İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİYEL VERİ İLETİŞİMİNDE TCP/IP UYGULAMASI

YÜKSEK LİSANS TEZİ

Müh. Timur BİR

(504021037)

Tezin Enstitüye Verildiği Tarih : 22 KASIM 2005

Tez Danışmanı : **Doç.Dr. Salman KURTULAN**

Diğer Jüri Üyeleri **Yrd.Doç.Dr. Osman Kaan Erol**

Yrd.Doç.Dr. Turan Söylemez

KASIM 2005

ÖNSÖZ

Ethernet haberleşmesi son yıllarda bilişim sektörüyle paralel olarak Endüstriyel Otomasyon sektörüne de girmiş bulunmaktadır. Bu çalışmada Ethernet, endüstriyel otomasyon yönünden genel olarak incelenmekte ve Ethernet'in açık ve esnek yapısından yararlanarak Siemens S7 PLC'ler ile PC'ler arasında haberleşme olanakları örnek yazılımlar ile birlikte sunulmaktadır. Ethernet'in her geçen gün Endüstriyel Otomasyon sektöründe daha fazla kullanılmaya başlanacağı ise açıkça gözükmektedir.

Bana bu konuda çalışma fırsatı sağlayan ve çalışmalarım sırasında benden desteğini esirgemeyen danışmanım Doç. Dr. Salman KURTULAN'a teşekkürü borç bilirim.

Timur BİR

İstanbul, Kasım, 2005

İÇİNDEKİLER

ÖNSÖZ **ii**
İÇİNDEKİLER **iii**
KISALTMALAR **v**
TABLO LİSTESİ **vii**
ŞEKİL LİSTESİ **viii**
ÖZET **ix**
SUMMARY **x**

1.GİRİŞ **1**

2.TCP/IP GENEL YAPI, MODELLER VE PROTOKOLLER **3**
2.1. TCP – IP Hakkında Genel Bilgiler 3
2.2 Model,Protokol ve Gerçekleştirimler 10
2.3 OSI Referans Modeli 11
2.4 Taşıma Protokolleri 16
2.4.1 Aktarım Kontrol Protokolü [Transmission Control Protocol (TCP)] 16
2.4.2 Kullanıcı Datagram Protokolü [User Datagram Protocol (UDP)] 22
2.4.3 TCP ile ISO (ISO over TCP RFC1006) Protokolü 23

3.SIEMENS S7 ETHERNET HABERLEŞMESİ **25**
3.1 S7 Ethernet CPler ile Simatic S7 haberleşme olanakları 25
3.2 Çeşitli Cihazlar Arasında Mümkün Olan Haberleşme Tipleri 26
3.3 SIMATIC S7 PLC'ler ile S5 Uyumlu Haberleşme Bağlantıları 28
3.3.1 S5 Uyumlu Haberleşmede Gönder/Al (Send/Receive) ara yüzü 28
3.3.2 Gönder/Al programlama fonksiyonları 29
3.3.3 S5 Uyumlu Haberleşmede Oku/Yaz (Fetch/Write) Servisleri 31
3.4 CP443-1'de Haberleşme Konfigürasyonlarının Yapılması 32
3.4.1 Servislerin ve Arayüzlerin Konfigüre Edilmesi 32
3.4.2 TCP-IP Bağlantılarının Konfigüre Edilmesi 34

4. PROGRAMLAMA **38**
4.1 Soket Programlama 38
4.1.1 Soket programlama Nedir 38
4.1.2. Temel Prensipler 38
4.2. Visual Basic'te Winsock.dll ile Soket Programlama 39
4.2.1 Winsock.dll Özellikler (Properties) 39
4.2.2 Winsock.dll Metotlar (Methods) 39
4.2.3 Winsock.dll Olaylar (Events) 40
4.3 S5 Header Protokolü 40
4.3.1 Yaz Çerçevesi (Write frame) yapısı 41
4.3.2 Oku Çerçevesi (Fetch frame) yapısı

4.3.3 Parametre değerleri 44
4.4 Program örneği 45
4.4.1 Veri yapısı ilk değerler 45
4.4.2 Okuma istemi 47
4.4.3 Okuma veri gelişi 49
4.4.4 Yazma İstemi 49
4.4.5 Yazma Cevabı 50
4.4.6 Kullanıcı Arayüzü 51

5. SONUÇ **53**

KAYNAKLAR **54**

EKLER **55**

ÖZGEÇMİŞ **56**

KISALTMALAR

TCP **:**Transmission Control Protocol

IP **:**Internet Protocol

UDP **:**User Datagram Protocol

PDU **:**Protocol Data Unit

OSI **:**Open System Interconnection

ISO **:**International Standards Organization

FTP **:**File Transfer Protocol

ACK **:**Acknowledge Character

RFC **:**Request for Comment

DoD **:**Department of Defence

SNMP **:**Simple Network Management

SMTP **:**Simple Mail Transfer Protocol

PG **:**Programming Device

OP **:**Operator Panel

HMI **:**Human Machine Interface

IT **:**Information Technology

PLC **:**Programmable Logic Controller

CP **:**Communication Processor

HW **:**Hardware

CPU **:**Central Processing Unit

TABLO LİSTESİ

Sayfa No

Tablo 2-1. IP adres sınıfları 4
Tablo 2-2. Özel ağlar için IP adresleri 6
Tablo 2-3. Yedi tabakalı model 11
Tablo 3-1. Cihazlar arasında haberleşme tipleri 27
Tablo 3-2. Gönder/Al ara yüzü bir iş içinde gönderilebilecek veri miktarları 28
Tablo 4-1. Sunucu İstemci Yapısı 39
Tablo 4-2. Yazma istem çerçevesi 41
Tablo 4-3. Yazma onay çerçevesi 42
Tablo 4-4. Okuma istem çerçevesi: 43
Tablo 4-5. Oku onay çerçevesi 43
Tablo 4-6. Parametre değerleri 1.Kısım 44
Tablo 4-7. Parametre değerleri 2.Kısım 45

ŞEKİL LİSTESİ

Sayfa No

Şekil 2-1: Basit bir ağ yapısı 3
Şekil 2-2: Ağ topolojileri 7
Şekil 2-3: Ethernet kablo bağlantı şekilleri 10
Şekil 2-4: TCP paket yapısı 18
Şekil 2-5: UDP paket yapısı 22
Şekil 3-1: Siemens Ethernet haberleşme tipleri 25
Şekil 3-2: Gönder/Al ara yüzü 29
Şekil 3-3: Donanım yapılandırma 33
Şekil 3-4: IP Adresi ayarları 34
Şekil 3-5: Bağlantı oluşturulması 35
Şekil 3-6: Port ve IP adreslerinin girilmesi 36
Şekil 3-7: Bağlantı seçenekleri 37
Şekil 4-1: Program Kullanıcı Arayüzü 51

ÖZET

Endüstriyel veri iletişiminde TCP/IP uygulaması adlı bu tezde, Siemens S7 PLC'ler ile Ethernet üzerinden haberleşme olanakları incelenmiş, bu haberleşmenin Endüstriyel Otomasyonda yaygın olarak kullanılan seri haberleşme protokollerine göre avantajları üzerinde durulmuştur. Endüstriyel otomasyon alanında daha yaygın kullanım için Ethernet'in yapısından kaynaklanan birtakım dezavantajları ortadan kaldırmak için protokoller üzerinde yapılan çalışmalar halen devam etmektedir, bu alana yönelik protokoller de protokoller kısmında tanıtılmaktadır. İkinci aşamada Ethernet üzerinden haberleşmeyi sağlayan soket programla incelenmiştir, bu bölümde Windows işletim sisteminin soket programlama ara yüzü winsock.dll ve kullanımı açıklanmaktadır. Son olarak Ethernet'in açık protokollerinden yararlanılarak visual basic programlama dili ile bir program oluşturulmuştur. Bu programın amacı TCP/IP üzerinden herhangi özel bir yazılım veya donanım kullanmadan standart programlama gereçleri ile bir PLC'nin hafıza ve giriş/çıkış alanlarına nasıl ulaşılacağını göstermek ve bir anlamda düşük maliyetli bir İnsan Makine Ara yüzü yazılımının temellerini oluşturmaktır. Bu örnek programın alt yapısı kullanılarak daha geniş çaplı programlar oluşturmak mümkündür, örnek program haberleşeme protokolünün genel yapısını ve çalışma mekanizmasını gösterir niteliktedir.

SUMMARY

In this thesis called a TCP/IP application in industrial communications, Ethernet communication capabilities of Siemens S7 PLC's are studied and the advantage against the serial communication, which is commonly used in Industrial Automation, is discussed. For a more common use of the Ethernet in industrial automation sector recent researches are made to avoid the disadvantages of the Ethernet protocols. These protocols for industrial automation are also described in the related chapter called Protocols. In the following section, socket programming, which serves programming programs supporting Ethernet communication, is introduced. In this section winsock.dll, the Ethernet programming API for Windows operating system is explained. At last, a software is developed using the open protocols of Ethernet. This software's goal is to communicate with the PLC over TCP/IP without any additional software and hardware support, and to show how to read and write the memory and the I/O areas of the PLC. This software can also be the basis of a low cost HMI (Human Machine Interface) system. It is possible to use this example programs' structure to form a more detailed industrial program. The example shows the protocol details and its working mechanism.

1.GİRİŞ

Giriş ve Çalışmanın Amacı

Günümüzde bilişim sektöründeki ilerlemelere paralel olarak Ethernet Haberleşmesi kullanımı Endüstriyel Otomasyon alanında da sıkça kullanılmaya başlanmıştır. Internet'in yaygın kullanımı, cep telefonlarından Internet'e erişim ve benzeri teknolojik uygulamalar yavaş yavaş otomasyon sistemlerini de etkilemektedir. Sistemlerin uzaktan izlenmesi (web üzerinden v.b.), hata durumlarının uzaktaki birimlere iletilmesi, üretim takibinin başka merkezlerden de yapılabilmesi gibi uygulamalar sıkça karşımıza çıkmaktadır. Bu noktada Ethernet bu izleme ve yönetim sistemleri ile otomasyon süreçleri arasındaki geçişi doldurmaktadır.

Bu noktada Ethernet'in avantajı hızı ve IT teknolojilerinin temel protokolü olmasıdır. PLC kontrollü otomasyon sistemimizi Ethernet ağına dâhil ettiğimizde izlenebilirlikte sınırları ortadan kaldırmış olmaktayız. Ethernet'in bir diğer avantajı ise protokollerin açık, basit ve net olmasıdır, dolayısı ile farklı üreticilerin farklı özellikteki cihazları aynı platform üzerinde toplanabilmektedir. Ethernet üzerinden bir sisteme hiçbir hazır yazılıma para ödemeden, standart programlama gereçleri ile ulaşmak mümkündür.

Bu tezde öncelikle Ethernet'in genel yapısı ve protokolleri incelenmiş daha sonra Siemens S7 serisi PLC'lerin Ethernet haberleşme kabiliyetlerine göz atılmıştır. Son bölümde ise Visual Basic programlama dilinde Ethernet programlama komutları incelendikten sonra TCP üzerinde S5 header protokolü kullanılarak bir PLC'nin hafıza ve G/Ç alanlarını okuyabilen ve bu alanlara yazabilen bir program geliştirilmiştir. Programın tek gereksinimi Ethernet kartı bulunduran standart bir PC'dir herhangi bir ayrı donanım ya da ücretli yazılım gerektirmez. Dolayısıyla bu yazılım geliştirilerek çok düşük maliyetli bir insan makine ara yüzü (HMI) oluşturmak mümkündür. Bu noktada Ethernet, sıklıkla kullanılan ve protokollerin

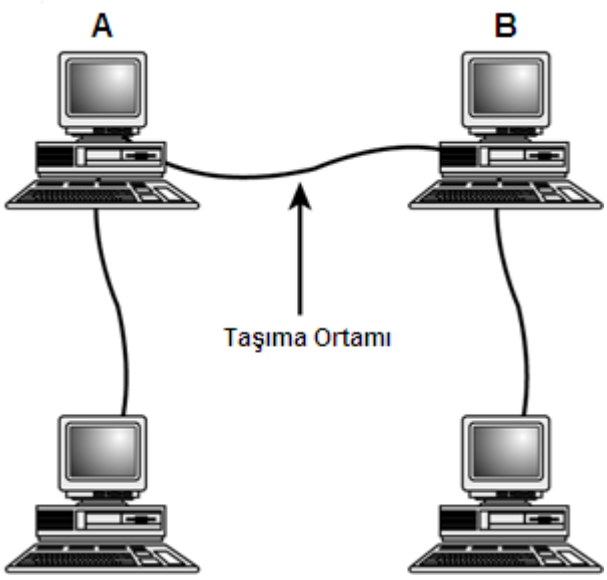
genellikle ticari şirketlerin mülkiyetinde olduğu (Modbus protokolü haricinde) seri haberleşmeye de önemli bir alternatif sunmaktadır. Ethernet alt yapısını kurmak için yapılan ekstra maliyetler bu şekilde karşılanmış olmaktadır.

Tezin uygulama kısmında programla dili olarak basit ve anlaşılabilir olmasından dolayı Visual Basic seçilmiştir. Daha kapsamlı ve büyük uygulamalar için daha gelişmiş programlama dillerinin seçilmesi uygulamanın sağlıklı çalışması için yaralı olacaktır.

2.TCP/IP GENEL YAPI, MODELLER VE PROTOKOLLER

2.1. TCP – IP Hakkında Genel Bilgiler

TCP-IP (Transmission Control Protocol - Internet Protocol), Internet'te, bilgisayarlar arası iletişimi sağlayan protokol grubudur. En basit haliyle bir ağ yapısı Şekil 2-1'de gösterilmiştir. Bilgisayarlar arasında bilgi alışverişi IP paketleri aracılığıyla yapılır. Bilgisayar A bilgisayarı, B ye göndermek istediği veriyi paketler halinde gönderir. Bu paketler içerisinde A'nın ve B'nin adresleri vardır. A'nın gönderdiği paketlere IP-Paketleri, A'nın adresine IP-Adresi denir. Gönderilen paketler, B'nin adresini taşıdığı için bilgisayar ağları arasında kullanılan yönlendiriciler (Router) aracılıyla doğru bilgisayara iletilir [1].



Şekil 2-1: Basit bir ağ yapısı

IP paketleri birbirinden bağımsız olarak gönderilir. B'nin görevi tüm IP paketlerini alıp, A'nın gönderdiği veriyi kazanmaktır. Gönderilen bazı IP paketleri B'ye ulaşmayabilir. Her paketin içinde, gönderilecek paket sayısı hakkında bilgi olduğu

için, B, kaç tane IP paketi alması gerektiğini bilir. Eksik olan paketler A'dan tekrar istenir. TCP'nin görevi, gönderilen IP paketlerini denetlemektir. TCP eksik olan paketleri tespit eder ve A tarafından tekrar gönderilmesi için gerekli işlemleri yapar.

Alt ağ (Subnet) tiplerini ve alt ağ içinde yer alacak bilgisayar miktarını tespit etmek amacıyla IP-numaraları A, B, C, D ve E sınıflarına ayrılır. Sınıflar arasında ayrım ilk 8 bit (bayt) üzerinde yapılır. A, B, C, D sınıflarının hangi adres alanlarına sahip olduklarını Tablo 2-1 de verilmiştir.

Tablo 2-1: IP adres sınıfları

Sınıf	Sistem	1.Bayt	2.Bayt	3.Bayt	4.Bayt
A	Ondalık İkilik	0-127 0xxx xxxx	0-255 xxxx xxxx	0-255 xxxx xxxx	0-255 xxxx xxxx
B	Ondalık İkilik	128-191 10xx xxxx	0-255 xxxx xxxx	0-255 xxxx xxxx	0-255 xxxx xxxx
C	Ondalık İkilik	192-223 110x xxxx	0-255 xxxx xxxx	0-255 xxxx xxxx	0-255 xxxx xxxx
D	Ondalık İkilik	224-255 1110 xxxx	0-255 xxxx xxxx	0-255 Xxxx xxxx	0-255 xxxx xxxx

A Sınıfı Adresler:

IP adresindeki ilk sekizli (octet) 0 ile 127 arasındadır ve varsayılan alt ağ maskesi (subnet mask) ise 255.0.0.0 'dır. Alt ağ maskesi IP adresinin hangi bitlerinin ağ adresini, hangi bitlerinin uç adreslerini gösterdiğini çözümlemeye yarar, bu işlem alt ağ maskesi ile IP adresi "ve" işleminden geçirilerek yapılır. A sınıfı IP adreslerinde ilk sekizli ağ ID' yi diğer üç sekizli ise katılımcı ID' yi gösterir. Burada ilk sekizlinin 0 ve 127 olma durumları özel durumlardır ve ağ'da kullanılmazlar. Örneğin 127.0.0.1 yerel geri döngü (loopback) adresidir. Dolayısıyla A sınıfı IP adresi kullanılabilecek ağ sayısı 126'dır. A sınıfı IP adresine sahip bir ağda tanımlanabilecek host sayısı ise şu formülle hesaplanır; $2^{24} - 2$. Bu işlemin sonucu olarak ta 16.777.214 adet katılımcı olabilir. Peki, burada kullandığımız 24 nereden geldi? A sınıfı adreste katılımcıyı tanımlamak için son üç sekizli (octet) kullanılıyordu. Yani toplam 24 bit'i katılımcı tanımlamak için kullanabiliyoruz. Bu bitler ya 0 ya da 1 olmak zorunda. Bu yüzden birbirinden farklı kaç kombinasyon olacağını 2^{24} ile bulabiliriz. Bu sayıdan 2 çıkarmamızın nedeni ise bu 24 bit'in

hepsinin 0 veya 1 olmasının özel bir anlamı olduğu ve herhangi bir katılımcıya IP adresi olarak verilemediği içindir. Örnek bir A sınıfı IP adresi 49.19.22.156 olarak verilebilir. Burada 49 bu IP adresinin ait olduğu ağın ID' sini 19.22.56 ise bu IP adresine sahip katılımcının katılımcı ID' sini gösterir.

B Sınıfı Adresler:

IP adresindeki ilk sekizli 128 ile 191 arasındadır ve kullanılan alt ağ maskesi (subnet mask) ise 255.255.0.0 'dır. Bu da demektir ki bu tür bir IP adresinde ilk iki sekizli Ağ ID' sini, diğer iki sekizli ise katılımcı ID' yi gösterir. B sınıfı IP adresinin kullanılabileceği ağ sayısı 16.384 ve her bir ağda kullanılabilecek katılımcı sayısı ise 65.534'dür. Örnek bir B sınıfı IP adresi 160.75.10.110.olarak verilebilir.

C Sınıfı Adresler:

IP adresindeki ilk sekizlinin değeri 192 ile 223 arasında olabilir ve varsayılan alt ağ maskesi değeri ise 255.255.255.0 'dır. Yani bu tür bir IP adresinde ilk üç sekizli Ağ ID' yi son sekizli ise katılımcı ID' yi belirtir. Örneğin 192.168.10.101 IP adresini inceleyelim. Bu IP adresi C sınıfı bir IP adresidir. Bunu ilk sekizlin değerine bakarak anladık. Bu IP adresinin ait olduğu ağın CD'si ise 192.168.10'dur. Bu IP adresine sahip cihazın katılımcı numarası ise 101'dir. C sınıfı IP adreslerinin kullanılabileceği ağ sayısı 2.097.152 ve bu ağların her birinde tanımlanabilecek katılımcı sayısı ise 254'dür. Bu ağı 128lik iki alt ağa ayırmak ister isek kullanmamız gereken alt ağ maskesi 255.255.255.128 olur, bu durumda 0 ve 255 adreslerini katılımcılara atayamayacağımız için kaybedilen IP adresi 2*2=4 adettir.

Bu üç IP sınıfının haricinde D ve E sınıfı IP adresleri de mevcuttur. D sınıfı IP adresleri çoğa gönderim (multicast) yayınlar için kullanılır. E sınıfı adresler ise bilimsel çalışmalar için saklı tutulmuştur.

Özel bir IP ağı oluşturmak için RFC1597/1918 tablo 2-2'de gösterilmiş olan adres alanlarını ayırmıştır.

Tablo 2-2: Özel ağlar için IP adresleri

Sınıf	Başlangıç IP	Bitiş IP	Standart alt ağ maskesi
A	10.0.0.0	10.255.255.255	255.0.0.0
B	172.16.0.0	172.16.255.255	255.255.0.0
C	192.168.0.0	192.168.255.255	255.255.255.0

Tabloda Konuk kimlikleri (Host-ID) altı çizili gösterilmiştir.

Bu adres aralıkları çakışma yaşamaksızın birçok kurum tarafından kullanılabilir, bu adresler internet için atanmamıştır veya internete yönlendirilmemiştir.

Konuk kimliği (Host-ID) olarak 0 veya maksimum seçilemez, maksimum adres o ağ için tüme gönderim (Broadcast) adresidir [1].

Topoloji:

Topoloji, iletişim sistemini oluşturan bileşenlerin ara bağlaşımını, sistemin işlevini ve coğrafi konum açısından iletişim sisteminin şeklini belirler. Coğrafi konuma göre iletişim sistemleri yerele alan ağları (LAN) ve geniş alan ağları (WAN) olarak ikiye ayrılır. Yerel alan ağlarında yoğun olarak ortak yol, yıldız veya halka topolojileri kullanılırken, geniş alan ağlarında ağaç ve örgü topolojisi kullanılır [2].

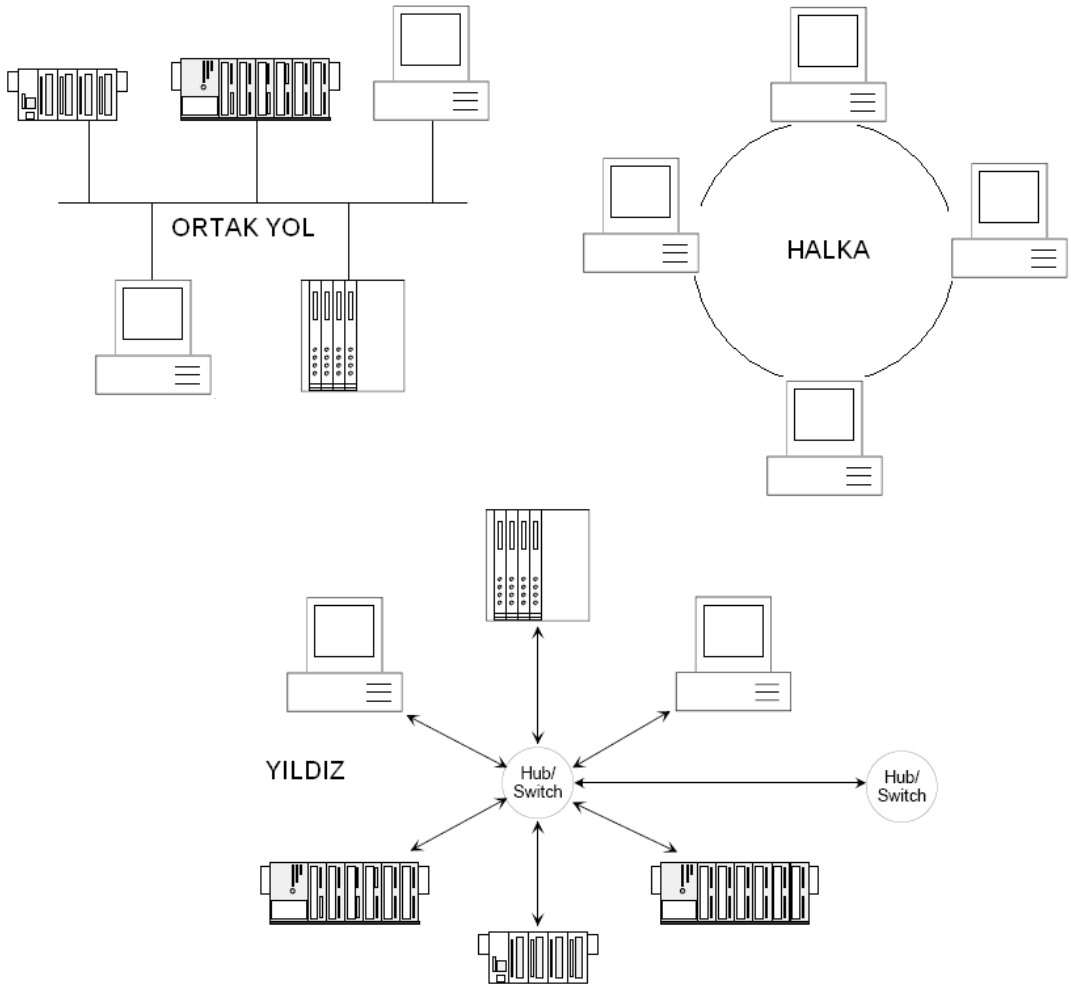
Ortak Yol, Halka ve Yıldız Topolojileri

Bu topolojiler LAN uygulamalarında yoğun olarak kullanılır. Şekil 2-2'de bu topolojilerin şekli görülmektedir.

Ortak yol topolojisinde ağdaki tüm düğümler aynı bir ortak iletim ortamı üzerinden haberleşirler. Veri ve denetim işaretleri aynı anda tüm düğümlere birden gönderilir. Her düğümün bir adresi vardır. Düğümler yol üzerindeki her mesajı okurlar ve yalnızca kendilerini adresleyeni işlerler. Ortak yola yeni düğüm ekleme kolay olur; ancak her yeni eklenen, düğüm başına düşen yol kapasitesini azaltır.

Halka topolojisinde her düğüm komşu iki düğüme bağlıdır; iletim yolu halka biçimindedir. Yolun ele geçirilmesi için en yaygın kullanılan yöntem jetonlu halka

üzerinde dolaştırmaktır. Yolu kimin kullanacağını, yolda dolaşan bir jeton belirler, jetonu ele geçiren düğüm, onu yoldan alıp yola verisini koyar. Göndereceği verisi bittikten sonra jetonu yeniden yola koyar. Aynı düğüm tekrar veri göndermek isterse jetonu ele geçirene kadar beklemelidir. Halka topolojisinin olumlu yanı yoğun iletişim anında bile başarımını fazla düşürmemesidir. Ancak ağa yeni kullanıcı eklenmesi biraz daha zahmetli olur. Halka üzerinden veri alış verişi uygulamaya göre tek veya iki yönlü olabilir.



Şekil 2-2: Ağ topolojileri

Yıldız topolojide düğülerin tamamı merkezi noktadan geçer. Günümüzde ağ uygulamalarında yoğun olarak yıldız topoloji kullanılmaktadır. Ortak yol ve halka uygulamaları bile, mantıksal olarak ortak yol ve halka biçiminde olsa dahi fiziksel bağlantıları yıldız şeklinde olabilmektedir. Yıldız topolojide de ağa

yeni düğüm kolayca eklenebilir. Ancak her düğüme ayrı kablo çekilmesi gerekliliği maliyeti arttıran bir unsurdur. Merkezdeki düğüm bozulduğunda tüm iletişim kopar.

Standart Belirleyen Kurumlar

Standartlaşma her alanda en önemli konudur; üstelik bilgisayar iletişimi söz konusu olduğunda, çok daha fazla önem kazanır. Çünkü farklı üreticiler tarafından geliştirilmiş olan sistemlerin karşılıklı anlaşması söz konusudur. Karşılıklı anlaşmanın sağlanması ve cihazlar arası bağlantıların sorunsuz gerçekleştirilmesi için ITU (CCITT), IEEE, ISO gibi birtakım enstitüler, organizasyonlar kurulmuş olup yeni gelişmeleri tüm üreticilerin uyacağı bir platforma çekmek üzere çalışmalar yapılmaktadır. Marka bağımsız tüm ürünlerin birbirleriyle karşılıklı çalışabilmelerini sağlamak ancak standartlar ile mümkün olabilmektedir. Bu kısımda sayısal iletişim konusunda adından çok sık söz edilen bazı kurumlar tanıtılmıştır.

ANSI- American National Standarts Institute

Birçok endüstri dalında standart belirlemek üzere oluşturulmuş bir enstitü. Hemen her dal için standart belirleyen komitelerden oluşturulmuştur. Bu kurumun yaygın olarak kullanılan standartlarından biri X3T9.5'dir

IEEE- Institute of electrical and Electronics Engineers

ANSI ve ISO'ya bağlı standartları yayınlayan, basan profesyonel bir enstitü; Elektrik ve Elektronik Mühendisleri Enstitüsü. IEEE 802.x LAN standartları bu enstitünün en iyi bilinen standartlarından biridir.

ITU- International Telecomunications Union

Veri haberleşmesi konusunda standart belirlemek üzere CCIT'nin (Consulative Committe for ınternational Telegraphy and Telephony) yeniden yapılandırılması ile oluşturuldu. Çok bilinen standartları V ve X serisi tanımlamalarıdır (V.24, V.42bis, X25 gibi standartlar).

IETF- Internet Engineering Task Force

Internet uygulamasında standart belirleyen, öneren bir komite; kırktan fazla çalışma grubundan oluşmuştur.

RFC- Request for Comment

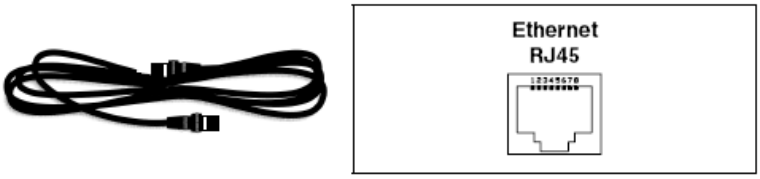
Internet'te kullanılan protokollerle ilgili standartları tanımlayan dokümanlar dizisi. Bütün Internet standartları RFC dokümanları olarak tanımlanır. Her dokümanın bir RFC numarası vardır. Örneğin, RFC1661 PPP'yi; RFC874, Telnet protokolünü; RFC1618, ISDN hizmeti üzerinden PPP'yi tanımlar. Buna göre bir yönlendirici RFC1618'i destekliyor ise bununla ISDN bağlantısı yapılabileceği ve PPP kullanılabileceği ortaya çıkar.

ISO- International Standarts Organization

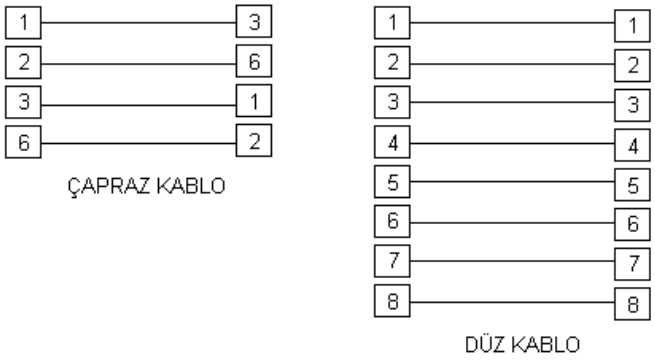
Genel olarak sayısal sistemlerin haberleşmesi ve bilgi alış verişi üzerine örnek standart yapıları belirleyen uluslararası organizasyon. ISO'nun Açık Sistemler Ara bağlaşım (Open System Interconnection, OSI) başvuru modeli tüm ağ mimarileri için örnek bir tanımlamadır [2].

Haberleşme Ara yüz Kabloları

Ethernet haberleşmesi için sadece iki düğümün bulunduğu ağlarda örneğin bir PLC bir PC çapraz Ethernet kablosu kullanılabilir, bu bağlantı bir göbek (Hub) gerektirmez. Üç ve daha fazla düğümün bulunduğu ağlar bir veya birden fazla göbek (Hub) üzerinde düz kablo aracılığıyla bağlanmalıdır. Şekil 2-3 Düz ve çapraz kablo bağlantı şekillerini göstermektedir [2].



Pin	Sinyal	Notlar
1	TX+	--
2	TX-	--
3	RX+	--
4	--	Pin 5 ile kapatıldığında 75 ohm sonlandırma
5	--	Pin 4 ile kapatıldığında 75 ohm sonlandırma
6	RX-	--
7	--	Pin 8 ile kapatıldığında 75 ohm sonlandırma
8	--	Pin 7 ile kapatıldığında 75 ohm sonlandırma



Şekil 2-3: Ethernet kablo bağlantı şekilleri

2.2 Model, Protokol ve Gerçekleştirimler

Modeller, bilgisayar ağı protokollerinin sağlaması gereken servisleri tasarlamaktadır. En popüler olan model ISO tarafından tasarlanan yedi katmanlı OSI modelidir. OSI modelinden önce tanımlanan bir model, Birleşik Devletler Savunma Bakanlığı, (Department of Defence) tarafından geliştirilmiş olan (DoD) modelidir. Bu model, 1970'lerin ortasında geliştirilmiş TCP/IP protokol suite'in modelidir. Yazılım ve donanım mühendisleri, protokolleri gerçekleştirerek ürünler oluşturmaktadırlar. Belli bir protokolü birden fazla firma gerçekleştirse bile aralarında farklılıklar olacaktır. Ancak eğer protokoller, doğru olarak gerçekleştirilirse, farklı ürünlerin aynı protokol ile birbirleri iletişimi problem olmayacaktır [3].

2.3 OSI Referans Modeli

OSI Referans Modeli International Standards Organization (ISO) tarafından sunulan bir model üzerine geliştirilmiştir. Bu model ISO OSI (Open Systems Interconnection) Referans Modeli olarak anılır zira açık sistemlerin yani diğer sistemlerle haberleşmeye açık sistemlerin bağlantısı ile ilgilenir. OSI modeli yedi tabakadan oluşur. Bu tabakaların oluşturulmasında uygulanan prensipler: **1**-Değişik seviye bir ayırım gerektiğinde bir tabaka oluşturulmalıdır.

2-Her tabaka iyi tanımlanmış bir fonksiyonu yerine getirmelidir.

3-Her tabakanın fonksiyonu uluslararası standartlaştırılmış protokoller açısından seçilmelidir.

4-Tabaka sınırları arabirimler arası bilgi akışını en aza indirecek şekilde seçilmelidir.

5-Tabakaların sayısı belirgin fonksiyonların aynı tabakalar üzerinde atlama yapmayacak kadar geniş, mimariyi hantallaştırmayacak kadar az olmalıdır. 7 tabakalı modelin sıradüzensel yapısı Tablo 2-3 de verilmiştir [4].

Tablo 2-3: Yedi tabakalı model

TABAKA	FONKSİYON	PROTOKOL
Tabaka 7	Uygulama katmanı (Application Layer)	S5 Header, Modbus TCP,vb
Tabaka 6	Sunum katmanı (Presentation Layer)	
Tabaka 5	Oturum katmanı (Session Layer)	
Tabaka 4	Taşıma katmanı (Transport Layer)	TCP
Tabaka 3	Ağ katmanı (Network Layer)	IP
Tabaka 2	Veri bağlantısı katmanı (Data link Layer)	
Tabaka 1	Fiziksel katman (Physical Layer)	

Yedi Tabakalı Model Tanımlanan yedi tabaka:

7) Uygulama Katmanı:

Uygulamalara değişik servisler sağlar. Kullanıcı tarafından çalıştırılan tüm uygulamalar bu katmanda tanımlıdırlar. Bu katmanda çalışan uygulamalara örnek

olarak, FTP (File Transfer Protocol), SNMP (Simple Network Management Protocol), e-posta uygulamalarını verebiliriz.

6) Sunum Katmanı:

Sunum katmanı, kullanıcıların problemleri kendi başlarına çözüm bulmaları yerine onlara yeterli bir genel çözüm sunar. Kısaca, diğer alt katmanların aksine, bit'leri bir uçtan diğerine güvenilir bir biçimde iletimleri ile ilgilenmek yerine oturum katmanı iletilen bilginin söz dizimi ve semantiği ile ilgilenir.

Sunum servislerine tipik bir örnek standart, üzerinde anlaşılan bir şekilde veriyi kodlamaktır. Birçok kullanıcı programları rast gele bit dizilerini kendi aralarında değişimini gerçekleştirmez. Şahıs adları, tarih, para gibi şeyleri değişirler. Bu başlıklar, karakter dizileri, tamsayılar, kayan nokta numaraları gibi daha basit veri yapıları olarak ifade edilirler. Değişik bilgisayarlar karakter dizileri ve tamsayıları ifade etmek için değişik kodlar kullanırlar. Bu bilgisayarlar arasında veri değişimini standartlara uygun olarak yerine getirmek sunum katmanının işidir.

Sunum katmanı ayrıca bilginin sunulmasının diğer yönleri ile de ilgilidir. Örneğin veri sıkıştırması iletilmesi gereken bir sayısını artırmak için kullanılabildiği gibi kriptografi güvenlik ve kullanıcı doğrulaması için sık sık kullanılır.

5) Oturum Katmanı:

Haberleşme ile ilgili olmayan problemlerle ilgilenir. Oturum katmanı, değişik makinelerdeki kullanıcıların birbirleri arasında oturumlar açmasını sağlar. Bir oturum taşıma katmanının yaptığı gibi sıradan veri taşıma işini gerçekleştirdiği gibi, bazı uygulamalarda çok yararlı gelişmiş hizmetler de sunar. Bir oturum bir kullanıcının uzaktaki zaman-paylaşımlı bir sisteme bağlanmasını (Log on, log in) veya iki makine arasında dosya transferi yapmasını sağlar.

Oturum katmanın sunduğu hizmetlerden biri de sistemlerin karşılıklı iletimlerinin yönetimidir. Oturumlar aynı anda tek yönlü veya aynı anda çift yönlü veri akışına izin verebilirler. Eğer trafik tek yönlü ise oturum katmanı iletim sırasının kimde olduğu konusunda yardımcı olur.

İlgili diğer bir oturum hizmeti jeton (token) yönetimidir. Bazı protokoller için, her iki tarafın aynı anda aynı işlevi yerine getirmeye çalışmaması çok önemlidir. Bu aktiviteleri yönetmek için oturum katmanı taraflar arasında değiştirilebilecek jetonlar sağlar. Jetona' a sahip taraf kritik uygulamayı çalıştırma hakkına sahip olur.

Diğer bir oturum servisi senkronizasyondur. Oturum katmanı veri akımının içine kontrol noktaları yerleştirir böylelikle bir çökmeden sonra en son kontrol noktasından sonraki veri gönderilir.

4) Taşıma Katmanı:

Uçtan uca haberleşme kontrolünü sağlar. Aktarım Katmanı ağın karmaşıklığını üst katman işlevlerinden gizlemek için tasarlanmıştır. Yüksek düzey mesajları bölümlere (segment) ayırır ve bölümleri güvenilir biçimde Oturum ya da daha üst katmanlara iletir.

Aktarım Katmanı genelde güvenilir veya bağlantı-tabanlı servislerin yokluğunu dengelemektedir. Güvenilir kelimesi tüm verinin her zaman ulaştırılabileceği anlamına gelmemektedir. Örneğin eğer kabloda bir kopukluk olursa Aktarım Katmanı verinin ulaştırılmasını garanti edemez. Halen, güvenilir Aktarım Katmanı protokol gerçekleştirimleri genelde verinin alındığı onaylar ya da reddeder. Eğer alıcı birimde veri düzgün biçimde alınamamışsa, Aktarım Katmanı verinin yeniden gönderilmesini başlatabilir ya da üst katmanları haberdar eder. Üst katmanlar gerekli düzeltme işlemini gerçekleştirebilir ya da kullanıcıya bazı seçenekler sunabilir.

3) Ağ Katmanı:

Ağ üzerinde bilgiyi yönlendirir. Ağ katmanın temel görevi verinin ağın belirli konumlarına aktarılmasını sağlamaktır. Bu işlev Veri-bağlantı katmanının fiziksel adresler üzerinden yaptığına benzerdir. Ancak Veri-bağlantı katmanın yaptığı

işlem tek bir ağ üzerindedir. Ağ katmanı, ağlar arası iletişim (internetwork) olarak adlandırılan birbirinden bağımsız birçok ağ arasındaki veri alış verişini düzenler.

Veri-bağlantı katmanı adreslemesi tek bir ağa veri iletimini sağlar ve alıcılara ulaşan verinin onlar için bir anlam ifade ettiği varsayımına dayanır. Buna karşılık Ağ katmanı ağlar arası iletişim (internetwork) içerisinde veriyi belirli bir ağa yönlendirir ve ilgili olmayan ağlara veri göndermez. Ağ katmanı bu işlemi anahtarlama, adresleme ve yol belirleme algoritmaları kullanarak gerçekleştirir. Ağ katmanı aynı zamanda bir ağlar arası iletişim (internetwork) içerisinde aynı niteliklere sahip olmayan ağlar arasında doğru rota (yol) belirlenmesinden de sorumludur.

Ağ katmanı konu ve yöntemleri;

• Mantıksal olarak ayrı olan ağlarınız tek bir ağ adresine sahip olurlar,
• Anahtarlama ile ağlar arası iletişim (internetwork) üzerinde bağlantıların nasıl yapıldığı belirlenir,

• Yol belirleme işlemi, bilgisayar ve yol atayıcıların (router) uygun veri hattını belirleyecek şekilde gerçekleştirilir,

• Ağlar arası iletişim (Internetwork) içerisinde kabul edilen hata düzeyine göre değişik ağ bağlantı servisleri gerçekleştirebilir.

2) Veri Bağlantısı Katmanı:

Bağlı uçlar arasında hata denetimini sağlar. Veri-Bağlantı katmanı protokol gerçekleştirimlerinin basit amaçları aşağıda açıklanmıştır.

• Fiziksel katmanın bitlerini çerçeve (frame) olarak adlandırılan mantıksal birimler halinde gruplandırmak (Bayt gibi çerçeve (frame) de sürekli devam eden bitler serisidir).

• Hata kontrolü (ve bazen düzeltilmesi),

• Veri akış kontrolü,

• Ağ üzerindeki bilgisayarların tanınması.

Diğer katmanların çoğu gibi Veri-Bağlantı katmanı da kendi kontrol bilgisini veri paketinin başına ekler. Bu başlık alanında, kaynak ve hedef adres bilgisi (fiziksel ya da donanımsal), çerçeve (frame) boyu bilgisini ve kapsanan üst katmanlar bilgisini içerir.

Aşağıdaki ağ bağlantı elemanları Veri-Bağlantı Katmanı kapsamındadır.
• Köprüler (Bridge)[1],

• Akıllı göbekler (intelligent Hub),

• Ağ arabirim kartları (Network Interface boards).

Veri-Bağlantı Katmanının fonksiyonları genelde aşağıdaki iki alt katman arasında bölünmüş biçimde anılır.

1) Fiziksel Katmanı:

[1] Aynı adresleme yöntemine sahip, aynı tür fiziksel iletim ortamını kullanan, ancak veri bağı katmanı protokolünde farklılaşan iki yerel alan ağı arasında iletişime elveren bağlantı

İletim ortamına bağlantıyı sağlar. OSI Fiziksel katman protokol gerçekleştirimleri bitlerin aktarımındaki kuralların düzenlemesini gerçekleştirir. Fiziksel katman şunları tanımlar:

• Fiziksel ağ yapıları,

•Aktarım ortamının kullanımındaki mekanik ve elektriksel belirlemeler,

• Bit aktarım kodlama ve zamanlama kuralları.

Fiziksel katman, aktarım ortamının tanımını içermez. Bunun yanında Fiziksel katman protokol gerçekleştirimleri aktarım ortamına özeldir. Aşağıdaki ağ bağlantı birimleri Fiziksel katmanın kapsamındadır:

•Yoğunlaştırıcı, göbek (Hub), yineleyici (repeater) gibi elektriksel sinyal üreten birimler.

•Birimlerin aktarım ortamıyla bağlantısını sağlayan, aktarım ortamı bağlantı donanımları.

•Sayısal-Analog çevrimi yapan modem ve kodlayıcı-kod çözücüler (codec). OSI Fiziksel katmanı, ağ üzerinde birimlerin birbirlerine göre nasıl yerleştiklerini ve fiziksel aktarım ortamından bitlerin ne şekilde aktarılacağını belirler. Fiziksel katman içinde, bağlantı türleri, fiziksel topoloji, Sayısal ve Analog sinyal kodlama teknikleri, bant kullanımı, bit senkronizasyonu ve çoğullama (multiplexing) önemli kavramları oluşturmaktadır.

2.4 Taşıma Protokolleri

2.4.1 Aktarım Kontrol Protokolü [Transmission Control Protocol (TCP)]

TCP, IP'nin bir üst katmanında çalışan iki aktarım katmanı protokolünden birisidir.

TCP, güvenilir ve sanal devre üzerinden çalışan bir protokoldür. Aynı ağ üzerinde ya da farklı ağlar üzerinde iki katılımcının (Host) birbirleriyle güvenilir bir şekilde haberleşmesini sağlar.

TCP‘nin başlıca özellikleri şunlardır:

•Bir bağlantının (connection) kurulması ve sonlandırılması

•Güvenilir (Reliable) paket dağıtımının sağlanması

•Akış kontrolü (flow control) olanağı ile katılımcılarda veri taşmasının (overflow) önlenmesi

•Bozulmuş ya da ikilenmiş verinin düzeltilmesi (error recovery)

•Alıcı katılımcı içerisinden birçok uygulama arasında çoğullama çözülüm (*demultiplexing)* yapılması

TCP istemci katılımcı üzerinde çalışan bazı uygulama katmanı protokolleri:

•TELNET

•Dosya Aktarım Protokolü [File Transfer Protocol (FTP)]

•Temel Posta Aktarım Protokolü [Simple Mail Transfer Protocol (SMTP)]

TCP, IP ‘den hizmet alan bir protokoldür. IP, TCP ‘nin sağladığı olanaklarla güvenli veri dağıtımı yapar. IP ‘nin TCP ‘ye sağladığı ana hizmetler şunlardır.

•Farklı ağlar üzerindeki kaynak ve hedef istasyonların adreslenmesi

•Internet üzerinde veri paketlerinin yönlendirilmesi

•Ağlar arasında küçük boyutlu paketleri iletebilen ağlarda daha sonra birleştirilmek üzere paket parçalama (fragmentation) fonksiyonunun yerine getirilmesi

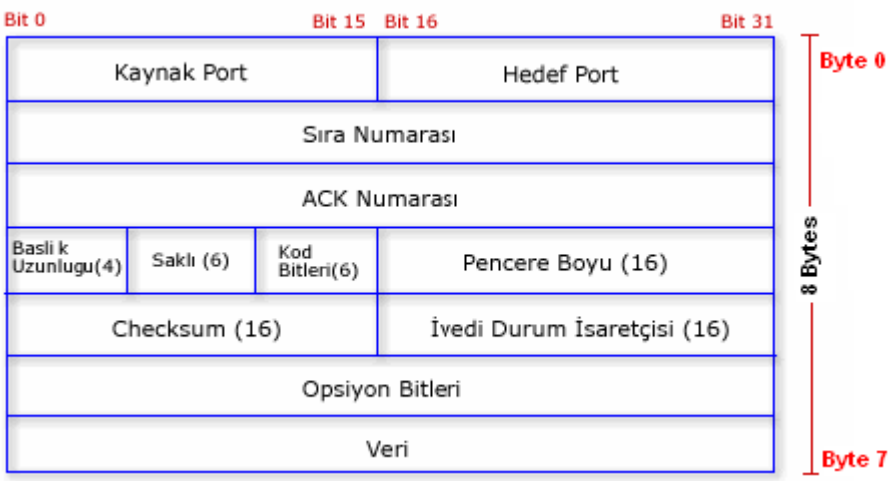
•IP paketinin protokol numarasına bakarak bu paketleri üst katmana yönlendirme işinin yapılması

•Yaşam Süresi [Time to Live (TTL)] alanına yerleştirilen değerle paketin internet üzerinde ömrünün sınırlandırılması

Bir TCP paketi örneği şekil 2-4'de verilmiştir.

TCP'nin Temel İşlevleri:

•TCP'nin üzerinde çalıştığı farklı katılımcılar (Host) arasında güvenilirliğe sahip, bağlantı tabanlı paket dağıtımını sağlar.



Şekil 2-4: TCP paket yapısı

•TCP, Internet ortamında şu işlevleri sağlar:

•Temel Veri Aktarımı (Basic Data Transfer)

•Güvenilirlik (Reliability)

•Uçtan Uca Akış Kontrolü (End to end flow control)

•Çoğullama (Multiplexing)

•Bağlantılar (Connections)

•Tam Çift Yönlü İşlem (Full duplex process)

•Temel Veri Aktarımı (Basic Data Transfer):

Haberleşen TCP katılımcılar üzerinde bu katmanlar arası bölüt (segment) aktarımı yoluyla haberleşme sağlanır. TCP veri akışını baytları sıralandırıp bölüt grupları halinde iletir. Eğer bir parçalama gerekliliği ortaya çıkmadıysa her bölüt bir IP paketine konarak iletilir.

Güvenilirlik (Reliability):

TCP, zarara uğramış, bozulmuş, ikilenmiş verinin doğru olarak elde edilmesinden sorumludur.

TCP, her bir bayt ’a sıra numarası verir. Daha sonra ilettiği bu baytlara karşılık onay (acknowledge) bekler. Eğer belirli aralıklarla beklediği onayları alamazsa onay alamadığı kısımları yeniden hedef katılımcıya iletir. Hedef katılımcı sıra numaralarına göre bölütleri sıralarken aynı bölüt numarasına sahip iki bölütle karşılaşabilir. Her bir bölüt sağlama toplamı (checksum) denilen kontrol bilgileri içerir. Bu kontrol bilgilerine göre hasara uğramış bölütler anlaşılır ve atılır. Kaynak katılımcıya onay gönderilmezse kaynak katılımcıdaki TCP onay alamadığı bölütleri yeniden gönderir.

Uçtan Uca Akış Kontrolü (End to end flow control):

Kaynak istasyondan hedefe iletilen veri miktarını kontrol eden bir mekanizmadır. Bu mekanizma pencere yönetimi (window management) tekniği ile sağlanır.

Çoğullama (Multiplexing):

TCP üzerinde birden fazla eş zamanlı TCP bağlantı kurmakta kullanılır.

Bağlantılar (Connections):

Bağlantı iki TCP katılımcının veri transferi için birbirlerinden haberdar olmaları ve gerekli hazırlıkları tamamlamış olmaları demektir. Bağlantı kurulumu için kaynak katılımcı hedef katılımcıya bağlantı kurmak istediğini bildirir. Hedef katılımcı gelen istekteki bağlantı noktası (port) numarasına karşılık gelen hizmeti sağlıyorsa ve bağlantı için hazırsa bir cevap mesajı göndererek bağlantı işlemini başlatır. Bağlantı, kaynak ve hedef katılımcılar arasında sanal bir devre oluşturulmasıyla kurulur. Bağlantı kurulup transfer gerçekleştirildikten sonra bağlantı koparılır.

Haberleşme sırasında, her bir katılımcı üzerindeki TCP protokolü, sürekli hata kontrolü yaparlar. Eğer herhangi bir nedenle bir iletişim problemi ortaya çıkarsa, her

iki uçtaki TCP yürütücüsü bu problemden haberdar olur ve üst katman protokollerini uyarırlar.

Bağlantının kurulmasında TCP paketinin başlığı içindeki SYN (Sync) ve ACK (onay) bitlerini kullanılır.

Bağlantı kurulumunda bağlantı talebinde bulunan ve bağlantıyı kabul eden katılımcılar arasında iki farklı durum oluşur.

Bağlantıyı kabul eden katılımcı pasif, bağlantı talebinde bulunan katılımcı aktif olur.

Tam çift yönlü işlem (Full Duplex Processs):

TCP işlemleri hem gönderim hem de alımın aynı anda yapılmasına olanak tanır. Bu da haberleşmenin daha hızlı olmasını sağlar.

Sıra Numarası (Sequence Number):

TCP üst katmandan aldığı veriyi bölütlere böler. Bu bölütlerin her biri genelde tek bir IP paketi içinde taşınır. Her bir bölüte TCP bir numara verir. Amaç ağlar üzerinde dolaşan bu bölütlerin hedefe varış sıralarının karışması durumunda hedef katılımcıda çalışan TCP yürütücüsünün bunları tekrar sıraya koyup üst katmana aktarmasıdır. (bölüt boyları sabit değildir)

TCP, karşı TCP ile bağlantıyı ilk kurduğunda, ilk gönderdiği bölüt'e bir numara verir ve daha sonraki bölüt'e numaraları sıralı olarak artan şekilde devam ettirir. Sıra numarası $0 - 2^{31}$ arasında olabilmektedir. Çünkü TCP paketi veri yapısında sıra numarası 4 baytlık bir alanla ifade edilir

TCP verideki bayt'ları gruplayarak bölütleri oluşturur ve her bir bölüt ayrı bir numara ile numaralandırılır. Bir bölüt bir numara aldığında bu bölüt numarasını içinde barındırdığı ilk bayt'a verir. Yine içinde barındırdığı diğer baytlara bu numaranın artanlarını verir. Bu bölütten sonra gelen bölütün alacağı numara, bir önceki bölüt'ün içindeki en son baytın aldığı numaranın bir fazlası olacaktır. Bu sıra numaraları TCP başlığı içinde taşınır.

2.4.2 Kullanıcı Datagram Protokolü [User Datagram Protocol (UDP)]

UDP, TCP / IP protokol grubunun iki aktarım katmanı protokolünden birisidir.

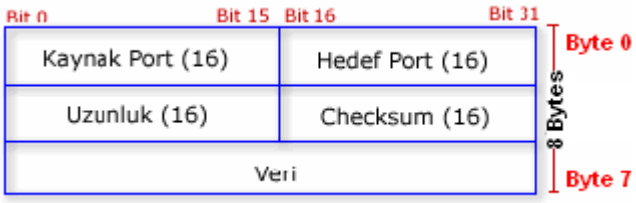
UDP, onay (Acknowledge) gönderip alacak mekanizmalara sahip değildir. Bu yüzden veri iletiminde başarıyı garantileyemez. Yani güvenilir bir aktarım servisi sağlamaz. Hedefe ulaşan paketler üzerinde sıralama yapıp doğru veri aktarımını da sağlayacak mekanizmalara sahip değildir. Uygulamalar güvenli ve sıralı paket dağıtımı gerektiriyorsa UDP yerine TCP protokolü tercih edilmelidir.

UDP, minimum protokol yükü (overhead) ile uygulama programları arasında TCP ye göre basit ama daha hızlı bir aktarım servisi sağlar.

Bağlantı noktası (port) numarasıyla çoğullama çözüm (demultiplexing) :

IP, UDP bağlantı noktası (port) numaraları aracılığıyla pek çok datagram paketi arasında ayrıştırma yapabilecek mekanizmaları içerir. Bu yolla birçok uygulamanın alıcı katılımcı üzerinde aynı anda çalışmasına ve ağ üzerinden haberleşmesine olanak tanır. UDP paketleri de ağ üzerinde IP paketleri içinde taşınır.

Bir UDP paketi örneği şekil 2-5'te verilmiştir.



Şekil 2-5: UDP paket yapısı

Kaynak Bağlantı Noktası (Source Port): Opsiyonel bir alandır. Gönderilen işlemin bağlantı noktasını gösterir. Eğer gönderen katılımcı bir kaynak numarasına sahip değilse bu alan "0" ile doludur.

Hedef Bağlantı Noktası (Destination Port): Hedef katılımcı içerisinde, işlemlere uygun ayrımları yapmak için kullanılır.

Uzunluk (Length): UDP veri ve UDP başlığının bayt cinsinden toplam uzunluğudur.

Hata Kontrolü (Checksum): Opsiyonel bir alandır. Hata kontrol mekanizması sağlar. Eğer hata kontrolü yapılmayacaksa bu alan "0" ile doludur.

UDP 'deki hata kontrolü sadece hatasız taşımanın bir ölçüsüdür. Yeniden gönderim veya güvenilirlik sağlamaz.

UDP ' nin bu kadar dezavantajına rağmen daha çok kullanıldığı açıktır. TCP bir veri karşıya 6x32+Veri boyu kadar bir paket olarak gitmektedir. Yani her paket fazladan 192 bit başlık (header) bilgisi taşımaktadır. Oysa UDP paketleri 64 bitlik başlık (header) bilgisine sahiptir.

UDP kullanmanın en önemli nedeni az protokol yüküdür. Video sunucu gibi gerçek zamanlı veri akışı gerektiren bir uygulama için TCP fazla yük getirir ve görüntü gerçek zamanlı (realtime) oynamaz. Bu nedenle çoğa gönderim (multicast) uygulamalarında Datagram soketler kullanılır. Ayrıca video ve ses görüntülerinde genelde az bir veri kaybı sesi veya görüntüyü bozmaz. Bu nedenle sıkı paket kontrolüne gerek yoktur. Eğer iyi bir fiziksel bağlantınız varsa hata oranı düşük olacaktır ve bu nedenle TCP'nin yaptığı hatalı paket kontrol işlemleri fazladan yük olacaktır.

UDP her ne kadar paket güvenliğini denetlemese de bunu yazılımcı yapabilir. Örneğin TCP bir paketi gönderdiğinde karşı tarafın onu aldığını anlamak için onay, kabul (acknowledgement) bekler. UDP bunu yapmaz. Fakat bunu soket yazılımcısı yapabilir. Yazılımcı, gönderilen her pakete bir cevap bekleyerek bunu sağlar.

2.4.3 TCP üzerinden ISO (ISO over TCP/RFC1006) Protokolü

TCP protokolü ile veri iletimi veri katarları (stream) şeklindedir. Bu kullanıcı tarafından paketlenen verilerin, alıcı istasyon tarafından aynı paketlerde alınamamış

olması olasılığını doğurur. Veri uzunluğuna bağlı olaraktan veriler doğru sekansta fakat farklı olarak paketlenmiş biçimde alınmış olabilirler. Bu alıcının paket sınırlarını artık doğru olarak ayırt edememesini doğurur. Örneğin gönderici 2 adet 10 Baytlık veri paketi gönderdiğinde bunlar alıcıda 20 Baytlık tek bir paket olarak alınmış olabilir. Fakat birçok uygulama için doğru paketleme çok önemlidir.

Bundan dolayı TCP üzerinde ayrı bir protokole daha ihtiyaç vardır. Bu amaç RFC1006 protokolünde açıklanmıştır ve bu protokol TCP (RFC793) taşıma ara yüzünü temel alan bir ISO taşıma ara yüzünü (ISO 8072) açıklar. Temel RFC1006 protokolü ISO 8073 teki TP0 (Transport Protocol, Class 0) ile hemen hemen aynıdır. RFC1006 bir TCP protokolü olarak çalışır ve paketlerin deşifre işlemi TCP paketinin veri kısmında yapılır.

RFC1006'nın temel özellikleri şunlardır:

TCP'nin aksine bir telegramın içeriği alıcı tarafından onaylanır.

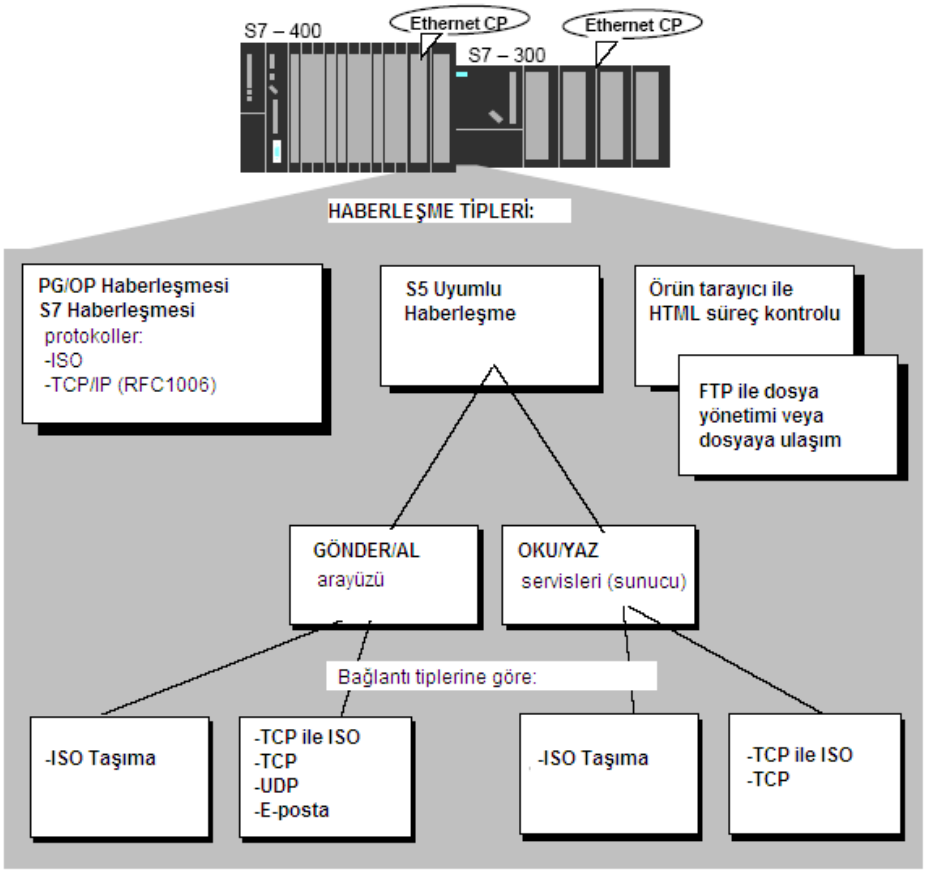
Bağlantı noktaları ve IP adresleri yerine TSAPlar kullanılır.TSAP uzunluğu 1 ile 16 bayt arasında değişebilir, adresleme girişi ASCII formatında veya Onaltılık sayı düzeninde yapılabilir. Gönderici ve alıcı TSAPları sadece bir bağlantı için aynı olabilir.

TCP'den farklı olarak RFC1006 ile farklı uzunluktaki veriler alınabilir.

3.SIEMENS S7 ETHERNET HABERLEŞMESİ

3.1 S7 Ethernet CPler ile Simatic S7 haberleşme olanakları

Siemens Ethernet haberleşme işlemcileri işlemci tiplerine göre aşağıdaki haberleşme tiplerini destekler (şekil 3-1).



Şekil 3-1: Siemens Ethernet haberleşme tipleri

PG/OP Haberleşmesi

PG/OP haberleşmesi işlemcilere program ve donanım yapılandırmasını yüklemek, test ve tanı koyma işlevlerini çalıştırmak ve bir operatör paneli ile çalışılıyorsa, sistemin izlenmesi amacıyla kullanılır.

S7 Haberleşmesi

S7 haberleşmesi S7 PLCler ile PGler arasında fonksiyonlar yardımıyla basit ve etkin bir haberleşme ara yüzüdür.

S5 Uyumlu Haberleşme

•GÖNDER/AL ara yüzü

CP tipine bağlı olarak, gönder/al ara yüzü başka bir S7 PLC'ye, S5 PLC'ye, PG/OP' ye veya başka bir istasyona program kontrollü önceden yapılandırılmış bağlantı sağlar.

•OKU/YAZ servisleri (sunucu)

S7 CPU'ların hafıza alanlarına S5 PLC'lerden veya başka istasyonlardan (PC yazılımı v.b.) doğrudan ulaşım sağlar ve sadece istemcinin istekte bulunduğu veriyi bir başlıkla beraber gönderir.

HTML Süreç Kontrolü

IT-CP'ler, Örün tarayıcı (web browser) ve özel fonksiyonlar aracılığıyla süreç kontrolünü mümkün kılmaktadır.

FTP ile Dosya Yönetimi veya Dosyaya Ulaşım

S7-400 ailesinde IT-CPler FTP sunucu veya FTP istemci olarak çalışabilmektedir. Bu yöntemle veri bloklarının iki yönde iletimi mümkündür.

3.2 Çeşitli Cihazlar Arasında Mümkün Olan Haberleşme Tipleri

PC ve çeşitli tip Siemens PLC'ler ve bunların arasında mümkün olan haberleşme tipleri Tablo 3-1 de gösterilmiştir. Bu tabloda PLC'lerin kendi aralarında haberleşebildikleri gibi destekledikleri bütün servis ve protokoller aracılığıyla PC'lere haberleşme yönünden açık olduğunu görmekteyiz. Burada bizim PLC'nin hafıza ve G/Ç alanlarına ulaşmak için kullanacağımız servis Oku/Yaz servisleridir. Bu servis ve bu servis ile kullanılan S5 Header protokolü ileriki bölümlerde

ayrıntılı bir biçimde incelenecektir. Sadece Siemens'in kendi donanımları tarafından desteklenen S7 Haberleşmesi ve PG/OP Haberleşmesi bu tezde incelenmeyecektir. Kısa bir bilgi vermek gerekirse bu haberleşme türleri Siemens'in hâlihazırda seri haberleşme amacıyla kullandığı iki protokolün TCP/IP üzerinde taşınan birer çeşididir.

Tablo 3-1: Cihazlar arasında haberleşme tipleri

CİHAZ	S7-300	S7-400	S5	PC
S7-300	GÖNDER/AL	S7Haberleşmesi GÖNDER/AL	GÖNDER/AL OKU/YAZ	PG/OP Haber. S7Haberleşmesi GÖNDER/AL OKU/YAZ HTML Süreç Kont.
S7-M7-400	S7Haberleşmesi GÖNDER/AL	S7Haberleşmesi GÖNDER/AL FTP Servisleri	GÖNDER/AL OKU/YAZ	PG/OP Haber. S7Haberleşmesi GÖNDER/AL OKU/YAZ HTML Süreç Kont. FTP Servisleri

3.3 SIMATIC S7 PLC'ler ile S5 Uyumlu Haberleşme Bağlantıları

3.3.1 S5 Uyumlu Haberleşmede Gönder/Al (Send/Receive) ara yüzü

Bu haberleşme ara yüzü ile S7 programının içinde kullanılan gönder ve al fonksiyonları ile programdan tetiklenen bağlantılar kurulabilir. Bu bağlantı tipi ile S7, S5 veya PC istasyonlarına veri blokları halinde bilgi gönderilip alınabilir.

Bu haberleşme tipi aşağıdaki protokolleri kullanabilir:

-ISO

-TCP ile ISO

-TCP

-UDP

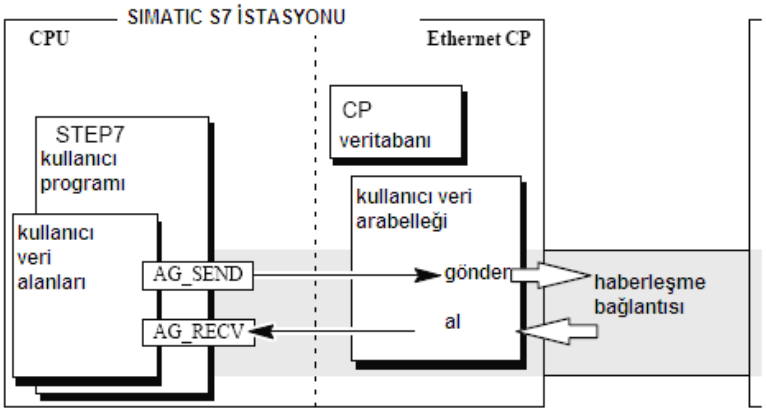
Bir bağlantıda iş başına CP'nin gönderebileceği veri uzunluğu Tablo 3-2'de gösterilmiştir.

Tablo 3-2: Gönder/Al ara yüzü bir iş içinde gönderilebilecek veri miktarları

	ISO	TCP ile ISO	TCP	UDP
Gönder	8192 Bayt	8192 Bayt	8192 Bayt	2048 Bayt
Al	8192 Bayt	8192 Bayt	8192 Bayt	2048 Bayt

3.3.2 Gönder/Al programlama fonksiyonları

Kullanıcı programından gönder/al ara yüzünü kontrol etmek için AG_SEND/AG_LSEND (FC5/FC50) ve veri uzunluğu 240 bayttan uzun olduğunda AG_RECV/AG_LRECV (FC6/FC60) fonksiyonları kullanılır [4].



Şekil 3-2: Gönder/Al ara yüzü

Gönder/Al ara yüzünün genel yapısı Şekil 3-2' de gösterilmiştir. Program içinde döngüsel olarak çağırılan AG_SEND ve AG_RECV fonksiyonları kullanıcı verisi ile CP arabelleği arasındaki alış verişi sağlar. Gönderim sırasında CP arabelleğinde tutulan veriler haberleşme bağlantısı üzerinden ilgili katılımcıya ulaştırılır. Aynı şekilde diğer katılımcılardan alınan veriler de CP arabelleğinde saklanıp, uygun zamanda kullanıcı programına dâhil olurlar [4].

OB35:

NETWORK 1: // Alma (Receive) bloğunun çevrimsel çağırılması

```
CALL  "AG_LRECV"

ID    :=2            //Job ID  (Netpro'da kurulumla belirlenir)

LADDR :=W#16#100     //Onaltılık tabanda modül adresi

RECV  :="RECEIVE_DB".DB_VAR   //Alınan verinin yazılacağı bellek alanı
```

NDR :=M200.0

ERROR :=M200.1 //Hata biti

STATUS:=MW202

LEN :=MW204 //Alınan veri uzunluğu

NETWORK 2: //Gönderme (Send) bloğunun çevrimsel çağırılması

CALL "AG_LSEND"

ACT :=M100.0 //Gönderimi başlat biti

ID :=1 //Job ID (Netpro'da kurulumla belirlenir)

LADDR :=W#16#100 //Onaltılık tabanda modül adresi

SEND :="SEND_DB".DB_VAR //Gönderilecek verinin bulunduğu bellek

LEN :=10 //Gönderilecek veri uzunluğu

DONE :=M100.1 //İş tamamlandı

ERROR :=M100.2 //Hata

STATUS:=MW102

SET //İş başlatma biti M100.0'ı iş tamamlandığında veya hata

R M 100.0 //olduğunda sıfıra çekip tekrar bir yapan program parçası

A M 100.1 //Yeni bir gönderim için blok bu işarette çıkan kenar ister

JC tamam

A M 100.2

JC hata

```
 BEU

tamam: NOP   0

SET

S    M    100.0

BEU

hata:  NOP   0

SET

S    M    100.0
```

Gönder/al ara yüzü için bir program örneği yukarıda verilmiştir. Program "SEND_DB" veri bloğunun içeriğini hata gönderim aktif olmadığı sürece göndermektedir. Fakat ilk gönderimin başlayabilmesi için OB100 başlangıç bloğu içinde M100.0 biti set edilmelidir. Bu sırada çevrimsel olarak alınan veriler ise "RECEIVE_DB" veri bloğunun içine yazılmaktadır. Bu programda dikkat edilmesi gereken fonksiyonların ID girişine donanım yapılandırma işlemi sırasında tanımlanan doğru iş numaralarını vermektir. Alınan ve gönderilen verinin kullanıcı programı içinde yanlış değerlendirilmesini önlemek amacıyla programın OB1 de değil de zamanda kesmeli çalışma bloklarından birinde (Örneğin OB35) çalıştırılıp kesme zamanının da ana çevrim süresinin biraz üzerinde belirlenmesinde fayda vardır.

3.3.3 S5 Uyumlu Haberleşmede Oku/Yaz (Fetch/Write) Servisleri

Oku/yaz servisleri genellikle S7 PLC'leri, S5 veya Simatic harici sistemlere bağlamak için kullanılır.

Oku: Diğer ağ katılımcıları PLC'nin sistem verilerini okuyabilirler.

Yaz: Diğer ağ katılımcıları PLC'nin sistem verilerine yazabilirler.

Bu haberleşme tipi aşağıdaki protokolleri kullanabilir:

-ISO

-TCP ile ISO

-TCP

Kullanıcı programında AG_LOCK (FC7) ve AG_UNLOCK (FC8) fonksiyonları aracılığıyla bu okuma yazma işlemine izin verilebilir veya engellenebilir. Bunun dışında eğer bir engellemeye ihtiyaç duyulmuyor ise özel bir kullanıcı programına ihtiyaç duyulmaz. Haberleşme protokolünü CP kendi işlemcisi üzerinden yürütür. Bu haberleşme tipi ilerideki bölümlerde detaylı olarak anlatılacak "S5 Header" protokolü üzerinden çalışır. CP 400 açısından pasif bir bağlantı tipidir dolayısıyla CP sunucu (server) olarak çalışır. S5 PLC'lerde ise hem aktif hem de pasif Oku/Yaz servisleri oluşturmak mümkündür. S5 kullanıcı programında Fetch active/passive ve Write active/passive fonksiyonları kullanılarak bu servislerin yürütülmesi sağlanır.

Oku/yaz servisleri S5 Header adı verilen bir uygulama protokolü üzerinden yürütülür bu protokol ile ilgili ayrıntılı açıklamalar 4.Bölüm altında verilmiştir.

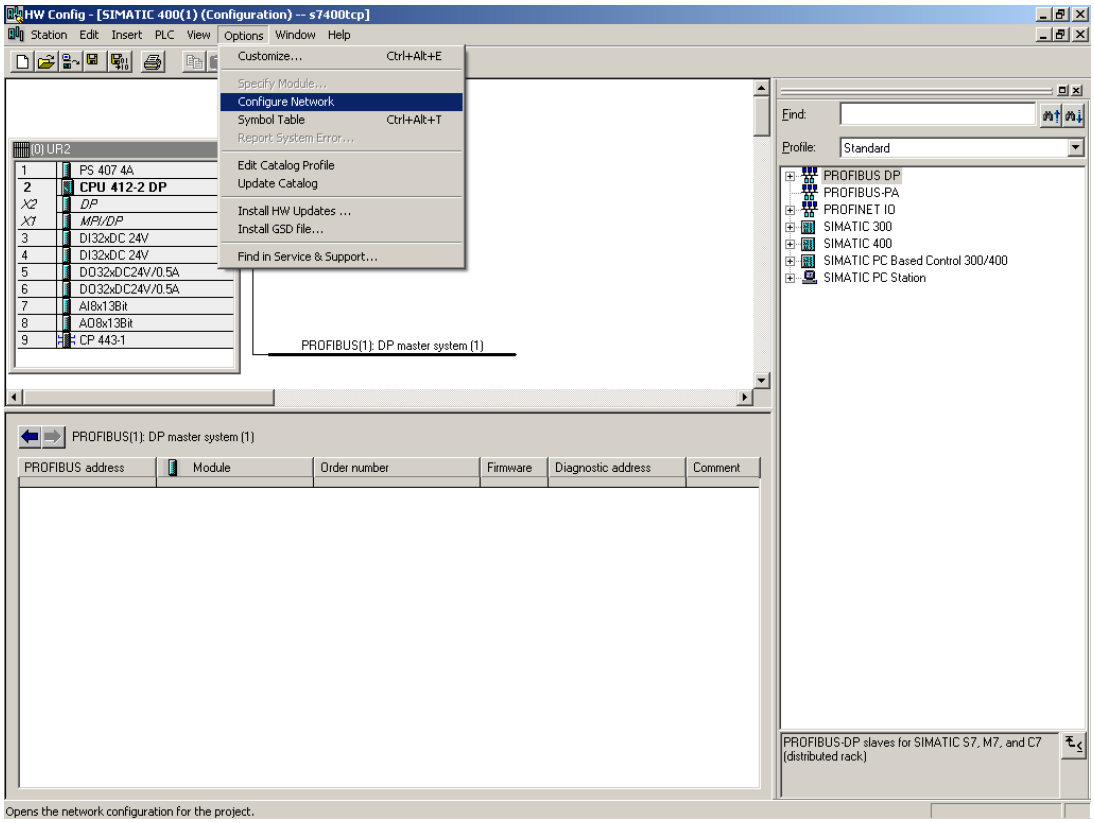
3.4 CP443-1'de Haberleşme Konfigürasyonlarının Yapılması

PLC'nin evvelki bölümde açıklanan haberleşme protokollerini yürütebilmesi veya haberleşme servislerini açabilmesi için CP443-1 haberleşme işlemcisinin konfigüre edilmesi gerekmektedir. Bu işlemler Simatic Manager altında bir alt program olan Netpro'da yapılmaktadır. Netpro programına Hardware Manager altında Options>>Configure Network menüsünden ulaşılabilir. Program çalıştırmadan evvel PLC'nin donanım yapılandırılması yapılmış olmalıdır [5] .

3.4.1 Servislerin ve Ara yüzlerin Konfigüre Edilmesi

Donanım yapılandırılması yapılmış bir PLC HW Configurator'da Şekil 3-3'deki gibi gözükmektedir. CP443-1 üzerine çift tıklanarak özellikler (Properties) penceresi açılır, Ara yüz (Interface) çerçevesi içindeki özellikler (Properties) butonu ile ara yüz

özellikleri penceresi açılır. Parametreler (Parameters) sekmesi altında Yeni (New) butonu ile yeni bir Ethernet hattı yaratılıp haberleşme işlemcisi bu hatta link edilir. Bundan sonra modüle bir IP adresi atamak gerekmektedir, evvelki bölümde anlatılan kurallar doğrultusunda ağ planlaması yapılıp bu planlamaya uygun olarak işlemciye bir IP ve Alt Ağ Maskesi adresi verilmelidir.

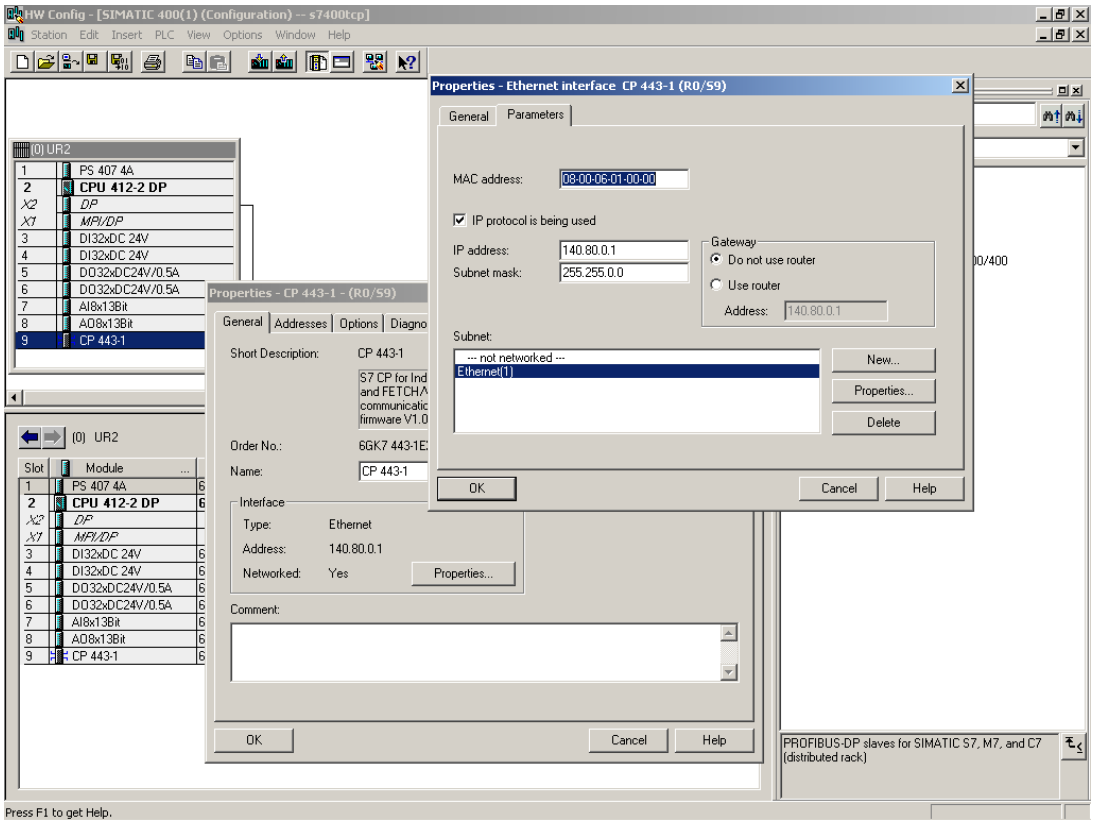


Şekil 3-3: Donanım yapılandırma

ISO on TCP (RFC1006) protokolü kullanılacak ise işlemciye sadece bir MAC adresi vermek yeterlidir, eğer TCP/IP protokolü kullanılacak ise "IP Protocol is being used" seçimi aktif edilip IP adresi verilmelidir. Adresleme ayarları yapılmış bir CP'nin konfigürasyonu Şekil 3-4'te gözükmektedir.

PLC'nin diğer PLC veya dış sistemler ile haberleşebilmesi bu yapılandırmaların tam ve doğru olarak yapılmış olmasına bağlıdır, dolayısı ile Ağ planlamasında dikkat edilmesi gereken noktalar dikkatlice göz önüne alınmalıdır.

İşlemcisinin adresleme ile ilgili yapılandırması bittikten sonra servis ve ara yüz yapılandırmaları için Netpro yazılımı kullanılmaktadır. Netpro yazılımını açmak için Options>>Configure Network menüsü kullanılır.

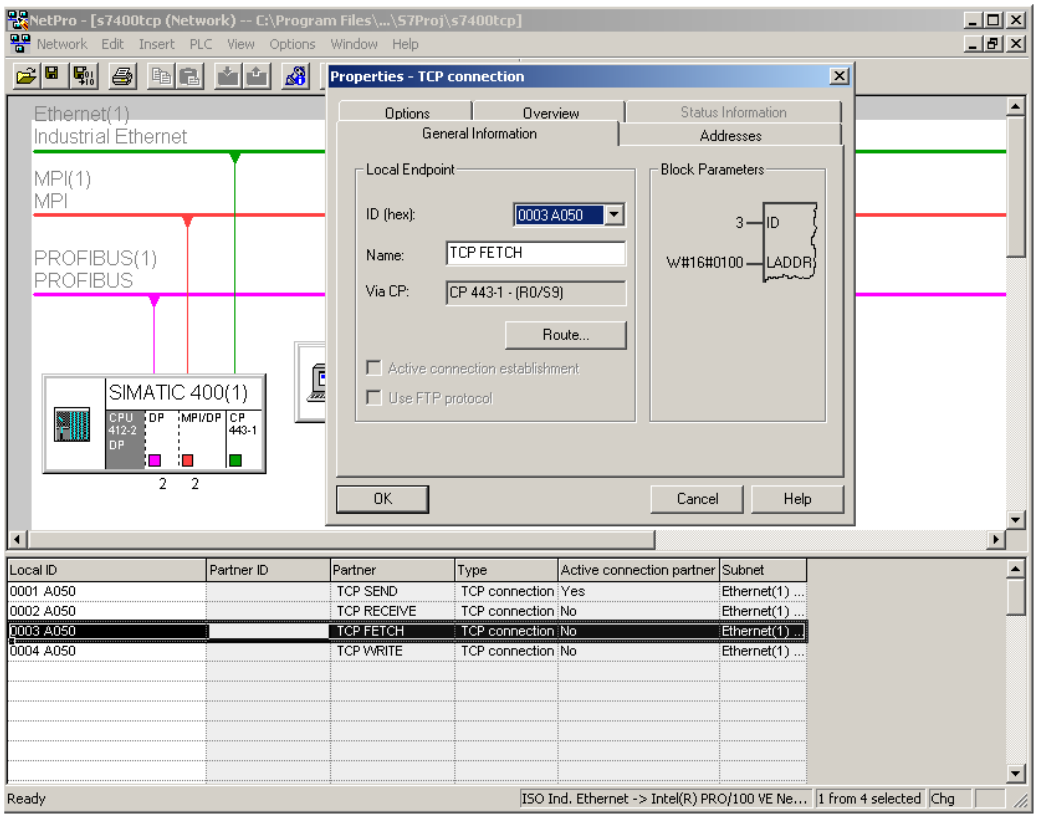


Şekil 3-4: IP Adresi ayarları

3.4.2 TCP-IP Bağlantılarının Konfigüre Edilmesi

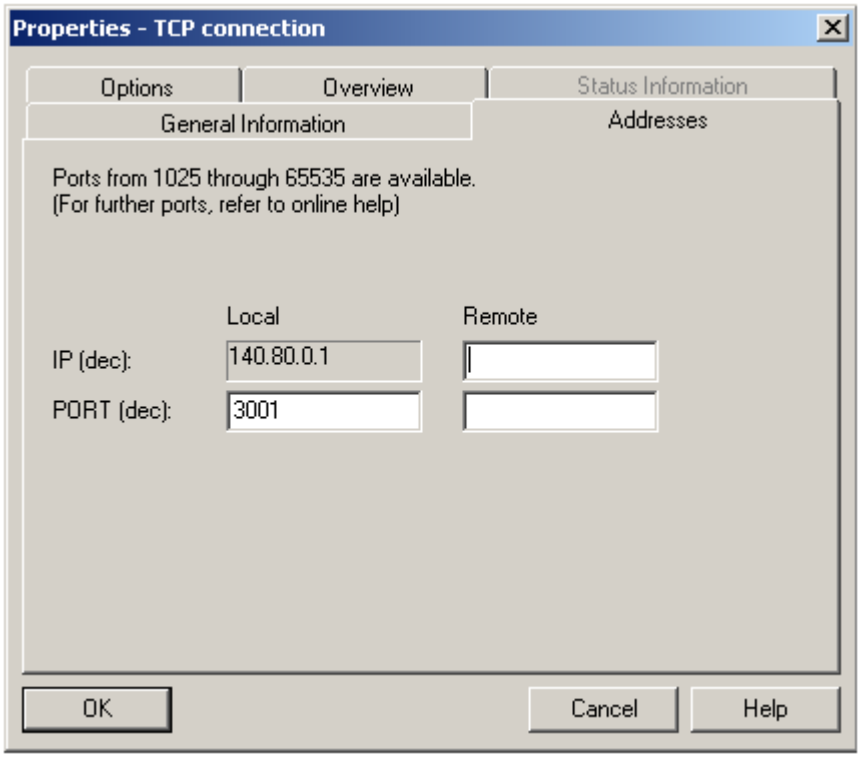
Donanım yapılandırması tamamlandıktan sonra Netpro içinde CPU üzerine tıklayıp Ekle (Insert) Yeni bağlantı (New connection) menüsü yardımıyla yeni bir bağlantı eklenir, Şekil 3-5. Açılan pencerede belirsiz katılımcı (Unspecified partner) seçilir. Yazılan program istemci (Client) olacağı için Aktif bağlantı kurulumu (Active connection establishment) seçeneği işaretlenmemelidir. Yani haberleşme işlemcisi belirtilen bağlantı noktasını (port) dinleyerek bir bağlantı kurulmasını bekleyecektir.

Ayarlar penceresinde adres (Adresses) sekmesi seçilerek bağlantı noktası ayarlarına geçilir, Şekil 3-6. Uzak port ve IP adres alanlarının boş bırakılması herhangi bir bilgisayar üzerinden bu servislerin kullanılmasına olanak sağlayacaktır, aksi halde işlemci sadece burada belirtilen IP adresli istemcinin belirtilen bağlantı noktası



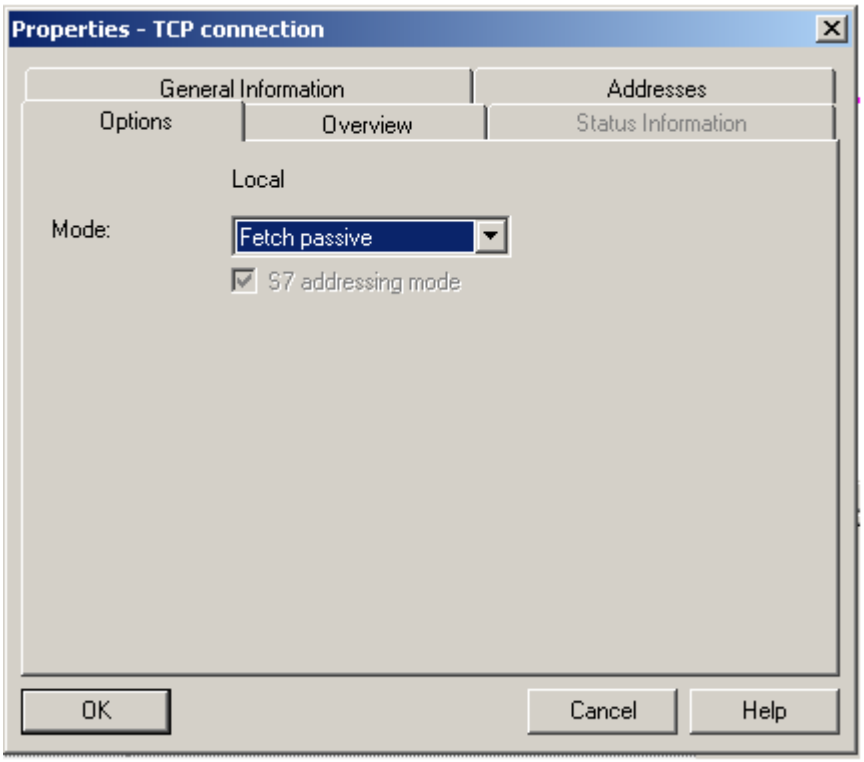
üzerinden bağlantı isteklerine izin verecektir.

Şekil 3-5: Bağlantı oluşturulması



Şekil 3-6: Port ve IP adreslerinin girilmesi

Seçenekler (Options) sekmesine geçilerek Şekil 3-7, bağlantının ne tür bir bağlantı olduğu seçilir, Oku/Yaz servisleri için Pasif oku ve Pasif yaz seçenekleri seçilir. Pasif seçeneğinin nedeni isteğin bilgisayar tarafından yapılacağı ve cevabı işlemci tarafından verileceğiyle ilgilidir. Bu uygulamanın CP443-1 ve diğer Siemens S7 Ethernet haberleşme işlemcileri aktif Oku/Yaz servislerini desteklememektedir. S5 tabanlı bir haberleşme istemcisinde ise Aktif Oku/Yaz servisleri sayesinde PLC içinden diğer bir PLC'nin adresleme alanlarına ulaşmak mümkündür, bu servisleri kullanmak için bir kullanıcı programı yazılması gerekmektedir



Şekil 3-7: Bağlantı seçenekleri

Haberleşme işlemcisinin Oku/Yaz servislerini kullanabilmek için bu yapılandırma yeterlidir, herhangi başka bir kullanıcı programı yazmadan işlemci bu bağlantı noktaları üzerinden yapılan sorgulamalara cevap verecektir. Birden fazla istemci tarafından sorgulama yapmak için bu yapılandırmanın farklı bağlantı noktası adresleri ile istenilen miktarda çoklanması gerekmektedir. CP443-1'de konfigüre edilebilecek toplam maksimum bağlantı sayısının 64 olduğu göz önüne alınmalıdır, daha fazla sayıda bağlantı için CPU üzerindeki haberleşme işlemcisi sayısı arttırılmalıdır.

4. PROGRAMLAMA

4.1 Soket Programlama

4.1.1 Soket programlama nedir

Soketler, bir tür süreçler arası haberleşme (interprocessing) yöntemidir. Soket, soyut bir tanımla haberleşme uç noktalarıdır. Pratik olarak soketler dosyalara benzer. Soketten okumak ile dosyadan okumak arasında hiçbir fark yoktur.

Akla gelebilecek hemen her Internet programı soket program olarak çalışır. Örneğin web sunucuları 80, telnet sunucuları 23 numaralı bağlantı noktası (port) üzerinde çalışan soket programlardır.

Soket programlamaya özel bazı terimler:

İstemci (Client): Hizmet isteyen soket programlara denir. İstediği zaman sunucuya (server) bağlanır, görev verir ve sonuçları alıp bağlantıyı koparır. **Sunucu (Server):** Hizmet veren soket programdır. İstemci herhangi bir anda kendisine bağlanıp, ondan hizmet isteyebileceğinden sürekli çalışmak zorundadır.

4.1.2. Temel Prensipler

Soketler her zaman iki uca sahiptir: Alıcı ve gönderici. Bütün mesajlar ve protokol gereği olan başlıklar nihayetinde fiziksel katmandan, mantıksal 1 ve 0'a karşılık gelen elektriksel sinyaller olarak gönderilir.

Soket program ya istemci, ya da sunucudur. Programlanmaları daha karışık olmakla beraber bazı soket programlar her iki görevi de yapmaktadır. Sunucu program ile istemci program arasında çalışma olarak bazı farklar vardır.Tablo 4-1 her iki tarafta olayların nasıl gittiğini göstermektedir:

Tablo 4-1: Sunucu İstemci Yapısı

İSTEMCİ	SUNUCU
Soket oluştur	Soket oluştur Sunucu adres bilgilerini yerleştir Soketi adres ve isim bilgileri ile ilişkilendir Soketi dinlemeye başla
Bağlantı yap	Bağlantıyı kabul et
Veri gönder	Veri al
Veri al	Veri gönder
Diğer işlemler…	Diğer işlemler…
Soketi kapat	Soketi kapat

4.2. Visual Basic'te Winsock.dll ile Soket Programlama

4.2.1 Winsock.dll Özellikler (Properties)

BytesReceived: Bağlantı sağlandıktan sonra alınmış olan toplam bayt sayısını verir.

LocalHostName: Yerel katılımcının adı. Programın döndüğü bilgisayarın adı.

LocalIP: Yerel katılımcının IP adresi.

LocalPort: Yerel bağlantı noktası numarası. Listen (dinleme) metodu ile dinlenecek bağlantı noktasının numarası.

Protocol: Taşıma katmanında kullanılacak protokol tipi. TCP veya UDP.

RemoteHost: Bağlanılacak katılımcının adı.

RemoteHostIP: Bağlanılacak katılımcının IP adresi.

SocketHandle: Soket durumunu gösterir.

State: Bağlantı durumunu gösterir.

4.2.2 Winsock.dll Metotlar (Methods)

State: Bir bağlantıyı kabul eder.

Bind: Kontrolü seçilen adaptöre geçirir. Birden fazla Ethernet ara yüz kartı var ise.

Close: Mevcut bağlantıyı kapatır.

Connect: İstenilen bağlantıyı kurar.

GetData: Gelen veriyi alıp ara belleği (buffer) temizler.

Listen: Belirtilen bağlantı noktasını (port) dinler.

PeekData: Gelen veriye ara belleği temizlemeden bakar.

SendData: Bağlantı kurulduğundan veriyi gönderir.

4.2.3 Winsock.dll Olaylar (Events)

Close: Bağlanılan katılımcı bağlantıyı kapattığında meydana gelir.

Connect: Bir bağlantı başarıyla kurulduğunda meydana gelir.

ConnectionRequest: Bir istemci bağlantı isteğinde bulununca meydana gelir.

DataArrival: Veri alımında meydana gelir.

Error: Hata yok ise meydana gelir.

SendComplete: Veri gönderildiğinde meydana gelir.

SendProgress: Veri gönderimi sürerken meydana gelir.

4.3 S5 Header Protokolü

ISO, TCP ile ISO ve TCP bağlantılarıyla kullanılabilen Oku/Yaz servisleri bize PLC'nin giriş, çıkış ve hafıza alanlarına ulaşma ve gerekiyor ise değiştirme imkânı sağlar. Bu servisleri sağlayan protokol S5 Header olarak adlandırılmaktadır. Ve 16 bayt sabit bir başlık (Header) içermektedir. Bu protokolden yaralana bilmemiz için PDU [Protokol veri birimi (Protocol data unit)] bilmemiz gerekir, bu bölümde bu yapı incelenmiştir [6].

4.3.1 Yaz Çerçevesi (Write frame) yapısı

Değer belirtilmemiş parametrelerin içeriği 4.3.3 Parametre değerleri adlı bölümde açıklanmıştır.

Tablo 4-2: Yazma istem çerçevesi

0	System ID	'S'
1		'5'
2	Length of header	16d
3	ID OP code	01
4	Length OP code	03
5	OP code	03
6	ORG field	03
7	Length ORG field	08
8	ORG ID	
9	DBNR	
A	Start address	High byte
B		Low byte
C	Length	High byte
D		Low byte
E	Empty field	FFh
F	Length empty field	02
	64 K Veri	

Yazma işlemi yapılacağı zaman gönderilmesi gereken çerçeve yapısı Tablo 4-2'de görülmektedir. Sıfırıncı ve birinci baytlarda gönderilmesi gereken S5 başlığı 'S' ve '5' ASCII karakterlerinin onaltılık düzende karşılığı olarak gönderilmelidir. İlk üç byte başlıkta sabit olup diğer baytlar yazma isteminde bulunulacak alana göre değişe bilmektedir. 16 Byte başlık alanı arkasından 64K veri gönderilebilir [6].

Tablo 4-3: Yazma onay çerçevesi

0	System ID	S'
1		5'
2	Length of header	16d
3	ID OP code	01
4	Length OP code	03
5	OP code	04
6	Ack field	0Fh
7	Length S field	03
8	Error no	No
9	Empty field	FFh
A	Length empty field	07
B	FREE	
C		
D		
E		
F		

Yazma çerçevesine cevap olarak PLC tarafından gönderilen cevap çerçevesi Tablo 4-3'te gösterilmiştir. Cevap çerçevesi 10 byte sabit veri bir baytlık hata numarası olmak üzere toplam 11 veri dolu bayttan oluşur. 16 baytı tamamlayan son 5 byte boştur [6].

4.3.2 Oku Çerçevesi (Fetch frame) yapısı

Okuma için gönderilen istem çerçevesi yazma istem çerçevesine benzemektedir, fakat burada çerçeve toplam 16 byte ile sınırlı olup yazma için gönderilen veri değerleri gönderilmemektedir. Oku çerçevesinin yapısı ayrıntılı olarak Tablo 4-4'te görülmektedir [6].

Tablo 4-4: Okuma istem çerçevesi**:**

0	System ID	‘S’
1		‘5’
2	Length of header	16d
3	ID OP code	01
4	Length OP code	03
5	OP code	05
6	ORG field	03
7	Length ORG field	08
8	ORG ID	
9	DBNR	
A	Start address	High byte
B		Low byte
C	Length	High byte
D		Low byte
E	Empty field	FFh
F	Length empty field	02

Tablo 4-5: Oku onay çerçevesi

0	System ID	**S'**
1		**5'**
2	Length of header	16d
3	ID OP code	01
4	Length OP code	03
5	OP code	06
6	Ack field	0Fh
7	Length S field	3
8	Error no	No
9	Empty field	FFh
A	Length empty field	7
B	**FREE**	
C		
D		
E		
F		

+ Error No = 0 ise 64 K ‘ya kadar veri

Okuma istemine cevap olarak gelen çerçevesi de Tablo 4-5'de ayrıntılı olarak verilmiştir. Bu çerçevede sabit 16 Byte yapıya ek olarak eğer hata numarası (error no) baytı sıfıra eşit ise 64K'ya kadar okumak istediğimiz veri sıralanmış olarak gönderilir [6].

4.3.3 Parametre değerleri

Oku/yaz çerçeve yapılarında boş değerlerin karşılıkları bu bölümde Tablo 4-6 ve Tablo 4-7'te verilmiştir.

Tablo 4-6: Parametre değerleri 1.Kısım

S7 adres alanı	**DB**	**M**	**I**	**Q**
ORG ID	01H Veri Bloğu	02H Hafıza	03H Giriş resmi (PII)	04H Çıkış resmi (PIQ)
DBNR **İzin verilen alan**	Okunacak/yazılacak Veri bloğu'nun Adresi 1..255	Geçersiz	Geçersiz	Geçersiz
Start adres (başlangıç adresi) **İzin verilen alan**	Verinin okunacağı/yazılacağı DW numarası 0..2047	Hafıza bayt no 0..255	Giriş bayt no 0..127	Çıkış bayt no 0..127
Length (uzunluk) **İzin verilen alan**	Word olarak yazılacak/okunacak veri uzunluğu 1..2048	Bayt olarak yazılacak/okunacak veri uzunluğu 1..256	Bayt olarak yazılacak/okunacak veri uzunluğu 1..128	Bayt olarak yazılacak/okunacak veri uzunluğu 1..128

Bu tablolar bize istemde bulunacağımız alanlar için başlık çerçevesini doldurmamıza yardım etmektedir. ORG ID, DBNR, Start Adress ve Length baytları tablonun ilk kolonunu oluşturmaktadır, ilk satırda ise PLC'nin adres alanlarının çeşitleri bulunmaktadır. Her bir PLC adres alanı için o Baytın alması gereken değer ve bu değer değişken olabiliyor ise izin verilen uzunluğu bulunmaktadır. Burada dikkat edilmesi gereken her bir PLC işlemci (CPU) tipinde bu alanların tamamın bulunmaması veya daha fazlasının bulunabilmesi durumudur. Örneğin bir S7 400

CPU'da hafıza (Memory Byte) alanları 1024 byte uzunlukta olmasına rağmen biz protokol izin vermediği için 256 bayttan fazlasını okuyamamaktayız.

Tablo 4-7: Parametre değerleri 2.Kısım

S7 adres alanı	**PI/PQ**	**C**	**T**
ORG ID	05H Çevresel G/Ç	06H Sayıcı	07H Zamanlayıcı
DBNR **İzin verilen alan**	Geçersiz	Geçersiz	Geçersiz
Start adress (başlangıç adresi) **İzin verilen alan**	Verinin okunacağı/yazılacağı G/Ç bayt no 0..127 Sayısal G/Ç 128..255 Analog G/Ç	Sayıcı no 0..255	Zamanlayıcı no 0..255
Length (uzunluk) **İzin verilen alan**	Bayt olarak yazılacak/okunacak veri uzunluğu 1..256	Word olarak yazılacak/okunacak veri uzunluğu 1	Word olarak yazılacak/okunacak veri uzunluğu 1

4.4 Program örneği

4.4.1 Veri yapısı ilk değerler

Öncelikle verinin bütün formlardan ulaşılabilir olması amacı ile module içerisinde yazma ve okuma veri değişkenleri aşağıdaki gibi tanımlanır.

```
Public Send_Fetch_Data(15) As Byte 'Gönderilecek fetch verisi - 16 byte sabit

Public Send_Write_Data() As Byte 'Yazma için gönderilecek başlık + kullanıcı verisi
```

'İlgili form açıldığında aşağıdaki gibi bu alanlara ilk değerler atılır.

Okuma verisi ilk değerler

```
Send_Fetch_Data(0) = 83   ' S
```

```
Send_Fetch_Data(1) = 53   ' 5

Send_Fetch_Data(2) = 16   ' Fetch istek bloğu uzunluğu

Send_Fetch_Data(3) = 1    ' Opcode ID

Send_Fetch_Data(4) = 3    ' Opcode uzunluğu

Send_Fetch_Data(5) = 5    ' Opcode = fetch

Send_Fetch_Data(6) = 3    ' Org Field

Send_Fetch_Data(7) = 8    ' Org Field uzunluğu

Send_Fetch_Data(8) = 2    ' Org ID (2=Memory Byte)

Send_Fetch_Data(9) = 0    ' DB numarası

Send_Fetch_Data(10) = 0   ' Başlangıç Adresi High byte

Send_Fetch_Data(11) = 0   ' Başlangıç Adresi Low Byte

Send_Fetch_Data(12) = 0   ' Fetch edilecek veri sayısı (high Byte)

Send_Fetch_Data(13) = 1   ' Fetch edilecek veri sayısı (low byte)

Send_Fetch_Data(14) = 255 ' Boş alan

Send_Fetch_Data(15) = 2   ' Boş alan uzunluğu


' Yazma istek verisi (başlık) ilk değerler

ReDim Send_Write_Data(16)

Send_Write_Data(0) = 83   ' S

Send_Write_Data(1) = 53   ' 5
```

```
Send_Write_Data(2) = 16   ' Fetch istek bloğu uzunluğu

Send_Write_Data(3) = 1    ' Opcode ID

Send_Write_Data(4) = 3    ' Opcode uzunluğu

Send_Write_Data(5) = 3    ' Opcode = fetch

Send_Write_Data(6) = 3    ' Org Field

Send_Write_Data(7) = 8    ' Org Field uzunluğu

Send_Write_Data(8) = 2    ' Org ID (2 = Memory Byte)

Send_Write_Data(9) = 0    ' DB numarası (kullanılmıyor ise 0)

Send_Write_Data(10) = 0   ' Başlangıç Adresi High byte

Send_Write_Data(11) = 0   ' Başlangıç Adresi Low Byte

Send_Write_Data(12) = 0   ' Fetch edilecek veri sayısı (high Byte)

Send_Write_Data(13) = 1   ' Fetch edilecek veri sayısı (low byte)

Send_Write_Data(14) = 255 ' Boş alan

Send_Write_Data(15) = 2   ' Boş alan uzunluğu

Send_Write_Data(16) = 0   ' İlk kullanıcı verisi baytı program içinde doldurulacak
```

4.4.2 Okuma istemi

Okuma işlemi bir buton aracılığı ile bir tek sefer yapılabileceği gibi sürekli oku seçeneği işaretlenerek girilen süre aralığında yürekli yenilenebilir. Okuma işlemi sırasında öncelikle yukarıda oluşturduğumuz istem veri bloğunun PLC'ye gönderilmesi gerekmektedir.Program öncelikle bağlantının kurulup kurulamadığına bakar eğer bağlantı kurulmuş ise veri bloğunu PLC'ye gönderir. Sürekli okuma durumunda istek veri bloğu bir zamanlayıcı yardımıyla sürekli olarak iletilir.

Zamanlayıcının süresini ayarlayarak okuma sıklığını değiştirmek mümkündür.

```
Private Sub fetch_button_Click()
 If (TCP_FETCH.State = 7) Then
TCP_FETCH.SendData Send_Fetch_Data
Else
Form1.status.Caption = "Bağlanılamadı"
End If
End Sub
Private Sub fetch_cycle_Timer()
If Check1.Value = 1 Then   'Sürekli okuma modu
If (TCP_FETCH.State = 7) Then
TCP_FETCH.SendData Send_Fetch_Data
Else
  If TCP_FETCH.State = sckConnected Then
Form1.status.Caption = "Bağlanılamadı"
End If
End If
End If
End Sub
```

4.4.3 Okuma veri gelişi

Veri alımında bu veri my_fetch_data değişkenine aktarılır ve bu veri baytlarına ayrılıp ekrana yazılır.

```
Private Sub TCP_FETCH_DataArrival(ByVal bytesTotal As Long)

TCP_FETCH.GetData my_fetch_data, vbArray + vbByte

If my_fetch_data(8) = 0 Then

'Form1.Text1.Text = my_fetch_data(16)

Form1.Text1.Text = ""

For i = 16 To (bytesTotal - 1)

Form1.Text1.Text = Form1.Text1.Text & Str$(my_fetch_data(i))

Next i

Else

Form1.Text1.Text = "hata"

End If

End Sub
```

4.4.4 Yazma İstemi

Veri okunacağı zaman daha evvelden hazır bekleyen istek başlığı eğer bağlantıda kurulmuş ise işlemciye gönderilir.

```
Private Sub TCP_Veri_Yaz()

 If (TCP_WRITE.State = 7) Then

 TCP_WRITE.SendData                    Send_Write_Data
```

End If

End Sub

4.4.5 Yazma Cevabı

Aşağıdaki uzak işlemci hata kodları CP1430 kullanım kılavuzundan alınmıştır sayfa C-12 [7]:

1: Yanlış veri tipi istemi

2: İstenen veri PLC'de mevcut değil

3: Gönderilen veri yazılacak alan için büyük

4: PLC'de zaman aşımı hatası (bozuk hafıza?)

5: PLC'de durum kelimesi (status word) işlenemiyor

6: Geçersiz alan tipi istemi

7: İşlemci ara belleği taşma (çok fazla veri gönderimi)

8: Yeteri kadar veri gönderilmedi

9: PLC işleme hatası veriyor

10: Bağlantı yok

11: Veri çok uzun ya da sistem hatası

12: RB başlığı yanlış

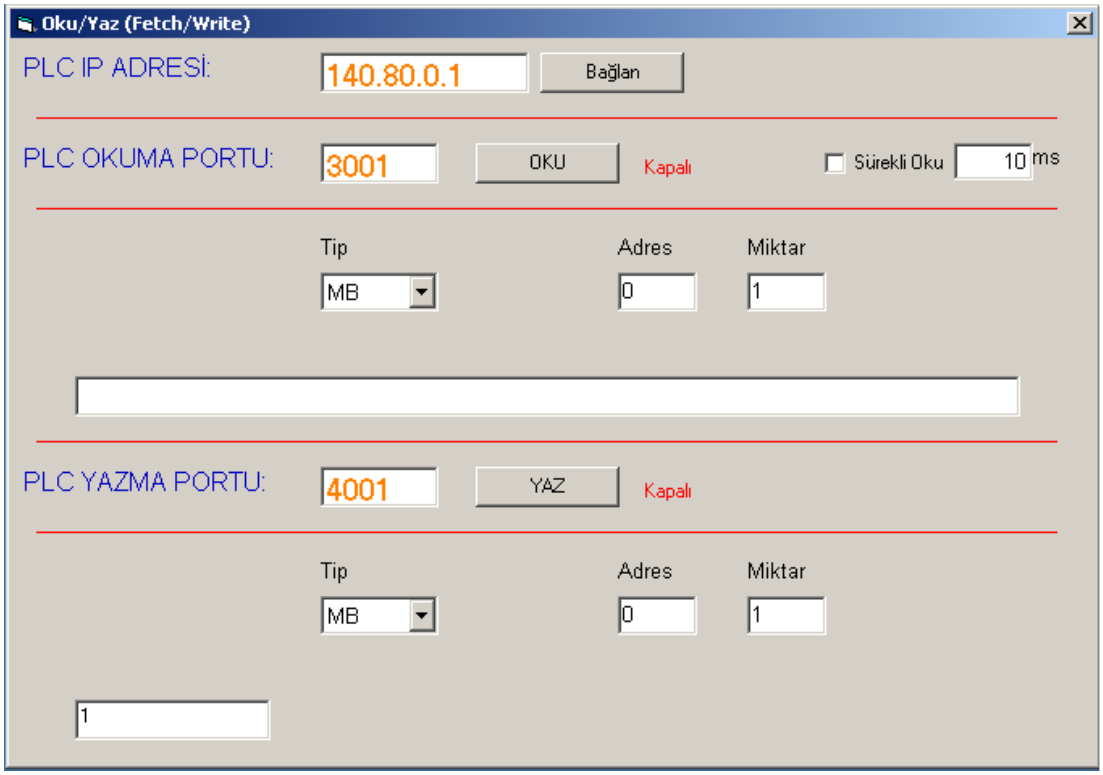
13: Yeniden başlama (reset) aktif, bekleyin

14: Bu pasif bir iş

15: İşlemcide tanımlanmamış iş, Netpro'yu kullanın

4.4.6 Kullanıcı Ara yüzü

Örnek (Demo) programın kullanıcı ara yüzü Şekil 4-1'de görülmektedir.



Şekil 4-1: Program Kullanıcı Ara yüzü

Program çalıştırıldıktan sonra PLC IP ADRESİ, PLC OKUMA PORTU ve PLC YAZMA PORTU alanları PLC'de konfigüre edildiği şekilde ilgili alanlara girilmelidir. Program PC'nin okuma ve yazma bağlantı noktası olarak 0 adresini atamaktadır, winsock.dll port adresi olarak 0 girildiğinde o an müsait olan boş bir bağlantı noktası bu bağlantı için kullanmaktadır. PLC yapılandırmasında karşı tarafın port ve IP adresleri önceden bildirilmiş ise programın kodunda ilgili satırlar değiştirilmelidir. Sabit bir port adresi üzerinden peş peşe bağlantı kurulup koparıldığında, bir daha ki bağlantının sağlıklı olarak kurulabilmesi için bir süre beklemek gerekmektedir, bunun winsock ile ilgili bir problem olduğunu düşünmekteyim. Port seçim işlemi winsock servisine bırakıldığında bu problem ortadan kalkmaktadır.

Adres giriş işlemleri tamamlandıktan sonra BAĞLAN butonuna basarak PLC ile bağlantı kurulur. Eğer bağlantı sağlıklı bir şekilde kurulabilmiş ise durum pencerelerinde BAĞLANDI eğer bir bağlantı problemi yaşandı ise HATA mesajlarını görürüz. Bağlantı kurulduktan sonra OKU/YAZ servisleri kullanıma hazır hale gelmektedir.

Oku servisini kullanmak için alan tipi, adresi ve okunacak veri miktarı girilmektedir. Verinin veya veri grubunun okunması için OKU butonuna basılmalıdır. Eğer veri sürekli okunacak ise SÜREKLİ OKU seçeneği işaretlenmelidir. Bu seçenek işaretlendiği zaman girilmiş süre sonunda veri değeri döngüsel olarak yenilenir.

Yaz servisini kullanmak için alan tipi, adresi ve yazılacak veri miktarı belirlendikten sonra YAZ butonuna basılmalıdır. Veri giriş alanına girilen değer PLC'ye gönderilir, eğer yazma işlemi sırasında bir hata oluşmuş ise hata açıklaması durum penceresinde gözükecektir.

5. SONUÇ

Endüstriyel veri iletişiminde Ethernet ülkemizde yeni yeni kullanılmaya başlanmıştır, dolayısıyla PLC kullanıcıları için yeni ve çok ayrıntılı bilinmeyen bir konudur. Bu haberleşme şekli PLC tarafında ek donanım maliyetleri getirmesinin yanında PLC sistemlerinin bilgisayarların açık standartları ile kavuşmasını sağlamaktadır. Üretici kaynaklı protokoller yerine bu açık protokollerin kullanılması HMI, SCADA veya planlama yazılımlarının saha cihazlarına entegrasyonunu kolaylaştırmaktadır. Günümüzde Ethernet haberleşme ara yüzü endüstriyel kontrol cihazlarına sıklıkla entegre edilmeye başlamıştır ve bu grafik artarak devam edecektir.

Yazılan örnek program bir PLC sistemine basit bir okuma ve yazma ara yüzü oluşturmuştur.Bu çalışmada oluşturulan haberleşme programının üzerine görsel sayfalar eklenerek kolayca bir insan makine ara yüzü oluşturulabilir. Daha büyük çaplı sistemler için Visual Basic programlama dili zayıf kalacağı için bu tezde anlatılan haberleşme yapısı başka bir programlama diline aktarılarak örneğin Delphi veya Visual C daha büyük çapta izleme ve kontrol sistemleri de kolaylıkla kurulabilir.

KAYNAKLAR

[1] **Weigmann, J. ve Kilian, G.,** 2000. Decentralization with PROFIBUS-DP : architecture and fundamentals, configuration and use with SIMATIC S7, Publicis MCD Verlag, Munich

[2] **Dr. Çölkesen, R., Prof.Dr. Örencik, B.,** 2000. Bilgisayar Haberleşmesi ve Ağ teknolojileri, Papatya Yayınevi, İstanbul

[3] **Held, G.,** 2003. Ethernet networks : design, implementation, operation, management, Wiley, Newyork

[4] **Berger, H.,** 2000. Automating with STEP 7 in STL and SCL : programmable controllers SIMATIC S7-300/400, Publicis MCD Verlag, Munich

[5] **Berger, H.,** 2000. Automating with simatic : integrated automation with simatic S7-300/400 controllers, software, programming, data communication, operator control and process monitoring, Publicis MCD Verlag, Munich

[6] SIMATIC NET NCM for Industrial Ethernet Manual, 6GK7080–1AA03–8BA0, 11/2000, C79000–G8976–C129 Release 04, Siemens

[7] SINEC CP1430 TF with COM1430 TF Volume 2 of 2, 6GK1970-1TA43-0AA1, C79000-G8976-C056 Release 01, Siemens

EKLER

Ek 1: CD/ Step 7 PLC Projesi ve Visual Basic örnek program kodları.

ÖZGEÇMİŞ

Timur Bir 1976 yılında İstanbul'da doğdu.Ortaokul ve lise öğrenimini Özel Saint Benoit Fransız lisesinde tamamladı.1996 yılında İTÜ Elektrik Elektronik Fak. Elektrik Müh. Bölümüne başladı. 2002 yılında mühendislik diplomasını alıp aynı fakülte ve bölümde yüksek lisans öğrenimine başladı.

Nisan 2002 ile Kasım 2005 arasında Robosistem Elektronik şirketinde Teknik Bölüm Yöneticisi olarak çalıştı.