alapul. Ebből következik, hogy a

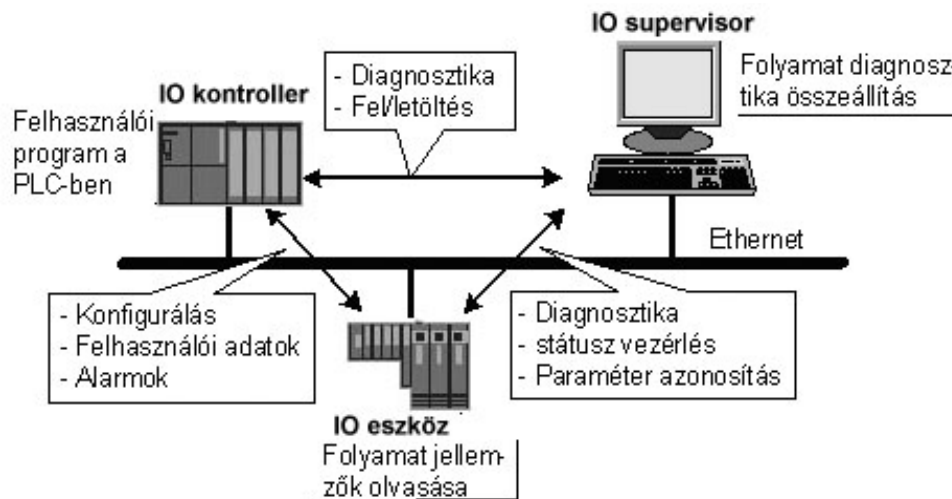
master-slave modellt az termelő-fogyasztó megoldás váltja fel, amely a kommunikációs kapcsolatok szemszögéből egyenrangú felekként tekinti a kapcsolatban résztvevő eszközöket.

A PROFINET IO rendszeren belül a nyílt internetes alkalmazások mellett mindhárom, a fejezet elején már bemutatott kommunikációs osztálybeli kapcsolat megvalósítható. Támogatja az összes lehetséges hálózati topológiát, beleértve a redundancián alapuló kapcsolatokat is. A rendszer felépítését tekintve alapvetően három elemre épül. Ezek a következők:

- PROFINET kontrollert (IO kontrollert): ez egy PLC, amelyen a felhasználói program fut,
- PROFINET IO eszköz (IO eszköz): decentralizált terepi eszközvezérlő, amely kommunikál a IO kontrollerttel,
- PROFINET IO supervisor: programozó vagy diagnosztikai eszköz.

Megjegyzés: Egy adott rendszeren belül általában több IO eszköz van konfigurálva. Maximális számuk a kontrollertől függ. (Tipikusan 1024 vagy 2048.)

A rendszer elemeit és az elemek közötti kapcsolatokat és a fontosabb feladatokat a 13. ábra szemlélteti.



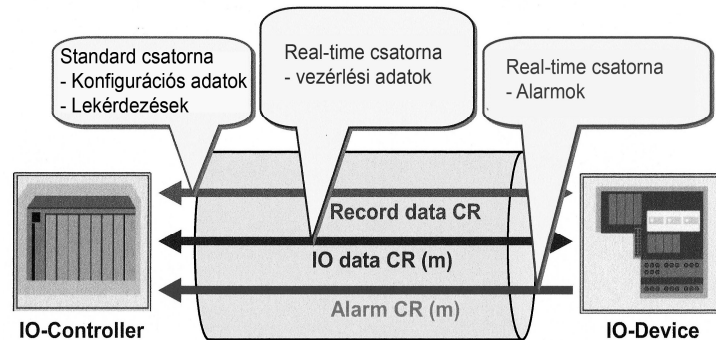
13. ábra. A PROFINET IO elemei és kommunikációs kapcsolatai

Míg a PROFINET CBA a komponens alapú modellt támogatja, a PROFINET IO moduláris rendszerű. A modulok fizikai slotokban vannak elhelyezve, amelyek bármikor bővíthetők újabb modulok hozzáadásával és konfigurálásával. Egy slotnak lehet egy vagy több al-slotja amelyek az IO csatornákat tartalmazzák 1-től n-ig [9]. Ezek a csatornák jól meghatározott egyedi IO címmel azonosíthatók, amelyek a konfiguráció kialakítása után leképzésre kerülnek a kontrollert adatmemóriájában. Az IO eszközökhöz tartozó konfigurációs fájlok XML alapúak. Ezeket a gyártó biztosítja, de a felhasználó is létrehozhatja a GSD-vel (General Station Description). Ebből következik, hogy adott gyártótól származó IO eszköz más ipari Ethernet rendszerbe is implementálható.

2.3 PROFINET kommunikációs stratégia

Az előzőekben már említettem, hogy a Profinet IO rendszer kommunikációs stratégiája az termelő-fogyasztó modellen alapszik. Az IO eszköz adatokat küld a kontrollertnek a folyamat állapotáról. Ilyenkor az IO eszköz az termelő, a kontrollert pedig a fogyasztó. Amikor a kontrollert küldi a folyamat vezérléséhez szükséges kimeneti adatokat, akkor szerepet cserélnek, a kontrollert lesz az termelő, az IO eszköz pedig a fogyasztó.

Ahhoz, hogy a kommunikációs csatorna létrejöjjön, az termelő és a fogyasztó között egy virtuális logikai kapcsolatra is szükség van, amit *alkalmazás relációnak* (Application Relation – AR) nevezünk. Az IO controller mindegyik IO eszközzel felépít egy-egy AR kapcsolatot. Ezek a kapcsolatok automatikusan felépülnek közvetlenül a rendszer indítása után. Amikor egy kapcsolat létrejön az termelő és a fogyasztó között, az alkalmazás reláción belül épülnek ki a különböző *kommunikációs relációk* (Communication relation – CR) (14. ábra, [8]). A szakirodalom szerint, mindezeket a *kontextus menedzsment* (Context management – CM) felügyeli és irányítja.



14. ábra. PROFINET IO kommunikációs relációk [2]

A kapcsolatfelvételt mindig az termelő kezdeményezi és a sikeres vétel nyugtázása után elbontja. Alapvetően háromféle kommunikációs kapcsolat alakulhat, attól függően, hogy milyen jellegű kommunikáció épül ki:

- Record Data CR nem ciklikus adatátvitelkor, pl. konfigurálás, lekérdezés,
- IO Data CR ciklikus bemeneti és kimeneti adatok továbbításakor,
- Alarm Data CR hibaesemények alkalmával (14. ábra).

Az utóbbi kettő a valós idejű, míg az első a hagyományos Ethernet csatornán keresztül jön létre.

A vezérlési adatok továbbítása tehát az IO Data CR ciklikus kapcsolaton keresztül bonyolódik, ezért ennek a működéséről érdemes kicsit bővebben beszélni. Az IO Data CR alapvető funkciója, hogy továbbítsa az IO adatokat a controller és az IO eszközök között a már említett termelő-fogyasztó modell szerint. A kapcsolat létrejötte során az adatokon kívül még az alábbi paraméterek is továbbításra kerülnek:

- egy lista a továbbításra kerülő IO adatobjektumokról,
- a küldési intervallum beállításairól, pl. küldési idő, küldési szorzó.
- az átviteli frekvencia

A létrehozandó kommunikációs kapcsolatok száma a PROFINET IO konfigurációjában van meghatározva attól függően, hogy hány IO eszköz van telepítve. A létrejött kapcsolat kétirányú, vagyis tulajdonképpen két ellentétes IO Data CR jön létre, amelyek állapotinformációt is tartalmaznak. Közvetlen nyugtázás a keretek érkezéséről nincs, viszont ha a fogyasztó nem kapja meg a kapcsolatfelvételkor a továbbítási listán szereplő adatokat három IO busz ciklusidő eltelte után sem, akkor hibaüzenetet generál. Az erre vonatkozó információkat az RT keret ciklusszámláló eleme tartalmazza, amely inkrementálódik, amikor az adott küldési ciklus alatt nem érkezik meg a vezérlési adat.

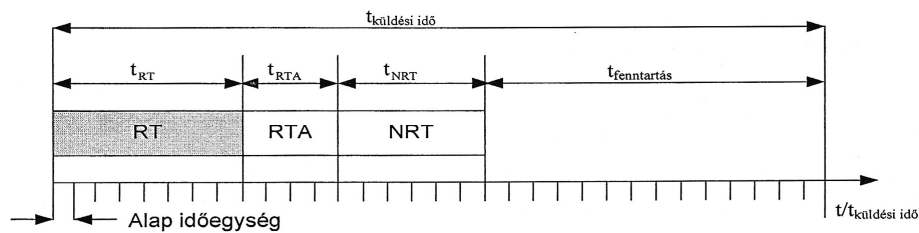
Néhány fontosabb jellemzőt szeretnék kiemelni itt, amelyek feltétlenül szükségesek a kommunikációs stratégiák elemzéséhez. Ezek a következők:

- a *küldési idő* (Send clock time)
- az *osztási tényező* (Reduction ratio)
- a *frissítési ciklusidő* (Update time)
- a *küldési ciklusidő* (Send cycle)

A küldési idő az az intervallum, amely alatt a ciklikus adatok továbbításra kerülnek az IO Data CR-en keresztül. Eszköz specifikusan határozható meg a 31,25 µs-os alapidő egész számú többszöröseként.

$$\text{Küldési idő} = \text{küldési tényező} \cdot 31,25 \mu\text{s}$$

A küldési tényező 1 és 128 közötti egész számú értékeket vehet fel. Nem szinkronizált valós idejű adatok esetén ez a tényező 32 vagy annál nagyobb értékű lehet. Ha 32-vel egyenlő, akkor a küldési idő 1 ms lesz. Ezt az esetet szemlélteti a következő, 15. ábra.



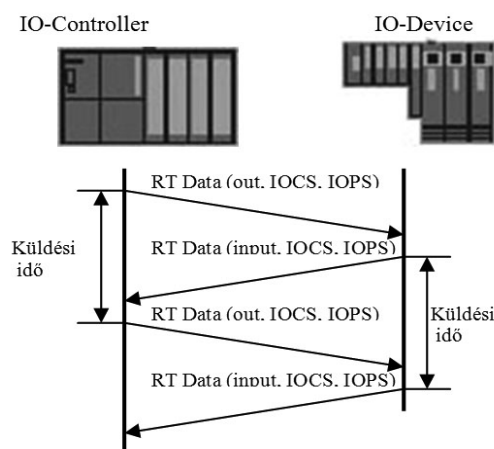
15. ábra. A küldési idő RT adatok esetében [8]

A fenti ábrában RT-vel a ciklikus, RTA-val a nem ciklikus valós idejű, valamint NRT-vel a nem valós idejű adatokat jelölik. Alapesetben a három időtartam együttesen nem haladja meg a teljes küldési idő 60%-át.

Mivel nincs mindig szükség nagyon gyors adatátvitelre, a küldési frekvenciák eltérők lehetnek a különböző IO eszközöknél. Így a leglassúbb állomás nem tartja fel a teljes adatátvitelt. Ezek számára az *osztási tényező* segítségével alacsonyabb küldési frekvenciákat tudunk beállítani. Az osztási tényező mindig 2^n . Következik, hogy egy adott IO eszközvezérlő frissítési ideje:

$$\text{Frissítési idő} = 2^n \cdot t_{\text{küldési idő}}$$

Ebben a ciklusban kapják meg az IO eszközök a kontrollertől az adatokat és ugyancsak ilyenkor továbbítják az IO eszközök is a controller felé adataikat. Ezt a kétirányú ciklikus adatátvitelt szemlélteti a 16. ábra. Mindegyik IO adat mellett jelen van még két-két állapotjelző attribútumot tartalmazó információ is. Az IOPS az termelő, míg az IOCS a fogyasztóra vonatkozó attribútumokat tartalmazza [8].

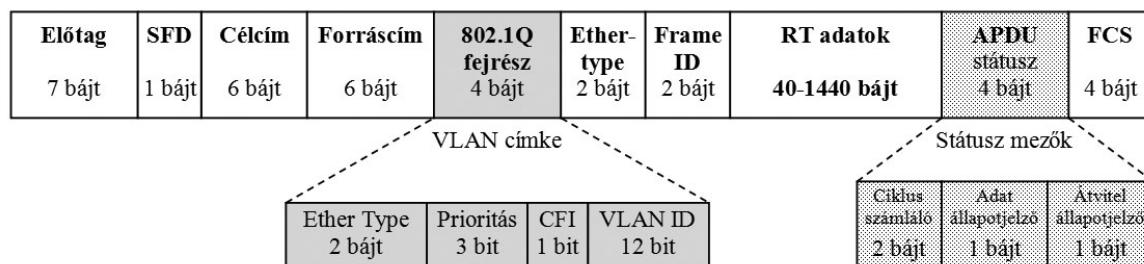


16. ábra. PROFINET IO valós idejű ciklikus adattovábbítási stratégia

A *küldési ciklus* azt az időtartamot jelenti, amely alatt az összes IO eszköz adatot cserél, vagyis az alrendszer technológiai ciklusidejét jelenti. Mivel az alrendszeren belül a kontroller és az egyes IO eszközvezérlők között különböző frissítési idők konfigurálhatóak, ez azt eredményezi, hogy a küldési cikluson belül, a tényleges küldési idő is változik, attól függően, hogy az adott frissítési ciklusban hány darab adatkeretet kell elküldjön.

2.4. A valós idejű protokoll elemei

Az előző fejezetben bemutatott kommunikációs relációknak megfelelően az adatok továbbításához a PROFINET különböző protokollokat használ, amelyek az Ethernet II keretformátumra épülnek. Ahhoz, hogy a valós idejű adatokat is továbbítani tudjuk Etherneten keresztül anélkül, hogy befolyásolnánk a hagyományos TCP/IP alapú IT kommunikációt, nyilvánvaló, hogy az IEEE 802.3 szabványnak megfelelő keretet, ki kell egészítenünk további elemekkel. Ezek egyrészt prioritást biztosítanak az RT kereteknek, másrészt segítségével azonosítani lehet az RT adattípusokat is, de az átvitel állapotára vonatkozóan is tartalmaznak információkat. A PROFINET erre a célra az IEEE 802.1Q szabványban foglalt VLAN címkés keretformátumot alkalmazza. Ez az előbbi formátumot kiegészíti még 2x2 bájtnyi információval (VLAN címke), a keret végére pedig az APDU (Application Protocol Data Unit) státuszinformációkat hordozó bájtok kerülnek (17. ábra).



17. ábra. Valós idejű adatokat szállító keretformátum

A VLAN címke négy fontos azonosítót tartalmaz:

- *EtherType* (2 bájt):
 - 0x8100 a keret VLAN protokoll azonosítója;
 - ha < 0x0600 a keret IEEE 802.3 szabvány szerinti;
 - ha 0x600 Ethernet II típusú keret (FrameID azonosítóval).
- *Prioritás* (3 bit): 0-7 közötti szintek, ahol az RT keretek 6, 7 prioritási szintet kapnak.
- *CFI* (1 Bit, formátum indikátor):
 - CFI = 0, Ethernet;
 - CFI = 1 Token ring.
- *VLAN-ID* (12 bit) prioritásazonosító:
 - 0x000 – a keret prioritással ellátott;
 - 0x001 – nincs prioritás;
 - 0x002-0xFFE – szabadon használható

További keretazonosítók:

- *EtherType* a protokollazonosító. A nem valós idejű adatokat tartalmazó keretek 0x0800, a valós idejű adatokat tartalmazó keretek pedig a 0x8892 PROFINET protokollazonosítót kapják.
- A *FrameID* információi a keret adatainak tulajdonságaira utal. (ciklikus, nem ciklikus, vagy Alarm típusú adatok) Például: 0xFC01 magas szintű hibajel típusú UDP/IP, vagy 0x8000-0xBEFF valós idejű (RT) adatot tartalmaz.

Megjegyzés: A 4 bájtos *VLAN tag* és a 2 bájtos *EtherType* számára a 6 bájtot az adatkitöltő bájtok (PAD) közül használunk fel azért, hogy az IEEE802.1Q-t nem támogató kapcsolók is átengedjék az ilyen típusú kereteket. Ezért az adatmező tartomány 40 és 1440 bájtközött tölthető fel. A keret teljes hossza pedig az előtaggal együtt 1522 bájtközött tölthető fel.

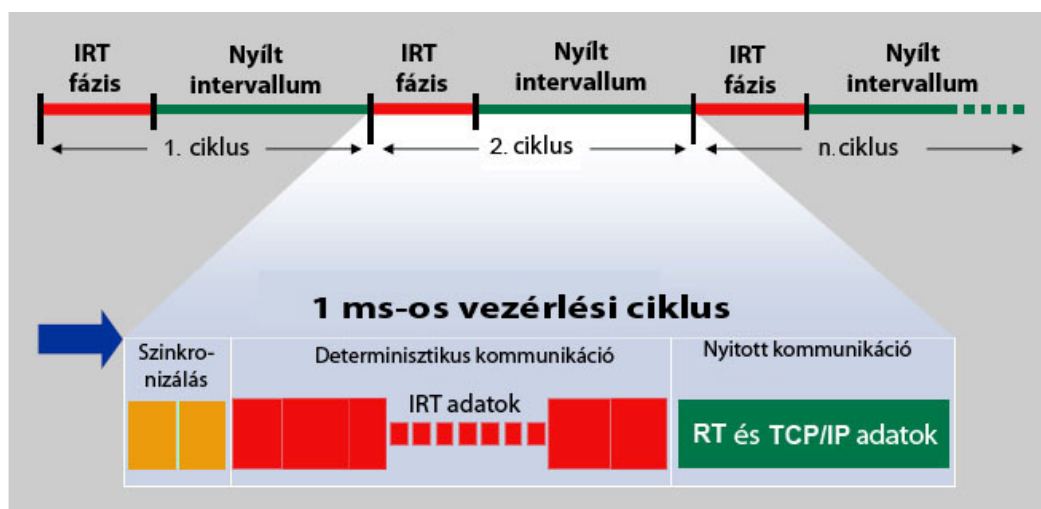
A státuszinformációkat tartalmazó bájtok közül kiemelném az adat-állapotjelző (Data Status) 5. bitjét, amely ha 1-es, akkor az átvitel hibamentes, vagy a Ciklusszámlálót (Cycle Counter), amely inkrementálódik, ha az termelő megismételte a küldést. Minden egyes inkrementálás időben 31,25 µs időt jelent. Ha a számláló túlsordul (meghaladja a megengedett ismétlések számát), a fogyasztó a keretet eldobja, az alkalmazás pedig hibaüzenettel reagál.

A nem ciklikus valós idejű adatok (Alarm Data CR) hasonló keretformátumot használnak, azzal a különbséggel, hogy mivel itt nincs szükség állapotjelzésre, ezért az APDU státusz bájtok hiányoznak.

2.5. A Profinet IRT kommunikáció elve

A motorvezérléseknél, ahol időben akár 2-3 vagy annál több tengely összehangolt, pontos mozgására van szükség, a szinkronizálás nélküli, valós idejű üzemmód nem alkalmazható. Az ilyen, időszinkronizált feladatok Ethernet hálózati kommunikációja a PROFINET IRT (Isochronous Real-Time) segítségével valósítható meg, amely röviden az alábbiakkal jellemezhető:

- A kommunikáció kizárólag egy hálózati szegmensen belül történik, mert a kapcsolat csak a fizikai és az adatkapcsolati OSI rétegeket használja [10].
- A busz ciklus időben két részre osztható (18. ábra): IRT fázisra (un. piros intervallum), és egy azt követő nyílt intervallumra (un. zöld intervallum).
- 1 ms-nál kisebb ciklusidők és 1 µs-os, vagy annál kisebb eltéréssel (*Jitter*),
- A küldési intervallumok szinkronizálása.



18. ábra. Az IRT kommunikáció időosztása [2]

A terepi eszközök kommunikációs időintervallumai rugalmasan állíthatók a felhasználó által. A piros intervallum (IRT fázis) nagysága főleg attól függ, hogy hány IRT eszköz csatlakozik a controllerhez. A controller konfigurálása során, amikor megadjuk az IRT eszközök számát, ez az intervallum automatikusan lefoglalásra kerül az IRT keretek számára. Az IRT keretekkel azonos időben érkező nem időkritikus kereteket az IRT kapcsoló (ASIC – Application Specific Integrated Circuit) pufferelem a zöld intervallumig, vagyis mindig szabad

utat biztosít az IRT adatok számára. A zöld intervallumban a valós idejű (RT), prioritással ellátott, IEEE 802.3Q szabvány szerinti, valamint a nem valós idejű (NRT) keretek továbbítása valósul meg.

A zöld intervallumból (nyílt intervallum) a pirosba (IRT fázis) való átmenetet, a szinkronizálás fázisa alatt (narancssárga intervallum) a determinisztikus kommunikáció alapjául szolgáló hardver (ERTEC 400 vagy ERTEC 200) felügyeli. A küldési ciklus mindig a szinkronizálással kezdődik, amikor a kontroller órájához igazodnak az IRT eszközök órái.

2.6. A Profinet IRT protokoll

Az előző fejezetben bemutatott időosztásos kommunikációs stratégia, valamint a hardvertámogatás (ASIC) miatt a protokoll prioritási információkat nem tartalmaz. Ezért a VLAN címke használata felesleges. Az Ether Type itt is 0x8892, a 2 bájtnyi keretazonosító (Frame ID) értéke (0x0100 – 0x7FFF) pedig ciklikus IRT kerettípusra utal. További 8 bájtt az eszközökre (IO Controller, IO Device) vonatkozó attribútumokat (IOPS, IOCS) és APDU státusz biteket tartalmazza. C osztályú kommunikációról lévén szó, a protokoll közvetlenül az adatkapcsolati rétegben építi fel a keretet, illetve választja le az IRT adatokat. Nincs beágyazott UDP/IP fejléc sem ezért a keret 36 – 1490 bájtt között terhelhető. (19. ábra)

Előtag	SFD	Cél cím	Forrás cím	Ether Type	Frame ID	IRT adatok	FCS
7 bájtt	1 bájtt	6 bájtt	6 bájtt	2 bájtt	2 bájtt	36-1490 bájtt	4 bájtt

19. ábra. PROFINET IRT keretformátum

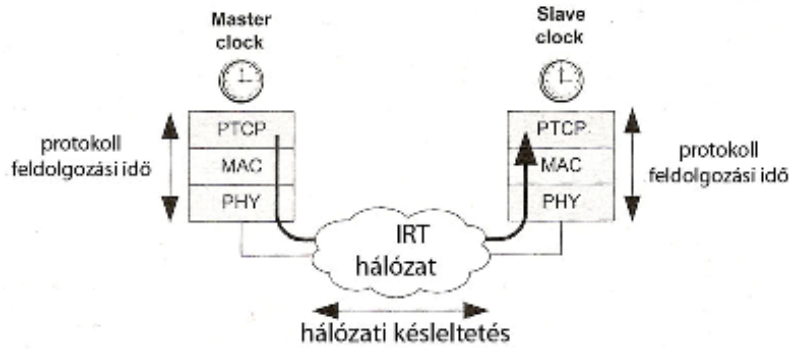
A motorvezérléseknél a vezérlési adatok nagysága csak ritkán haladja meg a 64 bájttot. A legtöbb esetben ez az adatmennyiség még a 36 bájttot sem éri el, ami azt jelenti, hogy csak a minimális 64 bájttos keret kerül továbbításra. Következik, hogy a keretterhelési tényező meglehetősen alacsony, a legtöbb esetben, alig több mint 56%. A keretterhelési tényező (F_{PF}) meghatározására a következő összefüggést használtam:

$$F_{PF} = \frac{d_{DATA}}{d_{FRAME}} \cdot 100\% \quad (2)$$

A fenti relációban d_{DATA} a vezérlési adatok nagyságát, d_{FRAME} pedig a teljes keretméretet jelöli bájttban kifejezve. Maximális keretterhelésnél ez az érték elérheti a 98%-ot.

2.7. Szinkronizálás PTCP protokollal

Szinkronizálás hiányában bármennyire is igyekszünk csökkenteni a ciklikus adatküldési időt, a mozgásvezérlők számára nem tudunk kellően precíz, összehangolt vezérlési adatokat továbbítani. Az IRT hálózat szinkronizálására a PROFINET a PTCP-t (Precision Transparent Clock Protokoll) használja, amely hasonló az IEEE 1588 szabványban leírt PTP (Precision Time Protocol) protokollhoz [4].



20. ábra. A PTCP beépülése az adatkapcsolati rétegbe [8]

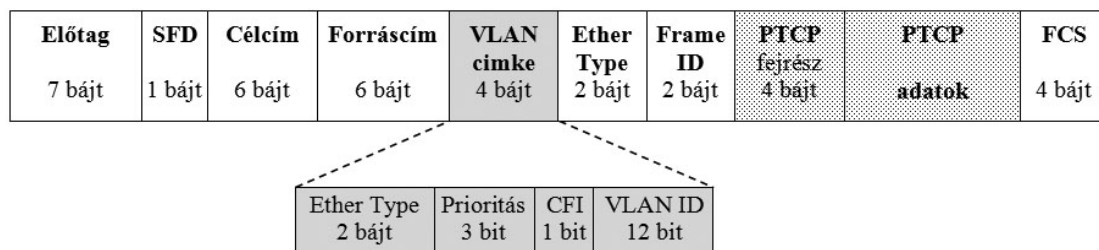
A szinkronizálás megvalósítása az ASIC egyik alapvető feladata. A PTCP az OSI réteg második szintjébe integrálódik, ez egyben azt is jelenti, hogy nem routolható (20. ábra). Ebből következik, hogy az IRT keretek csak azonos alhálózati szegmensen belül továbbíthatók.

A PTCP a következő fontosabb tulajdonságokkal rendelkezik:

- tízed mikroszekundum nagyságrendű szinkronizálási pontosság (jitter),
- alacsony erőforrás használat (CPU, RAM),
- minimális hálózati sáv szélességet igényel,
- alacsony adminisztrációs követelmények [8].

A PTCP segítségével akár 32 IRT eszköz szinkronizálása is megvalósítható egyetlen egy mester óra (Master Clock) alapján. A módszer lényege, hogy a csomópontokban un. átlátszó órát (transparent clock) alkalmaznak, amely a nem neki címzett szinkronkeretet átérteszti, csupán egy bejegyzést szúr be a keretbe, amely a csomópont késleltetéséről tartalmaz információt. Így a saját csomópontbeli órajel szabályozási kör szinkronizálási hibája nem kerül továbbításra. Ebből következik, hogy a jitter maximális értéke lineárisan növekszik, és csak az IRT eszközök számától függ [6].

A mester óra funkcióját betöltő eszköz rendszerint az IO controller. A szinkronizáláshoz szükséges adatokat a szinkronkeretekben továbbítja az IO eszközök számára. (21. ábra)



21. ábra. PTCP keretformátuma

A kereten belül fontos szerepet tölt be a keretazonosító (Frame ID). Ennek értéke mutatja, hogy éppen milyen típusú szinkronkeret kerül továbbításra: Sync, Follow Up, DelayReq vagy DelayRes. A PTCP fejrész számon tartja, hogy hányadik szinkronkeret került továbbításra, valamint tartalmazza a szinkronizálás pontosságára vonatkozó információt. (10 ns, 1 ns)

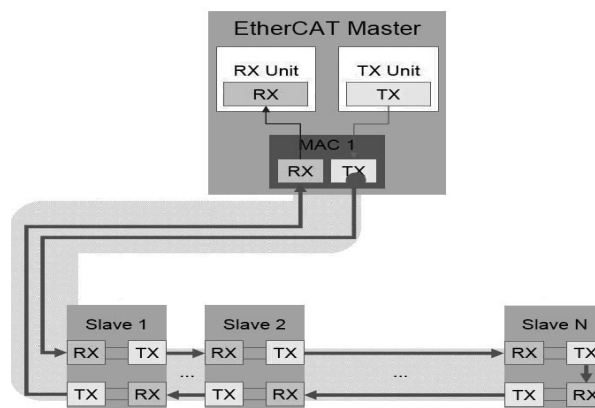
2.3. EtherCAT

A szigorúan valód idejű ipari Ethernet kommunikációs rendszereknél (C osztály) a vezérlési adatok továbbítása két módon valósítható meg. Az egyik megoldás, amikor a kontroller (master) mindegyik, vele Ethernet kommunikációs kapcsolatban lévő IO eszköznek (slave) külön-külön címzi a kereteket, amelyek a megfelelő vezérlési adatokat tartalmazzák. Ezt a módszert alkalmazza az előző fejezetben bemutatott PROFINET IRT rendszer. Láthattuk, hogy ebben az esetben 36 bájtnál kisebb vezérlési adatok gyakorlatilag nem befolyásolják a továbbításhoz szükséges időt, mert így a teljes keretméret nem haladja meg a minimális, 64 bájtot. Hátránya ennek a megoldásnak, hogy az előbb említett esetben a keretterhelési tényező a legjobb esetben is csak 56% lehet. Igazi előnye az 1000 Mb/s-os Gigabit Ethernet hálózaton mutatkozik meg, amikor az adattovábbításhoz szükséges idő gyakorlatilag már nem függ a keret méretétől.

A másik megoldás, amikor arra törekszünk, hogy kihasználjuk a maximális keretterhelést, amely az IEEE 802.3 szerint 1500 bájtnál lehet. Ebben az esetben az IO eszközök számára továbbítandó vezérlési adatokat (telegramokat) igyekszünk egy vagy több kereten belül úgy elhelyezni, hogy minél nagyobb legyen a keretek kihasználtsága. Ez a módszer kétségtelenül előnyt jelent kisméretű (36 bájtnál kisebb) telegramok továbbítása esetében. Így egy Ethernet kereten belül egyszerre továbbíthatja akár az összes csomópontban lévő eszközök vezérlési adatait. Ezt a megoldást alkalmazza a Beckhoff cég által fejlesztett EtherCAT (Ethernet for Control Automation Technology) [11].

3.1. Az EtherCAT rendszer fontosabb jellemzői

Az EtherCAT-et igen rövid ciklusidő, zavarmentes kommunikáció és nagy pontosságú szinkronizálás jellemzi. 100 Mb/s-os full duplex Ethernet hálózaton szegmensenként gyakorlatilag szinte korlátlan számú (max. 65 535) IO eszköz kapcsolható össze egymással. A gyártó adatai szerint 100 IO eszköz, összesen 1000 digitális IO csatorna esetében a frissítési idő mindösszesen 30 μ s, de 100 darab szervo tengely, egyenként 8 bájtos IO adatokkal történő hajtásakor sem sokkal haladja meg a 100 μ s-ot. E rendkívüli gyorsaságot annak köszönheti, hogy a csomópontokban lévő IO eszközök „menet közben”, közvetlen memória hozzáféréssel (DMA) olvassák le a nekik címzett, illetve írják fel a továbbítani kívánt adatokat (22. ábra). Az erre a célra kialakított Ethernet hardvernek köszönhetően a csomópontokon belüli késleltetési idő rendkívül rövid, mindösszesen 1,35 μ s full duplex hálózatban [12].



22. ábra. EtherCAT adattovábbítás

Az EtherCAT kontroller a feladat elvégzéséhez nem szükséges, hogy rendelkezzen semmilyen extra kommunikációs processzorral. Ebből következik, hogy bármilyen kontroller, amelyik rendelkezik szabványos Ethernet interfésszel alkalmas erre a feladatra, függetlenül attól, hogy milyen operációs rendszert vagy alkalmazást használ.

A rendszer a vezérlési adatok nagysága szempontjából sem mondható korlátozottnak. Noha a motorvezérléseknél nincs szükség nagyméretű telegramok továbbítására, a gyártó 60 kilobájtban határozta meg a maximális adatmennyiséget, amely továbbítható a csomópontoknak. (Ebben az esetben természetesen több keretre van szükség) [13].

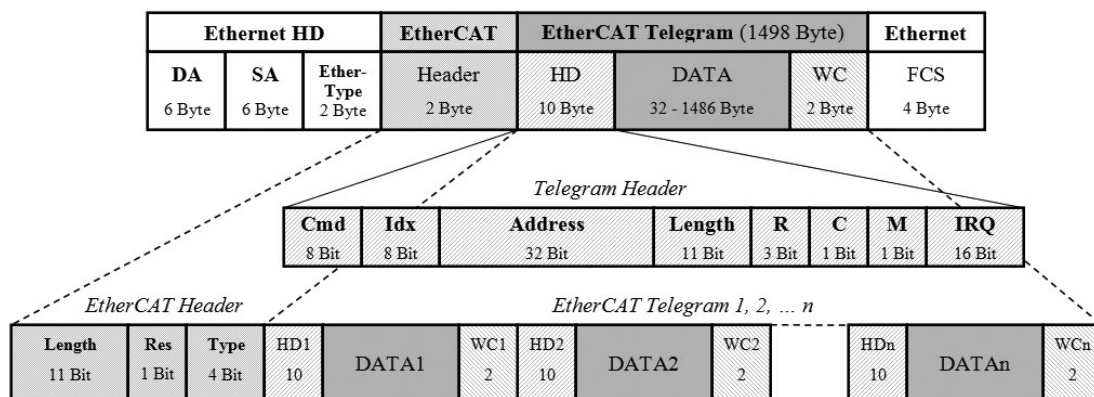
A rendszer elsősorban a busz topológiát támogatja, de jól használható, csillag, fa és vegyes topológiájú hálózatokban is. Vegyes topológiában a ciklusidő azért lényegesen nagyobb, mint például a busz topológiában.

3.2. Az EtherCAT protokollok

Egy teljes EtherCAT rendszer alapvetően kétféle kommunikációs protokollt használ, amelyek a szabványos IEEE 802.3 Ethernetre épülnek:

- nem szinkronizált folyamatokhoz, mint például folyamatautomatizálás vagy konfigurációs beállításokhoz, folyamatvizualizáláshoz valamint lekérdezésekhez a TCP/IP vagy UDP/IP alapú EtherCAT Automation Protokollt (EAP), vagy más néven mailbox protokollt,
- szinkronizált, ciklikus vezérlési adatok, mint például motorvezérlések vagy szabályzók adatainak továbbítására az EtherCAT eszköz-protokollt (EtherCAT Device Protocol).

Ez utóbbi esetben a keret azonosítása EtherType-al történik, amelynek értéke 0x88A4. A master így tudja megkülönböztetni a vezérlési adatokat tartalmazó kereteket más típusú Ethernet keretektől. Mindegyik keret tartalmaz még egy kétbájtos EtherCAT fejléccet is, amely főleg a beágyazott adatmennyiség hosszára (Length) és típusára (Type) vonatkozóan tartalmaz információt. (23. ábra). Az eszköz-protokoll esetében a Type mező értéke 1-el egyenlő. A motorvezérléseknél használt, erősen hardver-beállítottágú (FPGA) slave eszközök csak az ilyen típusú kereteket képesek kezelni.



23. ábra. EtherCAT eszköz-protokoll keret struktúrája

Azért, hogy egy kereten belül, több csomóponti eszköz számára is tudjuk továbbítani a vezérlési adatokat (Type = 1), azonosítás céljából mindegyik telegramot további fejléccel (Telegram Header, HD), illetve működésszámlálóval (Working Counter, WC) kell ellátnunk. Az erre a célra felhasználható maximális keretterhelés 1498 bájt lehet. Ebben az esetben a (2) szerinti keretterhelési tényező elérheti akár a 97%-ot is. Ha az összes telegram nem fér el egy

keretben, a master további keretekben helyezi el ezeket. Ha túlságosan kisméretű a telegram és kevés a slave eszköz, akkor töltelékbitekkel (Pad) egészíti ki egészen 64 bájtig [12].

3.3. Címzési módok

A slave eszközök címzése, illetve a telegramok célba juttatása, valamint az, hogy mit kell csinálni a kapott adatokkal, az EtherCAT kommunikációs stratégiájának a legfontosabb feladata. Ennek lebonyolításában nyújt segítséget a telegram fejléc, amelynek mezőit és azok funkcióit a 1. táblázat foglalja össze.

1. táblázat. Az EtherCAT telegram fejlécének mezői és funkciói

Mező	Adat típus	Érték/Magyarázat
Cmd	Bájt (1)	EtherCAT utasítás típus
Idx	Bájt (1)	A master által használt index a dupla vagy elveszett telegramok azonosítására.
Address	Dupla szó (4 Bájt)	Címzési mód meghatározása (Automatikus címinkrementálás, csomóponti címzés, logikai címzés)
Length	11 Bit	A telegramhoz tartozó adat hossza
R	3 Bit	Nem használt, értéke nulla
C	1 Bit	Keret körforgás: engedélyezés C = 1; tiltás C = 0
M	1 Bit	További telegram: utolsó telegram M = 0; van következő telegram M = 1
IRQ	Szó (2 Bájt)	Eseményregiszter. A slavek írják és logikai VAGY kapcsolaton keresztül a master elemézi.

A címzési stratégia során különbséget kell tennünk keret- illetve telegramcímzés között. Busz topológiában a master a keretet mindig a hozzá közvetlenül kapcsolódó legelső slave eszköznek címzi, vagyis az Ethernet célcím (DA) mezőbe ennek az eszköznek a MAC címe kerül. A forráscím (SA) értelemszerűen a master MAC címe lesz. A kereten belül a telegramok kétféle címzési mód szerint juthatnak célba: eszközcímzéssel illetve logikai címzéssel. Eszközcímzésnél lehetőség van háromféle címzésre: pozíció- illetve csomópontcímzés és broadcast.

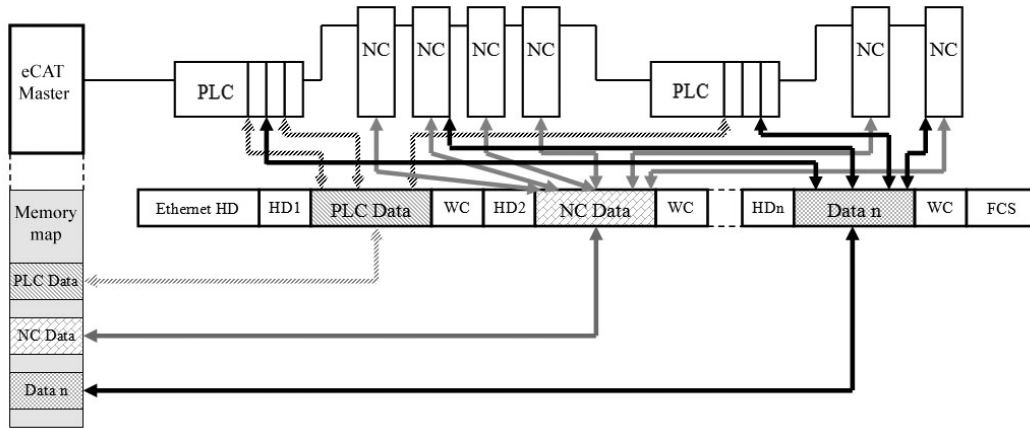
A legegyszerűbb és a leggyakrabban alkalmazott mód a pozíciócímzés. A telegramok, kivéve az elsőt, az eszközök negatív címértékét tartalmazzák. Mindegyik slave, kivéve az utolsót, miután befejezte a saját adatcseréjét, inkrementálja a soron következő telegramok címmezőjének pozíció bájtjait (2 bájt pozíció, 2 bájt offset). Az az eszköz lesz megcímezve, amelyik a nullás címet olvassa. Hátránya ennek a módszernek, hogy az eszközök csomópontbeli sorrendje nem cserélhető fel.

Csomópontcímzéskor a master a konfiguráció során beolvasott címek alapján végzi a telegramok címzését. Ezek a címek a slavek EPROM-jaiban vannak tárolva és nem módosíthatóak. Az eszköz azt a telegramot olvassa/írja, amelyiknek a címe megegyezik az EPROM-jába beírt értékkel. Leginkább vegyes topológia esetén használják.

A broadcast címzés során mindegyik csomóponti eszköz kiválasztásra kerül. Ezt a címzési módot inicializáláskor és állapotlekérdezőskor használja a vezérlő.

Logikai címzéskor a vezérlő memóriatérképének a konfiguráció során meghatározott részei kerülnek a telegramokba. A slave eszközök a memóriakezelő egységeik (FMMU – Fieldbus Memory Management Unit) segítségével a saját memóriájukba képezik ezeket az

adatokat. Indításkor, ugyancsak a konfiguráció során meghatározott beállításokkal, a vezérlő tudatja mindegyik eszköz FMMU-val, hogy pontosan melyik memóriaterületet olvashatják, illetve írhatják. A bitcímzési mód is megengedett ebben az esetben, így a digitális IO eszközök adatai is viszonylag könnyen továbbíthatóak. A módszert főleg a folyamatirányítási adatok célba juttatásakor használják, jelentősen csökkentve ezáltal a telegram fejlécek számát, tovább javítva a keretterhelési tényezőt (24. ábra) [14].

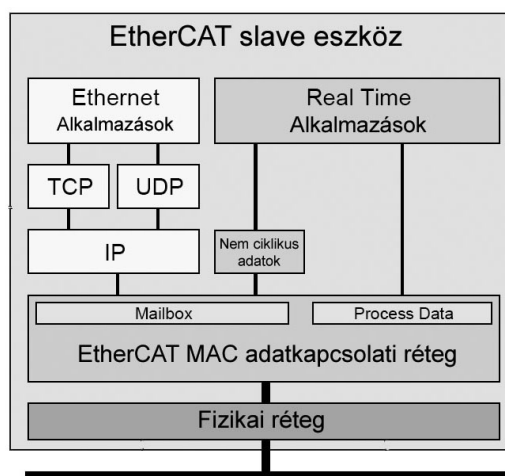


24. ábra. A logikai címzés

3.4. Ethernet az EtherCAT felett

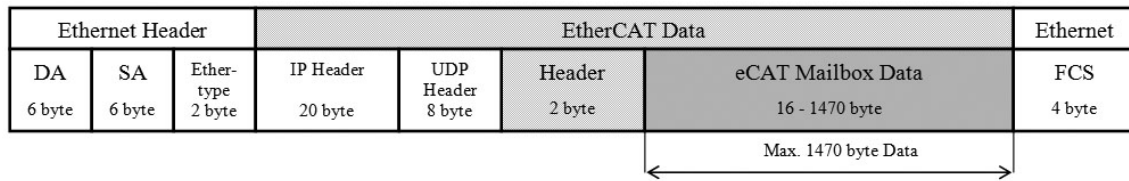
Az EtherCAT slave interfészek teljesen kompatibilisek a szabványos Ethernetnel, vagyis beágyazódnak az EtherCAT MAC alrétegébe. Ez az alréteg meg tudja különböztetni a ciklikus adatokat, a nem ciklikus „mailbox” típusú adatoktól. Ezen kívül a szabványos Ethernet alkalmazások számára is teljesen átjárhatóak a TCP/IP vagy UDP/IP protokollok, így az eszközök, legfőképpen a master zavartalanul kommunikálhat nem EtherCAT rendszerbeli eszközökkel is. Ez továbbá azt is jelenti, hogy a topológia bármely szabad Ethernet portjára bármilyen informatikai eszköz csatlakoztatható anélkül, hogy zavart okozna az EtherCAT rendszerben.

A következő ábrán (25. ábra) a mailbox protokoll beágyazását látjuk az adatkapcsolati rétegbe, egész pontosan az EtherCAT MAC alrétegbe.



25. ábra. A mailbox protokoll beágyazása a MAC alrétegbe

A folyamatirányítási adatok továbbításakor (Type = 4) vagy a nem ciklikus un. „mailbox” típusú adattovábbításhoz (Type = 5) az EAP protokoll az UDP/IP kommunikációt használja. Ilyenkor a fejrész is beágyazódik az EtherCAT adattartományba, így maximálisan már csak 1470 bájtnyi adat továbbítható (26. ábra).

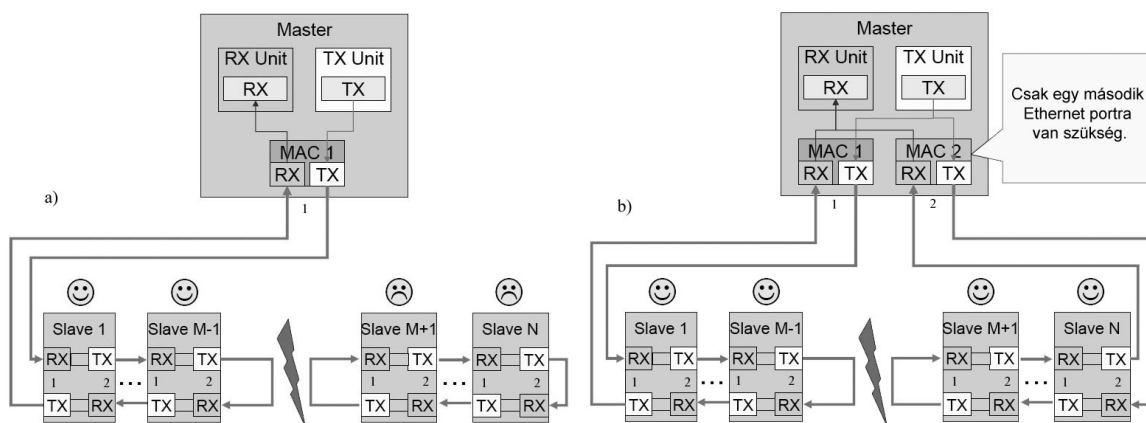


26. ábra. Mailbox típusú keret struktúra

Az ilyen típusú EtherCAT kereteknél az EtherType azonosító értéke 0x800 vagyis azonos a szokásos Ethernet keretek azonosítójával és a 0x88A4 UDP portot használja.

3.5. EtherCAT redundancia

A lineáris topológiát alkalmazó ipari Ethernet rendszerek egyik gyakran előforduló problémája a kábelszakadás. A slave eszközök felismerik a kábelszakadást. Ilyenkor azok a portok, amelyek érzékeli a szakadást automatikusan lehurkolják, Rx – Tx vonalukat. A vezérlő és a szakadási pont közötti szakasz látszólag továbbra is működőképes marad, viszont a vezérlő érzékeli, hogy a slave-ek egy része nem cserél adatot, mert a nekik címzett telegramok számlálója (WC) nem inkrementálódott. Egy másik probléma, hogy a szakadás utáni szakaszon pedig elkezd „körbeforgani” a legutolsó keret, mert a szakadást érzékelő port itt is lehurkolta vonalait. (27a. ábra) Ez a helyzet akár veszélyes is lehet ezért, hogy ezt elkerüljük, a szakadást érzékelő eszköz a fejlécben a C bitet (lásd 1. táblázat) nullára állítja, ezzel megakadályozza a körbeforgást, eldobja a telegramot és hibakóddal reagál.



27. ábra. Kábel redundancia megvalósítása [11]

A probléma megoldása nagyon egyszerű. A vezérlőben csak egy második Ethernet port konfigurálására van szükség és az utolsó slave 2. portját összekötni a vezérlő 2. portjával. (27b. ábra) Az így módosított topológia szerint a vezérlő elsődleges portjáról küldött kereteket a másodlagos port fogja fogadni és fordítva. Hibátlan rendszernél, a két keret tartalma azonos kell legyen. Ezt a vezérlő ellenőrzi. Azért, hogy ne duplázódjanak a keretek, a master az elsődleges port által fogadott kereteket eldobja. Ha szakadás történik, az ezt érzékelő portok azonnal lehurkolják vonalaikat, a kommunikáció nem szakad meg a vezérlővel egyik eszköz

felől sem. Viszont a master, miután összehasonlítja a két fogadott keretet látja, hogy azok különbözőek, nem dobja el egyiket sem, hanem a kettőből összerakja a helyes memóriatérképet.

4. További ipari Ethernet rendszerekről röviden

Az előző fejezetekben három alapvető, egymástól különböző elveken alapuló, igen elterjedt valós idejű ipari Ethernet megoldást mutattam be. Ezek a megvalósítások mind különböző gyártókhoz kapcsolhatók. Ez nem véletlen, ugyanis a gyártók, amikor a lehetőségek megnyíltak az Ethernet irányába igyekeztek egymástól független megoldásokat implementálni. A továbbiakban nagyon röviden jellemeznék még néhány jelentősebb gyártóhoz kapcsolódó ipari Ethernet megoldást.

Az osztrák B & R Automation cég által kifejlesztett Ethernet PowerLink (EPL) kifejezetten gyors és pontos, C osztályú átviteli sebességet biztosít akár 200 μ s-os ciklusidővel és 1 μ s-os jitterrel. Gyorsaságát annak köszönheti, hogy az adattovábbítást akárcsak a Profinet IRT két fázisban hajtja végre. A szinkron fázisban mindegyik állomás rögzített szélességű időrést kap a valós idejű adatok továbbítására. A telegramok azonosítását a keretbe ágyazott EPL azonosító biztosítja. Az aszinkron fázisban kerül sor a hagyományos IP alapú információ továbbítására. A vezérlési adatok küldésekor egy időben mindig csak egy állomás férhet hozzá a hálózathoz, így ütközés nem fordulhat elő [15, 16].

A német SERCOS International cég az EtherCAT-hez hasonló elven működő megoldást dolgozott ki, amelyet SERCOS III néven forgalmaznak. A szigorúan valós idejű követelményeket is képes teljesíteni. Busz vagy gyűrű topológiában ciklusideje 31,25 μ s-tól 65ms-ig konfigurálható. 100 Mb/s-es Full duplex Ethernet hálózatot feltételez csavart érpáron vagy optikai szálon [17, 18].

A MODBUS/TCP alkalmazása főleg a Schneider Electric gyártmányokban kerül felhasználásra. A MODBUS az egyik legismertebb ipari protokoll, amelyet még a MODICON cég fejlesztett ki. Ezt fejlesztették tovább ipari Ethernetre, úgy hogy tulajdonképpen a MODBUS keretbe ágyazzák be a TCP keretbe. Ezáltal A osztályú valós idejű kommunikációt valósít meg kérés/válasz mechanizmussal, amely jól illeszkedik a master-slave technikához. Ezzel akár több ezer eszközt is képes „megszólítani” [19].

Végül, de semmiképp sem utolsó sorban a Mitsubishi Electric CC-Link elnevezésű megoldását említeném, amely 1 Gb/s-os Ethernet hálózaton, gyűrű topológiában a „token passing” stratégiát alkalmazza. Optikai kábelon keresztül akár 66 km átviteli távolságú hálózat kialakítására is alkalmas. A valós idejű adatok számára ciklikus kommunikációt biztosít. A nem valós idejű adatok továbbítása a tranziens fázis alatt történik. Egy controller maximum 119 állomást tud kiszolgálni. Ugyan nagy sebességű hálózatot használ, de ennek ellenére meglehetősen lassú, 16 csomópontnál 256 bájttal a ciklusidő kb. 2 ms [20].

IRODALOMJEGYZÉK

- [1] Mark Chaffee, CIP Motion Implementation Considerations, CIP Networks Conference, február 2009.
- [2] PROFINET System Description – Technology and Application, Version June 2011.
- [3] Introduction to Modbus TCP by Acromag Inc. Wixom, USA, 2005. www.acromag.com/sites/default/files/Acromag_Intro_ModbusTCP_765A.pdf
- [4] Dirk Mohl, IEEE 1588 - Precise Time Synchronization as the Basis for Real Time Applications in Automation, 2004.
- [5] Rockwell Automation Publication ENET-RM002A-EN-P, 2011. július.

- [6] Ferenczi István: *PTC protokoll alkalmazása a Profinet IRT kapcsolatok szinkronizálására*, IV. Nyíregyházi Doktorandusz (Phd/DLA) Konferencia Nyíregyháza, Magyarország, 2010.12.07. ISBN 978-963-318-174-4.
- [7] Max Felser, Thilo Sauter: Standardization of Industrial Ethernet – the next battlefield? 2004. IEEE International Workshop on Factory Communication Systems, 2004. Proceedings.
- [8] Raimond Pigan, Mark Metter: Automating with Profinet (2006)
- [9] Ferenczi István: *PROFINET - több mint ipari Ethernet*, microCAD 2009, XXIII. International Scientific Conference, Miskolc, Magyarország, 2009.03.19-2009.03.20. Automation and Telecommunication Section, pp. 21-26. ISBN 978-963-661-874-2
- [10] PROFINET – Technology and Application, 2006. április.
- [11] EtherCAT Master – Application Developers Manual, Doc. Nr. P.4500.21/Rev. 1.4, 2011.
- [12] Ferenczi István: *Beágyazott telegramokat tartalmazó ipari Ethernet keretek ciklusideje*, GÉP, Vol. 63/5, pp. 51-54. ISSN 0016-8572, 2012.
- [13] EtherCAT – The Ethernet Fieldbus, 2009, www.ethercat.org
- [14] EtherCAT Slave Controller Technology, Ver. 1.7 by Backhoff Automation GmbH, 2010.
- [15] Ethernet Powerlink, Perfection in Automation, 2004. www.br-automation.com
- [16] Ethernet Powerlink Basics, 2011, www.ethernet-powerlink.org
- [17] SERCOS III, by SERCOS International e.V. 2011. www.sercos.org
- [18] Peter Lutz, SERCOS III – Universal Real-Time Communication for Automation, MOF Summit, Tokyo, 2011. november.
- [19] Introduction to Modbus TCP by Acromag Inc. Wixom, USA, 2005. www.acromag.com/sites/default/files/Acromag_Intro_ModbusTCP_765A.pdf
- [20] Mitsubishi Electric FA, CC Link IE, 07.2008.