T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PROFINET IO AĞ ÇÖZÜMLEYİCİSİ VE AĞ OYNATICISI

YÜKSEK LİSANS TEZİ

Bilg. Müh. Orçun SEDEN

Enstitü Anabilim Dalı	**:**	**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Yrd.Doç. Dr. İbrahim Özçelik**

Mayıs 2010

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PROFINET IO AĞ ÇÖZÜMLEYİCİSİ VE AĞ OYNATICISI

YÜKSEK LİSANS TEZİ

Bilg. Müh. Orçun SEDEN

Enstitü Anabilim Dalı	:	BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Bu tez 31/05/2010 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Prof. Dr. Ümit KOCABIÇAK Jüri Başkanı	Prof Dr. İsmail ERTÜRK Üye	Yrd. Doç. Dr. İbrahim ÖZÇELİK Üye

TEŞEKKÜR

Bu tez çalışmasının tamamlanmasında değerli katkılarını esirgemeyen danışmanın Yrd. Doç Dr. İbrahim ÖZÇELİK'e, çalışmalarım sırasında bana sürekli ve sabırla destek olan aileme teşekkürü bir borç bilirim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

ARP : Address Resolution Protocol
ASCII : American Standard Code for Information Interchange
CBA : Component Based Automation
CSMA/CD : Carrier Sense Multiple Access With Collision Detection
DCP : Dynamic Configuration Protocol
DSA : Digital Signature Algorithm
EBCDIC : Extended Binary Coded Decimal Interchange Code
FTP : File Transfer Protocol
GPL : General Public License
ICMP : Internet Control Message Protocol
IEEE : Institute of Electrical and Electronics Engineers
IP : Internet Protocol
IRT : Isochronous Real Time
ISO : International Organization for Standardization
LAN : Local Area Network
LLDP : Link Layer Discovery Protocol
MAC : Media Access Control
NRT : Non Real Time
OS : Operating System
OSI : Open System Interconnection
PCAP : Packet Capture
PTCP : Precision Time Control Protocol
RPC : Remote Procedure Call
RSA : Rivest , Shamir and Adleman
RT : Real Time
SNMP : Simple Network Management Protocol
SSH : Secure Shell
SSL : Secure Sockets Layer

TCP	: Transmission Control Protocol
TLS	: Transport Layer Security
UDP	: User Datagram Protocol
VLAN	: Virtual Local Area Network
VPN	: Virtual Private Network
WAN	: Wide Area Network

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar Kelimeler: Profinet IO, Paket Koklama, Profinet Protokol Çözümleme, Wireshark, jNetPcap

Otomasyon sistemlerinde gerçek zamanlı iletişimi sağlamak için üreticiler kendi özel protokollerini geliştirmişlerdir. Siemens'in geliştirdiği Profinet IO, Ethernet tabanlı gerçek zamanlı bir iletişim sistemidir. Profinet ile standart Ethernet prokolleri kullanılabildiği gibi, saha seviyesinde gerçek zamanlı iletişimi sağlayabilmek için Ethernet'in veri bağı katmanı ve ağ katmanının özelleştirilmesi ile oluşturulan Profinet IO protokolleri de kullanılmaktadır. Bu protokollerden IRT protokolü ile 1 ms, RT protokolü ile 10 ms mertebesinde veri iletimi yapabilmek mümkündür. Ayrıca gerçek zaman ihtiyacı olmayan veri iletimi için NRT protokolü kullanılmaktadır. Profinet IO sisteminde, sistemdeki cihazlar Profinet'in DCP protokolü ile birbirlerini tanırlar ve bu protokolün incelenmesi ile de sistemdeki cihazlar ve özellikleri elde edilebilir.

Bu tez çalışmasında Profinet IO sistemi için ağ çözümleyici ve ağ oynatıcı uygulamaları geliştirilmiştir. PROFINET IO ağ çözümleyicisi, bir Profinet ağı üzerindeki bütün Profinet çerçevelerinin yakalanmasını, incelenmesini ve sınıflandırılmasını sağlamaktadır. Profinet IO ağ çözümleyici ile yakalanan çerçeveler sonucunda ağ topolojisinin görsel olarak sergilenmesi de gerçekleştirilmiştir. PROFINET IO ağ oynatıcısı ise Profinet ağ çözümleme algoritmasının tersten çalıştırılması sonucunda oluşturulan çerçevelerin ağa gönderilmesi ve ağ üzerinde sonuçlarının görülmesi için geliştirilmiştir. Profinet ağ oynatıcısı ile daha önceden kaydedilmiş paketlerin veya kullanıcı arayüzü üzerinden oluşturulacak paketlerin sistemin tepkisini ölçmek amaçlı olarak ağa gönderilmesi de sağlanabilir.

PROFINET IO PROTOCOL ANALYZER AND PROTOCOL PLAYER

SUMMARY

Keywords: Profinet IO, Packet Sniffing, Profinet Protocol Analyzer, Wireshark, jNetPcap

At Automation systems, providers develope their own protocols to support real-time communication.Profinet IO is Ethernet based real-time communication system which is developed by Siemens. In addition to standart Ethernet protocols, Profinet IO protocols which is enhanced datalink layer and network layer of the Ethernet, can be used to support real-time communication. Among these protocols, IRT protocol support 1ms, RT protocol support 10 ms. data communication. Addition to these protocols, NRT protocol should be used to non real time operations. At Profinet IO system, devices know each other through DCP protocol and analyzing that protocol, devices and their features can be found.

In that thesis, network analyzer and network player applications are developed for Profinet IO system. PROFINET IO network analyzer supplies to capture all network packets, analyze them and classify them. As a result of packet capturing with Profinet IO network analyzer, topology of the network is displayed in a graphical user interface. PROFINET IO network player supplies to send network packets and user can see effects of these packets through executing Profinet network analyzer algorithm reverse. Previously captured and saved packets, or new packets that is created by the user through a GUI can be send to the network to see its effects.

BÖLÜM 1. GİRİŞ

Günümüzde kullanılan haberleşme ağlarında en yaygın olarak kullanılan haberleşme standartı Ethernet'tir. Ethernet en küçük ev ağlarından en büyük kampüs ağlarına ve tüm Dünya'da kullanılan Internet'e kadar haberleşmenin temel standartı haline gelmiştir. Bu yaygın kullanım Ethernet'in Endüstriyel haberleşme alanında kullanılmasına da yol açmıştır. Ancak Endüstriyel haberleşme için olan gereksinimler herhangi bir ofis, ev, kampüs ağında ihtiyaç duyulandan çok daha fazladır, örneğin, gerçek zaman yeteneği, dağıtılmış saha cihazlarının entegrasyonu vs. Bu gereksinimler açık ve üreticiden bağımsız endüstriyel Ethernet standartı Profinet tarafından, ofis dünyasından saha seviyesine kadar kesintisiz haberleşme sağlanacak şekilde karşılanır. Profinet, TCP/IP gibi IT-Standartlarından faydalanıp, otomasyon görevleri için gerçek zaman veri haberleşmesini sağlayarak ,yüksek performanslı hareket kontrolünü gerçekleştirir.

Bu tez çalışması ile öncelikle Ethernet standartı olarak Profinet kullanan bir ağ ve bu ağdaki cihazlar tarafından üretilen ağ trafiğinin paket koklama (ağ çözümleme) yöntemleri kullanılarak incelenmesi amaçlanmıştır. Günümüzde çok çeşitli paket koklama araçları kullanılmaktadır, bunlardan en yaygın olarak kullanılanı Wireshark (eski adı Ethereal) paket koklama aracıdır. Wireshark ile belirli bir detay seviyesine kadar Profinet çerçevelerinin çözümünün yapılması mümkündür ancak asıl verinin taşındığı bölüm "Undecoded Data (Çözümlenememiş Veri)" olarak gösterilmektedir. Bu tez çalışması ile Wireshark tarafından çözümlenemeyen bu bölümler de sınıflandırılma amaçlı çözümlenebilmektedir. Böyle bir ağ çözümleyici yapılması, özellikle Ethernet iç yapısının öğrenilmesi açısından oldukça öğreticidir. Profinet ağ çözümleyici uygulaması ile sistemin genel işleyişi ve oluşan problemlerin nedenleri üzerinde bilgi sahibi olmak mümkündür.

Profinet protokollerinin ayrıntısı Siemens tarafından gizli kapalı tutulmaktadır. Profinet çerçevelerinin incelenmesi ile sistemin bazı fonksiyonlarının paketler üzerindeki verilerde ne gibi etki yaptığı gözlenebilmektedir. Bu sayede bu tez kapsamında geliştirilen Profinet Ağ Oynatıcı (Player) paket oluşturma aracı temel alınarak ve daha da geliştirilerek sistemdeki bazı fonksiyonlar Siemens'in kendi yazılımları olmadan dışarıdan gerçeklenebilir.

Bu tez çalışması 5 ayrı bölümden oluşmaktadır. 2. Bölümde paket koklama kavramı ayrıntılı olarak ele alınmaktadır. Bir paket koklama uygulaması için gerekli donanım ve yazılım, gerekli ağ topolojisi hakkında bilgi verilmektedir Paket koklama uygulamalarının günümüzdeki genel kullanım şekilleri ve kullanılan örnek programlar listelenmektedir. Bu bölümde son olarak paket koklayıcıya karşı kullanılan karşı atak yöntemleri ve paket koklayıcı olarak çalışan programlardan korunma yöntemleri üzerinde durulmaktadır.

Tezin 3. Bölümünde, günümüzde en yaygın olarak kullanılan Wireshark uygulaması hakkında bilgi verilmektedir. Wireshark'ın genel yapısı şekilsel olarak gösterilmekte ve programın genel özellikleri listelenmektedir. Ağ üzerinde dinleme yapabilmek için Wireshark'ın doğru ve yanlış yerleşimi örnek şekiller ile gösterilmektedir. Wireshark kullanıcı grafik arayüzdeki bazı temel fonksiyonlar şekiller ile gösterilmekte ve son olarak programın kullanım ve uygulama alanları hakkında bilgi verilmektedir.

Tezin 4. Bölümü Profinet hakkında genel bilgiler ve Profinet protokollleri hakkında ayrıntılı bilgiler içermektedir. Gerçek zamanlı iletişim kavramı incelenmekte ve gerçek zamanlılık için gereken özellikler açıklanmaktadır. Anahtarmalı Ethernet ve VLAN kavramları üzerinde durulmakta örnek çerçeve yapısı verilmektedir. Profinet katmanlı mimari yapısı şekilsel olarak gösterilmekte ve Profinet protokollerinin bu katmanlı yapıdaki görevleri hakkında bilgi verilmektedir. Bu bölümde son olarak Profinetin iletişim mimarisi ve profilleri anlatılmakta ve Profinet IO içinde kullanılan 5 adet protokolün ayrıntılı çerçeve yapıları ve Wireshark ile yakalanmış örnekleri şekilsel olarak gösterilmektedir.

Tezin 5.Bölümü, tez kapsamında gerçekleştirilen uygulamaların kullandığı geliştirme ortamları ve kullanılan kütüphaneler hakkında bilgiler vermektedir. Profinet Ağ Çözümleyici uygulaması ve Profinet Ağ Çözümleme Algoritması kullanılarak gerçekleştirilen paket koklama ve Profinet DCP protokolünün çözümlenmesi ile oluşturulan Topoloji gösterme uygulamarı algoritmik ve ekran görüntüleri kullanılarak şekilsel olarak anlatılmaktadır. Uygulamaların örnek ekran görüntüleri şekil olarak gösterilmekte ve temel fonksiyonlar Java kodları olarak satır satır anlatılmaktadır.

Tezin 6.Bölümünde tez kapsamında gerçekleştirilen Profinet Ağ Oynatıcısı (Player) uygulaması hakkında bilgiler verilmektedir. Bu uygulama ile Profinet ağına, kullanıcı arayüzü tarafından oluşturulan veya daha önceden bir dosyaya kaydedilmiş halde bulunan Profinet çerçevelerinin gönderilmesi mümkündür. Ağ oynatıcı uygulamasının her iki türlü çerçeve oluşturma ve gönderme şekli, ekran görüntüleri ile ayrıntılı olarak gösterilmektedir.

BÖLÜM 2. PAKET KOKLAMA (PACKET SNIFFING)

Paket koklama (veya ağ yorumlayıcısı) ağ üzerindeki noktalar arasında iletilen bütün paketlerin toplanması işlemidir. Paket koklama tekniği ile ağ üzerinde başka kullanıcılara ait bütün veriler ele geçirebilir. Paket koklama araçları ağ üzerinde yönetici aracı veya kötü niyetli amaçlar için kullanılabilir, kullanım şekli tamamen kullanıcının niyetine ve insiyatifine bağlıdır. Ağ yöneticileri paket koklama programlarını ağ trafiğini inceleme ve doğrulama amaçlı olarak kullanabilirler [1].

Paket koklama araçları özel bir donanım ve bu donanım üzerinde çalışan yazılım veya sadece bilgisayara kurulabilen uygulama programlarıdır. OSI ağ modelinin veri bağı, ağ katmanı ve ulaşım katmanı seviyesinde çalışabilirler. Ancak genel olarak ağ katmanından alınan paketlerin veri kısımlarının incelenmesi yöntemiyle çalışırlar [1]. Paket koklayıcı programlarını birbirinden ayıran en temel farklar, destekledikleri protokol sayısı, kullanıcı grafik arayüzleri ve kullanıcıya sağladıkları istatistiksel verilerin çeşitliliğidir [2].

Paket koklayıcılar donanım ve yazılımın kombinasyonu şeklindedir. Bütün ürünlerde farklılıklar olmasına rağmen, genel olarak bir paket koklayıcı veya ağ yorumlayıcısı (network analyzer) 5 parçadan oluşur [2]:

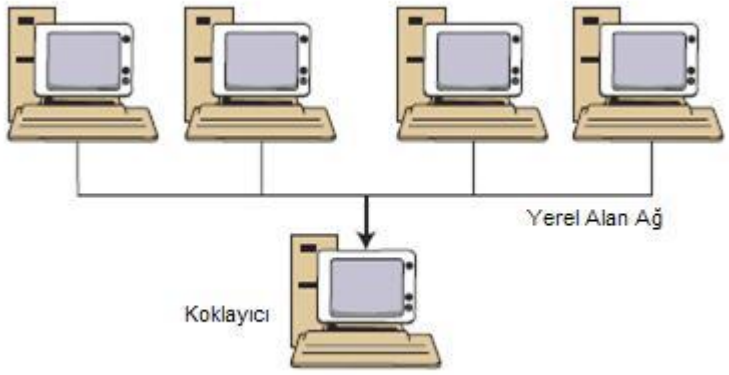
- **Donanım:** Paket koklayıcıların büyük bir çoğunluğu işletim sistemleri üzerinde ve bir ağ arayüz kartı ile çalışan yazılımlardan oluşmasına rağmen, donanımsal tabanlı koklayıcı uygulamaları da vardır. Donanım tabanlı koklayıcılar, temel olarak donanımsal hataları, voltaj problemlerini ve kablo problemlerini bulmak için kullanılır.

- **Ağ Arayüzü Kartı ve Sürücüsü:** Bu kısım ağ kablosundaki ağ trafiği içindeki paketlerin yakalanmasından sorumludur. Filtreleme yöntemleri sayesinde istenen veri paketleri tampon bölgeye alınabilir. Paket koklayıcılar için en temel parça ağ arayüz kartı ve sürücüsüdür. Bu parça olmadan paket toplama işlemi yapılamaz.

– **Tampon Bölge:** Yakalanan verilerin depolandığı kısımdır. Tampon bölge dolana kadar veya yeni bir veri gelene kadar eski veriler depolanır. Tamponlar disk veya hafıza tabanlı olabilir.

– **Gerçek Zamanlı Yorumlama/Analiz:** Veriler kablodan okunduğu şekliyle yorumlanır. Bazı koklayıcılar bu kısmı ağ performansını ölçmek veya nüfuz tespit sistemlerinde bir saldırı olup olmadığını incelemek için kullanırlar.

– **Çözümleme:** Bu kısım ağ trafiğinden toplanan verilerin incelenmesi esasına dayanır. Bu inceleme koklayıcı içinde tanımlı olan protokollere göre yapılır. Her protokole özgü bir çözümleme algoritması vardır. Yeni prokollere ait algoritmalar eklenerek koklayıcı programı genişletilebilir.



Şekil 2.1. Yerel Ağ (LAN)

Şekil 2.1 Ethernet tabanlı yerel ağ (LAN) üzerinde çalışan bir paket koklama topolojisini göstermektedir. Paket koklayıcı kendi ağ arayüz kartına gelen paketleri dinler. Paket koklayıcılar sadece yerel ağ ile sınırlı değildirler. Benzer işlevde Geniş Alan Ağ (WAN) üzerinde de çalışan paket koklayıcılar mevcuttur. Eğer paket koklayıcı geniş alan ağ üzerinde birbirine bağlı iki makinenin bağlantı yolu üzerindeyse, paket koklayıcı bu iki makine arasındaki trafik akışını dinleyebilir [1].

Ağ üzerindeki koklayıcılar, telefonla konuşan iki kişinin konuşmasını gizlice dinlemeye benzetilebilir. O anda ağ üzerinde iletilen başka birine ait bilgi habersizce

toplanır. Bu bilgiler arasında şifre veya benzeri gizlilik düzeyi son derece yüksek veriler de olabilir [1].

Koklayıcı programlar iki ana kategoriye ayrılabilir. Ticari koklayıcı programları, genel olarak ağ yöneticileri tarafından ağın bakımı ve performansının incelenmesi gibi işlemlerde kullanılır. Yeraltı paket koklayıcı programları ise kişisel çıkarlar için şifre ve bunun gibi gizlilik düzeyi yüksek bilgilerin ele geçirilmesi amacıyla kullanılır. Paket koklayıcı programların genel kullanım şekillerinden bazıları [1, 2]:

- Ağ trafiğinin loglanması,
- Ağ üzerindeki iletişim problemlerinin çözümü: (Ör: A bilgisayarı ile B bilgisayarının haberleşememesinin nedenleri vs.),
- Ağ performansının incelenmesi. Bu yöntemle ağ üzerinde bulunan darboğazlar veya ağın bir parçasında tıkanıklıktan dolayı oluşan veri kayıpları bulunabilir,
- Kullanıcıların, kullanıcı adı ve şifreleri elde edilebilir,
- Paketler içindeki ikili sistemdeki veriler okunabilir halde getirilebilir,
- Hukuksal konularda kanıt olması için ağ trafiği kayıt altına alınabilir,
- Ağ kartlarındaki problemler tespit edilebilir,
- Virus veya DoS ataklarının kaynakları tespit edilebilir,
- Casus yazılımlar tespit edilebilir,
- Geliştirme sürecindeki ağ uygulamalarının debug edilmesi için kullanılabilir,
- Eğitim amaçlı olarak ağ protokollerinin incelenmesinde kullanılabilir,
- Ağa nüfuz eden saldırganlar tespit edilebilir.

Bir ağda koklamanın olabilmesi için öncelikle ilgili programın ağa erişiminin sağlanması gereklidir. Bunun için ya ağın bir bölümüne bağlanılması veya iletişim

ağındaki bir noktaya ağ kablosunun takılması gereklidir. Eğer kötü amaçlı bir koklama yapılacaksa ve saldırgan fiziksel olarak bağlı değilse bile çeşitli yöntemlerle ağ üzerinde koklama işlemi gerçekleştirilebilir [2]. Bunlardan bazıları aşağıda verilmiştir:

– Sistemdeki bir bilgisayara erişim sağlanıp uzaktan kontrol edilebilen koklayıcı program bu bilgisayara kurulabilir,

– İletişim erişim noktalarına veya servis sağlayıcılara gizlice girilip koklayıcı program kurulabilir,

– Servis sağlayıcı üzerine halihazırda içinde koklayıcı program çalışan yeni bir sistem kurulabilir,

– Sosyal mühendislik yöntemleri kullanılarak servis sağlayıcıya fiziksel olarak erişilip koklayacı program kurulabilir,

– İletişim hattınının kendisi veya kopyası koklayıcı programın çalıştığı ağa yönlendirilebilir.

2.1. Paket Koklama Programları

En yaygın olarak kullanılan programlardan bazıları [2]:

Wireshark: Wireshark şu anda da en yaygın olarak kullanılan ve açık olarak dağıtılan paket koklama programıdır. Oldukça kullanışlı bir kullanıcı grafik arayüzüne, 400den fazla protokolü destekleyen bir altyapıya sahiptir ve aktif olarak halen geliştirilmesi de sürmektedir. UNIX tabanlı sistemler ile Mac OSX ve Windows işletim sistemlerinde çalışmaktadır.

TcpDump: En eski ve en yaygın olarak kullanılan paket koklama programıdır. UNIX tabanlı sistemlerde komut satırından çalıştırılan ve hala geliştirilen bir programdır. Mac OS X işletim sisteminde de çalışan versiyonu mevcuttur.

WinDump: Tcpdump programının Windows işletim sisteminde çalışan versiyonudur. WinPcap kütüphanesini kullanır.

Network General Sniffer: En ünlü ticari koklama araçlarından biridir.

Windows 2000 ve 2003 Server Network Monitor: Windows 2000 ve 2003 sunucularında işletim sistemi ile birlikte gelen ağ analizi programlarıdır.

EtherPeek: Ticari ağ analizi programıdır, Windows ve Mac altında çalıştırılabilir.

Snoop: Sun Solaris OS üzerinde komut satırından çalışan koklama programıdır.

Snort: Ağ koklama yöntemini kullanan Nüfuz Tespit Sistemidir ve geliştirilmesi devam etmektedir.

Dsniff: Oldukça tanınmış bir paket koklama uygulama paketidir. İçeriğinde özellikle şifre gibi gizli bilgileri yakalamaya çalışan uygulamalar barındırmaktadır.

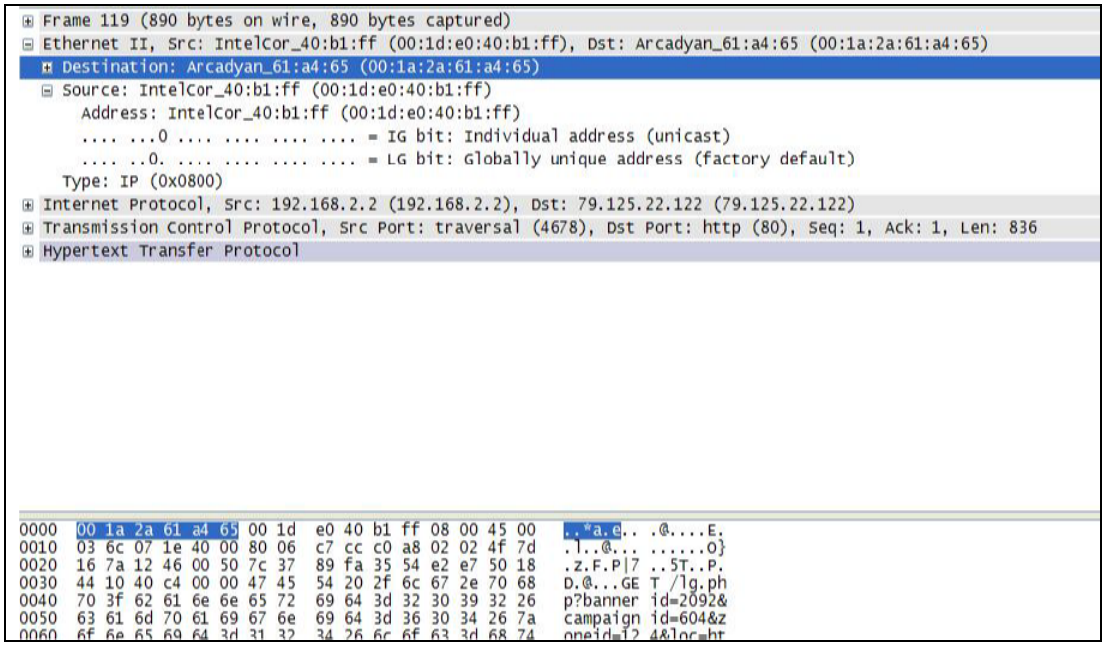
Cain & Abel: Windows işletim sistemi için üretilmiş, şifre (password) bulma programıdır. Çeşitli formattaki şifreleri; ağı koklayarak, şifrelenmiş (enctypted) haldeki şifreleri (password) sözlük (dictionary), brute-force gibi yöntemleri kullanarak elde eder, VoIP paketlerini kaydeder, yönlendirme protokollerini inceler. Özellikler ağ yöneticileri, ağın güvenliğinden sorumlu kişilerin kullanımı için üretilmişti.

2.2. Paket Koklama Tekniği

IEEE 802.3 Ethernet Yerel Alan Ağ, çoklu yayın (broadcast) teknolojisine göre çalışır. Bir mesaj ağ üzerinde başka bir makineye gönderilmek için yayınlandığında, bu mesaj bir hub ile ağa bağlı bütün makinelere yayınlanır. Mesajı alan makinenin ağ arayüz kartı gelen paketin hedef MAC adresi ile kendi MAC adresini karşılaştırır. Eğer MAC adresleri eşleşirse kart mesajı alır aksi takdirde mesajı yok sayar [1].

Paket koklayıcılar ise ağ arayüz kartlarını “promiscuous” modunu aktif hale getirir. Bu mod karta gelen paketlerin MAC adresi ile kendi MAC adresi uyuşmasa bile paketleri yok saymaz, gelen bütün paketleri kabul eder [1].

Bir makine başka bir makine ile Ethernet üzerinden haberleşirken, hedef makinenin Internet Protokolü adresi ve port numarasını belirtir. IP adresi 32 bit uzunluğa sahip bir adrestir. Ethernet kartlarının ise 48 bit uzunluğa sahip MAC (Media Adress Control) adresleri bulunmaktadır (Şekil 2.2). MAC adresleri bütün kartlar için özel olacak şekilde tasarlanmıştır. 48 bitlik MAC adreslerinin ilk 24 biti üretici firmayı temsil eder. Aynı firma tarafından üretilen bütün kartların ilk 24 biti aynı değere sahiptir. Sonraki 22 bit kartın seri numarasını gösterir ve üretici tarafından verilir. Kalan 2 bitin 1 tanesi çoklu yayın adresi olup olmadığını, diğeri ise kartın adresinin elle geçici olarak atanıp atanmadığını gösterir. IP paketleri Ethernet çerçevelerinin içine koyularak yayınlanır [1].



Şekil 2.2. Örnek MAC Adresi

IP paketleri parçalanarak, Ethernet çerçeveleri içine sığacak hale getirilir. Herbir Ethernet çerçevesinin hedef ve kaynak adresi bulunmaktadır ve maksimum 1500 bayt uzunluğundadır [1].

Anahtar (switch) ile oluşturulan ağlarda hub ile oluşturulan ağlara göre paket koklama işlemi daha zordur. Hub kendisine gelen paketleri üzerinde hiçbir işlem yapmadan aynen kendisine bağlı bütün makinelere gönderirken, anahtarlar ise her paketi hedef adresine uygun olan makineye gönderirler.

Ancak anahtar ile oluşturulan ağlarda da paket koklama işlemi yapılabilir, bunun için aşağıdaki yöntemler uygulanabilir [2]:

Anahtar Taşması (Switch Flooding): Bazı anahtarların hub gibi davranıp gelen bütün paketleri ağdaki bütün cihazlara göndermesi sağlanabilir. Bu işlem için büyük sayılardaki yapay MAC adresleri ile anahtarların adres tablosunun tamamen doldurulup taşırılması yeterlidir. Bu tablonun taşması ile bazı anahtarların güvenlik önlemleri etkisiz hale kalmaktadır. Dsniff adlı ağ koklama paketi içindeki macof adlı program MAC adresi taşması oluşturmak için geliştirilmiştir [2].

ARP Yeniden Yönlendirmesi (ARP Redirect): Ağ üzerindeki bir cihaz başka bir cihazın MAC adresini öğrenmek istediğinde ARP isteği gönderir. Bütün cihazlar MAC adreslerini bir ARP tablosunda tutarlar. ARP paketleri anahtar üzerinde çoklu olarak iletilir (broadcast), bu yüzden ağdaki bütün cihazlar MAC adreslerine ait bilgileri görebilir. Bu bilgi çeşitli yöntemlerle koklamayı yapan cihaza ağdaki trafiğin yönlendirilmesi amacıyla kullanılabilir. Bu yöntemlerden bir tanesi, nüfuz yapan cihazın sahte bir MAC adresi ile anahtara kendisini başka bir cihaz gibi tanıtması ve o cihaza ait paketleri alması şeklindedir. Bir diğer yöntemde ise nüfuzu yapan cihaz kendisini anahtara bir yönlendirciymiş (router) gibi tanıtması şeklinde olur. Bunun sonucu olarak cihazlar paketlerini koklamayı yapan cihaza yönlendirirler. Bir diğer yöntemde ise koklamayı yapan cihaz sadece belirli bir cihaza ARP paketleri yollayarak kendisini yönlendirici gibi tanıtır ve o cihazın gönderdiği paketleri alır [2].

ICMP Yeniden Yönlendirmesi (ICMP Redirect): Ağ üzerinde cihazlar aynı fiziksel bölüm (segment) ve anahtar üzerinde ancak farklı IP altağından (subnet) dolayı farklı mantıksal (logical) bölümde olabilirler. Bu durumda bir cihaz diğerine paket gönderirken, bu paketi yönlendiriciye gönderir, yönlendirici bu iki cihazın aynı fiziksel bölümde olduğunu bilir ve ICMP paketi ile gönderen cihaza, hedef cihaza

paketleri direk olarak gönderebileceği bilgisini iletir. Nüfuz yapan cihaz bu noktada sahte ICMP paketleri göndererek kaynak cihaza kendisini hedef cihaz gibi gösterip ona ait paketleri alabilir [2].

MAC Adresi Değiştirme (Spoofing): Koklama yapmak isteyen cihaz MAC adresini değiştirerek kendisini başka bir cihaz gibi gösterebilir. Linux ve Solaris tabanlı sistemlerde

- **ifconfig eth0 down**
- **ifconfig eth0 hw ether 00:02:b3:00:00:AA**
- **ifconfig eth0 up**

ile MAC adresini değiştirmek mümkündür [2].

2.3. Karşı Atak Yöntemleri

Paket koklama yönteminde de diğer saldıralara karşı olduğu gibi çeşitli önlemler almak mümkündür. Bunun için öncelikle ağ üzerinde koklayıcı programın olup olmadığını belirlemek önemlidir. Kullanıcı çeşitli sayıdaki koklayıcı bulma yöntemlerini kullanarak programın varlığını kestirebilir. Bu yöntemlerden bazıları koklayıcı programın çalışmasını durdurabilecek özelliktedir [1].

Ağ üzerinde koklayıcı bulma yöntemlerinden bazıları:

- Kullanıcı geçici olarak kendi kartının MAC adresini değiştirir. Kullanıcı bu adresle paketler üretebilir. Eski adresine paket yolladığında normal şartlarda bu paketlerin yerine ulaşmaması beklenir. Eğer ağ üzerindeki bir makine bu paketleri alıyorsa, bu makine üzerinde bir koklayıcı çalıştığı sonucuna ulaşılabilir.

- AntiSniff gibi bazı programlar yardımıyla koklayıcılar bulunabilir. Bu programlar saldırı amacı içermeyen çeşitli testler ile ağ yöneticisine sistemde koklayıcı çalışıp çalışmadığı bilgisini ulaştırabilir. Koklayıcı çalışması için öncelikle ağ arayüz

kartının “promiscuous” modda olması gereklidir, ağ arayüz kartının “promiscuous” modda olup olmadığını anlamak için cpm, ifstatus, chkrootkit gibi programlar kullanılabilir. AntiSniff dışında sentinel, sniffdet, kstat 1.1-2, neped programları yardımıyla da koklayıcılar bulunabilir.

SNMP (Simple Network Management Protocol) kullanılarak da koklayıcının varlığı tespit edilebilir. SNMP TCP/IP ağlarda ağ yöneticisine ağ üzerindeki sorunun yerini gösteren ve çözümü hakkında yardım eden bir protokoldür. SNMP sunucu kullandığı model yardımı ile alınan ve gönderilen paket sayısı gibi parametleri toplar. SNMP ile belirli bir porta olan bağlantılar görüntülenebilir. Bu sayede Ağ üzerinde koklayıcının varlığı tespit edilebilir.

2.4. Korunma Yöntemleri

Basit koklayıcı programlardan korunma için öncelikle hub yerine anahtar tercih edilebilir, ancak daha öncede anlatıldığı gibi anahtar ile oluşturulmuş ağlarda da açık noktalar oluşturabilecek yöntemler bulunmaktadır. Buna rağmen anahtar kullanmak ağ üzerinde güvenliği arttırmak ve performansı yükseltmek için hala en iyi yöntemlerden biridir. Anahtar kullanımı basit koklayıcılara karşı güvenlik sağlasa da, verileri korumak için en iyi yöntem verilerin şifrelenmesi ile oluşturulur. Şifreleme ile oluşturulan trafik üzerinden yine veriler aynı yöntemlerle toplanabilsede, elde edilen veriler okunamaz halde olacaktır. Sadece hedef kullanıcı bu şifrelenmiş veriyi açarak okuyabilecek yeterliliktedir [2].

Bazı şifreleme yöntemlerinde paketlerin sadece veri bölümü şifrelenir, başlık bölümleri şifrelenmez bu durumda koklamayı yapan cihaz paketlerin hedef ve kaynak adreslerini ve ağın yapısını ele geçirebilir.

VPN şifreleme ve kimlik doğrulama kullanarak güvensiz bir ağ üzerinden güvenli bir iletişim hattı oluşturabilir. VPN hem internet gibi genel ağlarda hem de yerel alan gibi kapalı ağlarda veri iletişimini korur. VPN methodlarından bazıları [2]:

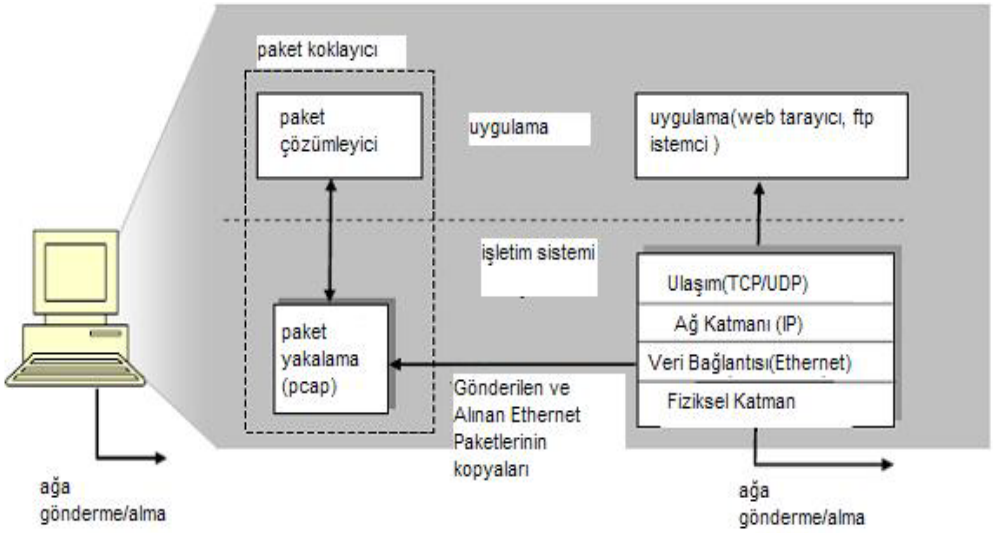
– **SSH**: SSH uygulama katmanında TCP tabanlı olarak çalışan ve istemci sunucu arası haberleşmeyi güvenli hale getiren bir VPN uygulamasıdır. Kimlik doğrulama gerektiren Telnet, FTP gibi uygulamalarda kullanılır. Ancak TCP kullanan başka uygulamalar da SSH bağlantısını tünel olarak kullanabilir. Bu yöntemle SSH çok fazla sayıda uygulama için kullanılır. SSH kimlik doğrulama için RSA veya DSA algoritmalarını kullanmaktadır. SSH uygulamasında paketlerin başlık kısımları şifrelenmez bu yüzden koklama programı kaynak ve hedef adresleri görebilir.

– **SSL(Secure Sockets Layer)/TLS(Transport Layer Security)**: SSL ilk olarak Netscape tarafından güvenli internet iletişimi için geliştirilmiştir. Daha sonra TLS ile yer değiştirdi. TLS ulaşım katmanı seviyesinde bir güvenlik sağlar.

– **IPSec(IP Security)**: IPSec ağ katmanı seviyesinde bir güvenlik sağlar. IP paket başlığını genişleterek IPv4 ve IPv6 protokollerinde paket seviyesinde güvenlik içerir.

BÖLÜM 3. WIRESHARK

Wireshark, ilk olarak 1997 yılında Ethereal adıyla geliştirilmeye başlanmış ve 1998 Temmuz'unda ilk sürümü yayınlanmıştır. Wireshark libpcap (Promiscuous Capture Library) tabanlı bir paket yakalama formatına sahiptir [3] ve genel olarak Şekil 3.1'de görüldüğü gibi bir sistem yapısı vardır:



Şekil 3.1. Wireshark Genel Yapısı

Wireshark şu anda kullanılan en iyi açık kaynak kodlu ağ yorumlayıcılarından biridir. Birçok geliştirici tarafından sürekli olarak geliştirilen, yeni eklentileri çıkarılan oldukça kararlı çalışan bir yazılımdır [3].

Temel olarak Wireshark ağ üzerinden paketleri alır, bu paketleri parçalarına ayırarak inceler ve kolay anlaşılır bir kullanıcı grafik arayüzü ile sergiler. Wireshark programının en önemli yanı açık kaynak kodlu olması, sürekli geliştirilmesi ve ücretsiz serbest olarak kullanılabilmesidir [3]. Diğer önemli özellikleri:

– GNU GPL açık kaynak lisansı ile dağıtımı yapılır.

– “Promiscuous” ve “non-promiscuous” modlarında çalışabilir.

– Ağ üzerinden veya daha önceden kaydedilmiş bir dosyadan gelen paketleri alabilir.

– Oldukça kolay okunabilen ve konfigüre edilebilen bir arayüze sahiptir.

– Zengin gösterim filtrelerine sahiptir.

– Tcpdump formatında paket yakalama filtrelerini destekler. Bir TCP oturumunu tekrardan oluşturarak bunu ASCII, EBCDIC, hex dump ve C dizileri olarak gösterebilir.

– Kaynak koduna erişim imkanı vardır. Kaynak kodu C programlama dili ile yazılmıştır.

– Çalıştığı platformlar [4]:

UNIX:

– Apple Mac OS X

– BeOS

– FreeBSD

– HP-UX

– IBM AIX

– NetBSD

– OpenBSD

– SCO UnixWare/OpenUnix

– SGI Irix

– Sun Solaris/Intel

- Sun Solaris/Sparc
- Tru64 UNIX (Digital UNIX)

LINUX:

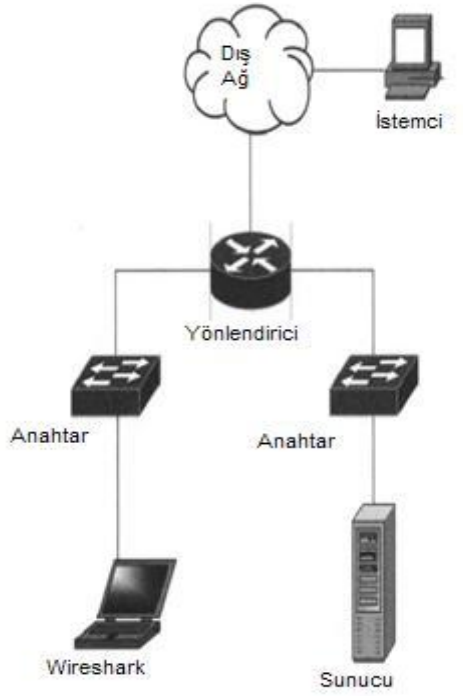
- Debian GNU/Linux
- Gentoo Linux
- IBM S/390 Linux (Red Hat)
- Mandrake Linux
- PLD Linux
- Red Hat Linux
- Rock Linux
- Slackware Linux
- Suse Linux

WINDOWS:

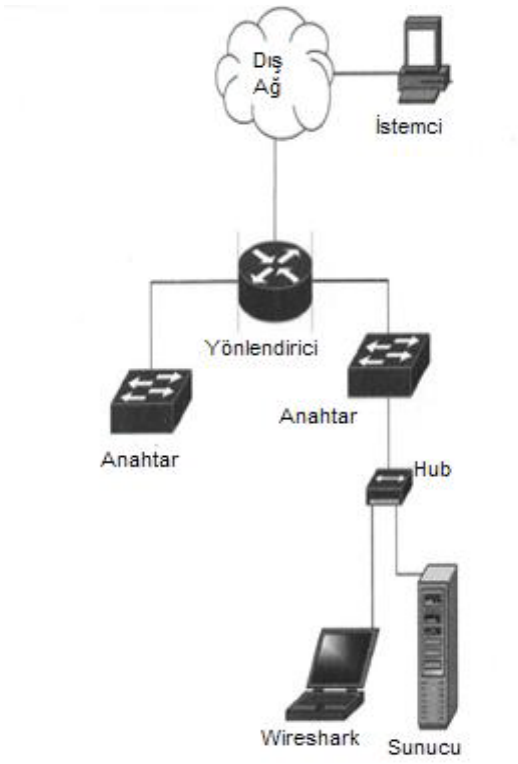
- 750'den fazla protokolu destekler ve açık kaynak olduğu için bu sayı sürekli artış göstermektedir. Desteklediği protokollerden bazıları:
- AppleTalkProtocol Ailesi: LLAP, AARP, DDP, NBP, ZIP, ATP, ASP, AFP,…
- FieldBusProtocol Ailesi: BACnet, PROFIBUS, PROFINET,…
- InternetProtocol Ailesi: ARP, IP, ICMP, UDP, TCP, DCCP, HTTP, FTP,LAN Protokol Ailesi: Ethernet, FDDI, TokenRing, IEEE_802.11,...
- Protokollerin güncel listesi [10] ve [11]'den bulunabilir.
- 25'in üzerinde ürüne ait kayıt dosyasından paketleri okuyabilir.
- Çok değişik formatlarda dosyaya kayıt yapabilir.
- Ethernet, Token Ring, 802.11 Kablosuz iletişim vb. birçok değişik ortamdaki paketleri yakalayabilir.
- Tshark isimli komut satırından çalışan bir versiyonu da vardır.

3.1. Ağ Üzerinde Wireshark Yerleşimi

Wireshark ile uygun ve etkili bir şekilde paket toplamak için öncelikle ağ üzerinde doğru bir noktaya kurulumunun yapılması gerekmektedir. Genel olarak ağın çok değişik noktalarından paket toplama ihtiyacı olabileceği için Wireshark'ı taşınabilir bir bilgisayara kurmak, bir hub ve düz ve/veya çapraz ağ kablosu bulundurmak en uygun çözümdür. Şekil 3.2Şekil 3.2. Wireshark'ın yanlış kurulumu'de istemci (client) ile sunucu (server) arasındaki paketler alınmak istiyorsa Wireshark'ın ağ üzerindeki yanlış kurulumu gösterilmiştir. Ağdaki yönlendirici (router) sunucuya gelen paketleri sunucunun bağlı olduğu anahtara yönlendirecektir. Şekil 3.3'de ise istemci ile sunucu arasındaki haberleşme hub yardımı ile incelebilmektedir [3].



Şekil 3.2. Wireshark'ın yanlış kurulumu

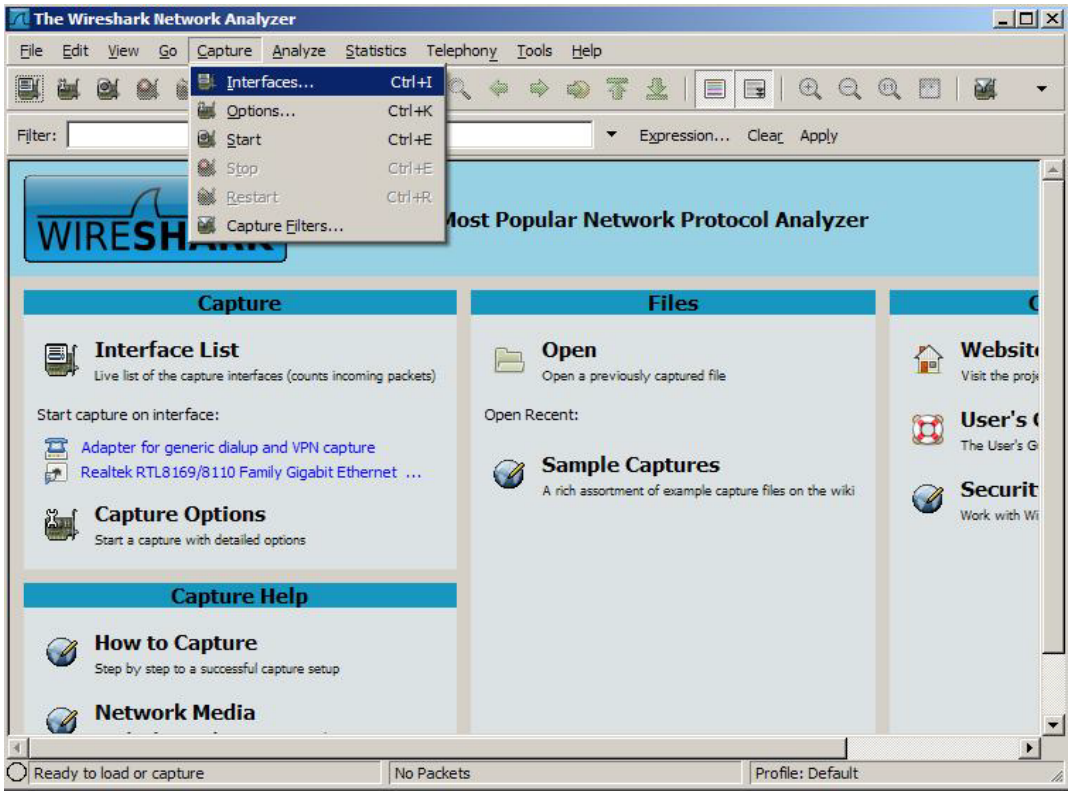


Şekil 3.3. Wireshark'ın doğru kurulumu

3.2. Wireshark Programının Kullanımı ve Çalışma Ortamı

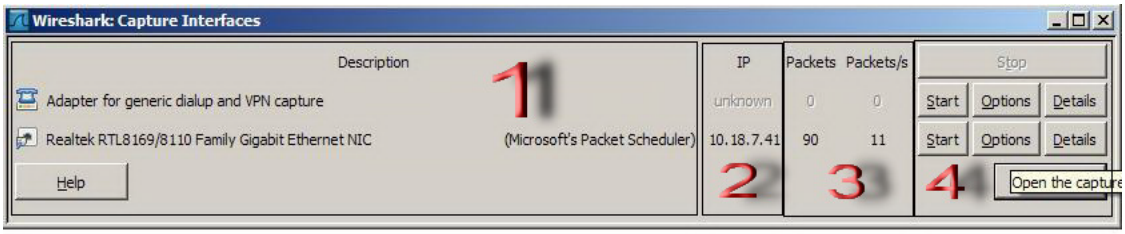
Bu bölüm içerisinde Wireshark programının kullanımı ile çalışma ortamı hakkında bilgiler verilecektir.

– Hangi ağ arayüzünden paket yakalanacağı seçilir (Şekil 3.4).



Şekil 3.4. Ağ Arayüz Seçiminin Açılması

– Uygun Ağ arayüzleri listelenir ve özellikleri incelenmek için “options“ tuşuna basılır (Şekil 3.5).



Şekil 3.5. Paket Yakalama İşleminin Yapılacağı Ağ Arayüzünün Seçilmesi

1. Description: Ağ arayüzüne işletim sistemi tarafından verilen tanımlamadır.

2. IP: İlgili arayüze ait ilk IP adresidir. Eğer ağ arayüzüne ait bir IP adresi yoksa “unknown” ibaresi yazılır. Eğer ağ arayüzüne ait birden fazla IP adresi varsa ilk IP adresi yazılır. Ancak ilk olarak hangisinin seçileceği tahmin edilemez.

3. Packet: Pencere açık olduğu sürece ilgili arayüz tarafından yakalan paketleri gösterir. Eğer son 1 saniye içinde paket alınmamışsa gri olarak sergilenir.

Packets/s: Son 1 saniye içinde yakalanan paket sayısını gösterir. Eğer son 1 saniye içinde paket alınmamışsa gri olarak sergilenir.

4. Stop:Aktif olarak devam eden paket yakalama işlemi sonlandırılır.

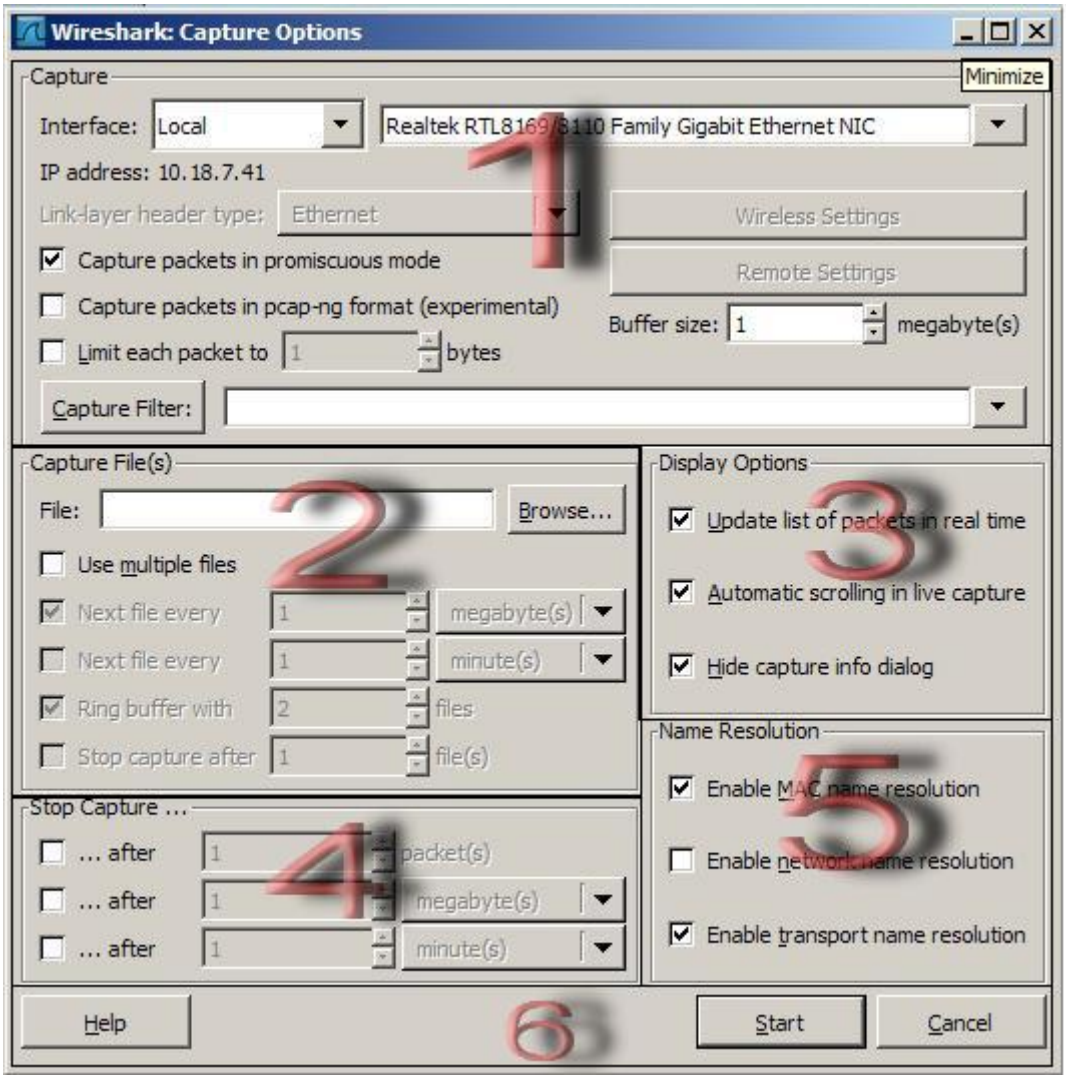
Start: İlgili ağ arayüzüne ait son paket yakalama ayarlarına göre paket yakalama işlemi başlatılır.

Options: Paket yakalama ayarlarının düzenlenebileceği pencere açılır.

Details: Ağ arayüzüne ait detaylı bilgilerin görüntülenebileceği pencere açılır. Bu özellik sadece Windows işletim sisteminde vardır.

Close: Ağ arayüzlerinin listesini gösteren pencere kapatılır.

– İlgili Ağ kartının özelliklerini gösteren pencere açılır (Şekil 3.6).



Şekil 3.6. Ağ Arayüzünün Özelliklerini Gösteren Pencere

1. Paket Yakalama Özellikleri

Bu alanda paketlerin yakalanmak istediği ağ arayüzü seçimi yapılır. Seçilen ağ arayüzünün IP adresi ve link katmanı başlık bilgisi görüntülenir. Bu alandan işletim sisteminin yakalama süresince ayıracağı tampon bölge miktarı ayarlanır. Paketlerin "promiscuous" modda yakalanıp yakalanmayacağı seçilebilir. Eğer "promiscuous" aktive edilmezse sadece o arayüze gelen ve giden paketler yakalabilir. Yine bu alan üzerinden her paketin data alanının ne kadarlık bölümü alınacağı ayarlanabilir. Bütün paketler alınmak istenmiyorsa filtreleme işlemi yine bu alanda yapılabilir.

2. Paket Kayıt Dosyası Özellikleri

Yakalanan paketler daha sonra incelenmek üzere dosyaya kaydedilebilir. Hangi dosyaya kaydedileceği bu alan üzerinden seçilir. Dosyaya kayıt işleminin belirli kriterlere göre birden fazla dosyaya bölünmesi işlemi yapılabilir.

3. Görüntüleme Özellikleri

Bu alanda üzerindeki seçenekler ile yakalanan paketlerin gerçek zamanlı olarak listelenmesi, paketler geldikçe listenin aşağı doğru kayması ve yakalama bilgi penceresinin görüntülenip görüntülenmeyeceği ayarlanabilir.

4. Paket Yakalama Sonlandırma Özellikleri

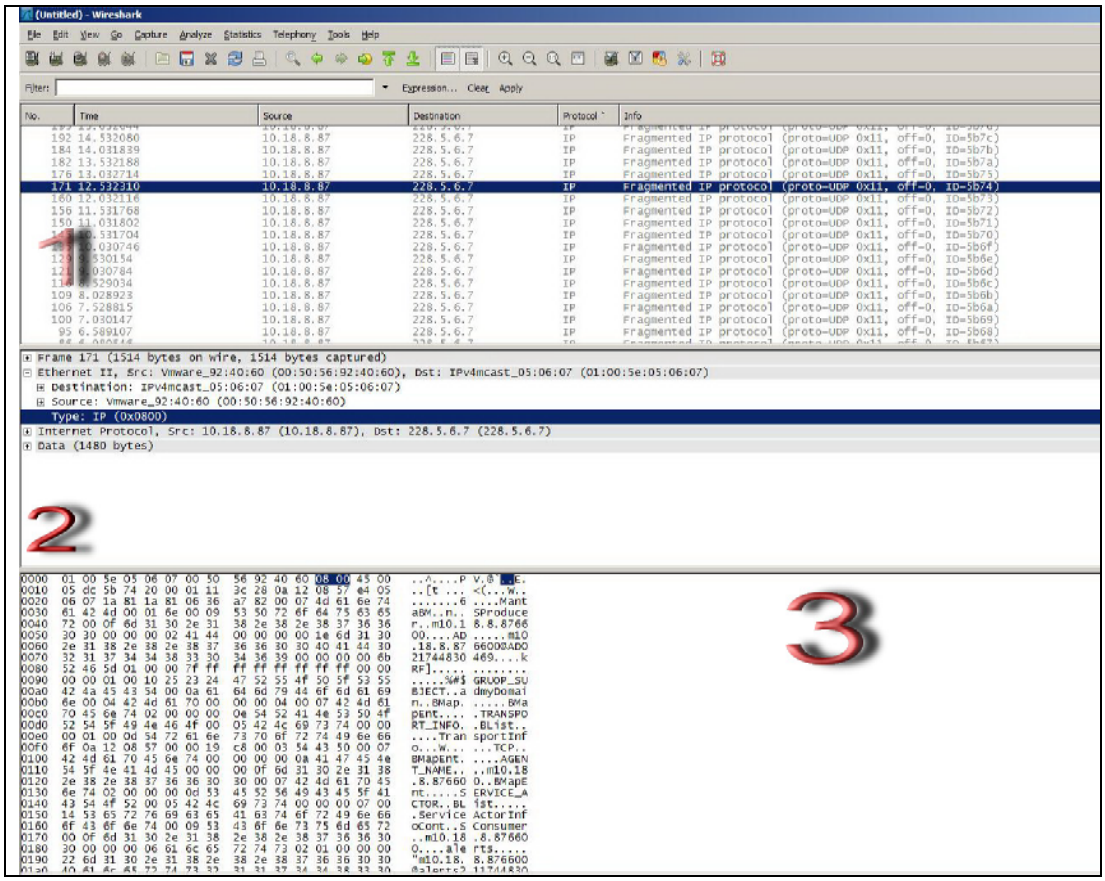
Bu alanda paket yakalamanın, belirli bir sayıda pakete ulaşıldığında veya paketlerin toplam boyutu belirli bir değere ulaştığında veya paket yakalama işlemi için belirlenen süre dolduğunda, sonlandırılması ayarlabilir.

5. Adres Çözümleme Özellikleri

Bu alanda yakalanan paketlerin MAC adreslerinin, ağ adreslerinin (IP adresi) ve ulaşım katmanı adreslerinin (port) çözümlenmesi ayarlanabilir.

6. Pencere Genel Tuşları

“Start” tuşu ile yapılan ayarlara göre paket yakalama işlemi başlatılır.
“Cancel” tuşu ayarların iptal edilmesini ve pencerenin kapanmasını sağlar.
“Help” tuşu ile yardım penceresi görüntülenir.
“Start” tuşuna basılarak paket yakalama işlemi başlatılır. Paketler Şekil 3.7’de görüldüğü gibi listelenir. Herhangi bir paketin üstüne tıklandığında seçili hale gelir ve paketin içeriği ilgili katmanlara göre görüntülenilir. Örneğin Şekil 3.7’de veri bağı katmanındaki Ethernet paketinin veri kısmında bulunan paketin bir üst katmandaki IP’ye (dataField: 0x0800) ait olduğu kısım görüntülenmektedir.



Şekil 3.7. Paket Yakalama Örneği

1. Yakalanan paketler alt alta listenir. Bu alanda paketlerin,

– Yakalanma sırası,

– Zamanı,

– Kaynak ve Hedef IP adresleri,

– Protokol bilgisi,

– Açıklama bölümü görüntülenir.

2. 1.bölümde seçilen paketin her ağ katmanına ait ayrıntılı içerikleri görüntülenir. Paketin ait olduğu protokole göre bu alandaki özellikler değişkenlik gösterir.

3. Bu alanda 1.bölümde seçilen paketin 16'lık düzende içeriği sergilenir.

3.3. Wireshark Kullanım ve Uygulama Alanları

Wireshark bir paket yakalama uygulamasıdır ve uygulamanın en büyük amacı ağ üzerinde mümkün olan bütün paketleri yakalamak ve olabildiğince ayrıntılı bir şekilde incelemektir. Bir elektrikçinin kablodaki gerilimi ölçmesi gibi bir paket yakalayıcı ağ kartındaki paketleri inceler ve ağ kablosunun içinde ne olduğunu bulmaya çalışır. Eskiden bu tarz bir uygulama hem geliştirilmesi zor hem de oldukça maliyetliydi. Ancak Wireshark ile bu problem ortadan kalmış oldu. Wireshark'ı kullanarak:

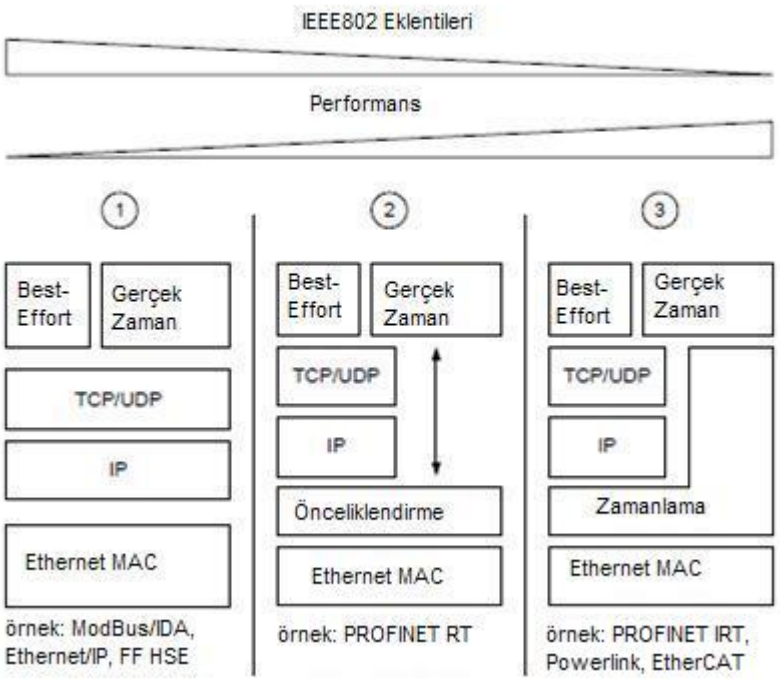
- Ağ yöneticisi ağ üzerindeki problemleri inceleyebilir,
- Güvenlik yöneticileri güvenlik açıklarını tespit edebilir,
- Uygulama geliştiricileri istedikleri protokolün paket yapısını inceleyebilir,
- Ağ protokollerinin iç yapısı ilgilenenler tarafından öğrenilebilir.

Wireshark bu örneklerin yanısıra sayısız alanda ve amaçla kullanılabilir.

BÖLÜM 4. PROFINET

Endüstriyel veri iletişim sistemleri, günümüz modern üretim süreçlerinin otomasyonu için gerekli en önemli bileşenlerden biridir. Üretim sürecine ilişkin verilerin toplanması ve değerlendirilmesi bu sistemler aracılığıyla gerçeklenir. Endüstriyel tesislerin işletiminin düzenli bir şekilde yapılabilmesi, sistemin uzaktan izlenmesi ve sistemde oluşan hataların gerçek zamanlı aktarımı ile olanaklıdır. Günümüzde, üretici kuruluşların kendi ürünleri arasındaki iletişimi sağlamak üzere geliştirdiği birçok veri iletişim sistemi bulunur. Profibus, Profinet, DeviceNet, EtherNet/IP, Modbus Plus, Modbus-RTU, Modbus-TCP, CC-Link, ControlNet, CANopen, Interbus, FIPIO, Powerlink, EtherCat, Fieldbus Foundation, Lonworks, AS-Interface ve FL-net bu tür sistemlere örnek gösterilebilir. Endüstriyel veri iletişiminde çok sayıda sistem ve protokol kullanılması, farklı ürünler arasındaki iletişimin sağlanmasında sorun yaratmaktadır. Bu sorunları aşmak için, belirli standartlar ve protokoller belirlenerek bu çeşitliliği azaltma yolunda çalışmalar yapılmıştır. Buna örnek olarak, 15 Mart 1996'da Profibus iletişim sisteminin Avrupa standardı olarak kabul edilmesi gösterilebilir. Profibus, günümüzde en yaygın kullanıma sahip iletişim sistemlerinin başında gelmektedir. Profinet ise bilgi teknolojileri (IT) standardı olarak kabul görmüş Ethernet altyapısı üzerine kurulmuş bir Profibus düzenlemesidir [7].

Günümüzde Endüstriyel Ethernet protokolleri Şekil 4.1'de görüldüğü üzere 3 ana sınıfa ayrılırlar. Protokoller, 1. kategoriden 3.kategoriye doğru veri bağı katmanında ek fonksiyonlar kullanılarak daha performanslı hale gelir [8].



Şekil 4.1. Endüstriyel Ethernet Protokolleri Sınıflandırması

– ModBus/IDA, Ethernet/IP, FF HSE ilk kategoriye örnek protokollerdir. Protokoller, sadece TCP/IP'nin üzerine endüstriyel otomasyona özgü uygulama katmanı konularak, Ethernet'i olduğu gibi kullanırlar. TCP/IP protokol yığınının tamamı kullanıldığı için gerçek zamanlı veri iletiminde performans kaybı yaşanır. Veri iletim hızı yaklaşık 100 milisaniye civarındadır [8].

– İkinci kategoriye ait protokolleri kullanan endüstriyel otomasyon sistemleri Ethernet standartının hızı ile kabul edilebilir gerçek zamanlı veri transfer hızı arasında veri iletimi yapabilmektedirler. Profinet RT bu kategoriye örnek olan bir protokoldür. Bu kategorideki protokoller Ethernet MAC katmanında IEEE 802.1D/Q'nün öncelik (priority) yapısını kullanırlar. Buna ek olarak ağ (network) ve ulaşım (transport) katmanları optimizasyon amaçlı olarak atlanır (bypass). Bu optimizasyon sonucu veri transferi 10 milisaniye mertebesine çekilebilir [8].

Sanal Yerel Alan Ağları IEEE 802.1Q standartı ile tanımlanmıştır. Genel olarak bir yerel alan ağ içinde anahtarlar yardımı ile mantıksal olarak farklı alt ağlar (subnet) kurmak için kullanılır. IEEE 802.1Q Ethernet çerçevesinin içine 32 bit uzunluğunda ek Qtag konulmasını tanımlar. Qtag alanı Ethernet çerçevesi içerisinde kaynak MAC adresi ile çerçeve tipi arasına yerleştirilir [9].

Qtag ek olarak 4 alan içerir. Tip bölümü 2 bayt uzunluğunda ve sabit 0x8100 değerine sahiptir. Diğer alanlar ise sırasıyla 3 bit uzunlğunda öncelik (PRI), 1 bit format göstergeci (CFI) ve 12 bit uzunluğunda VLAN ID'dir. VLAN ID ilgili çerçevinin hangi VLAN'a ait olduğunu gösterir. 3 bit uzunluğundaki öncelik alanı en büyüğünün değeri 7 olmak üzere toplam 8 adet öncelik değeri ve servis sınıfı tanımlanmasını sağlar (Şekil 4.2). Eğer çerçeve tarfik önceliklendirmesi için kullanılıyorsa öncelik değeri 0 olur [9].

802.1 Qtag Type (0x8100) 16 bit	Priority (Öncelik) 3 bit	CFI (Format Gösterici) 1 bit	VLAN ID 12 bit

Şekil 4.2. VLAN QTag Çerçeve Yapısı

– Endüstriyel otomasyonda veri iletimini daha zaman kritik yapmak MAC katmanına zamanlama fonksiyonu eklemek ile mümkün olabilmektedir. MAC katmanına zamanlama özelliğini ekleyen protokoller üçüncü kategoriyi oluşturur. Bu protokoller özel bir yazılım ve/veya donanım ihtiyacı duyarlar. 3. Kategoriye ait protokoller ile 1 milisaniye civarında veri iletimi yapmak mümkün olmaktadır [8].

Ethernet yapısı itibariyle önceden tahmin edilebilir bir çalışma düzenine sahip değildi. CSMA/CD erişim yönteminin kullanımı ve gönderilen paketlerin ağ üzerinde çarpışmalarla karşılaşabilecek olması olasılığı Ethernet'in çalışmasını önceden tahmin edilemez hale getirmekteydi. Ancak zamanla veri iletim hızlarının artması ve hub yerine anahtarların kullanılması ile Ethernet'in performansında ciddi artışlar elde edildi. Önce transfer hızının 100Mbps seviyesine ulaşması ile Hızlı (Fast) Ethernet kullanılmaya başlandı, hızın artmasının yanı sıra tamamen

anahtarlamalı ağların kullanılmaya başlanması sonucu çerçevelerin çarpışmasının önüne geçilmiş oldu. Ethernet'in iyice yaygınlaşmaya başlaması ile fiyatlarda ciddi düşüşler elde edildi ve artık Ethernet'in endüstriyel ortamda kullanılmaya başlanmasının önü açılmış oldu [9].

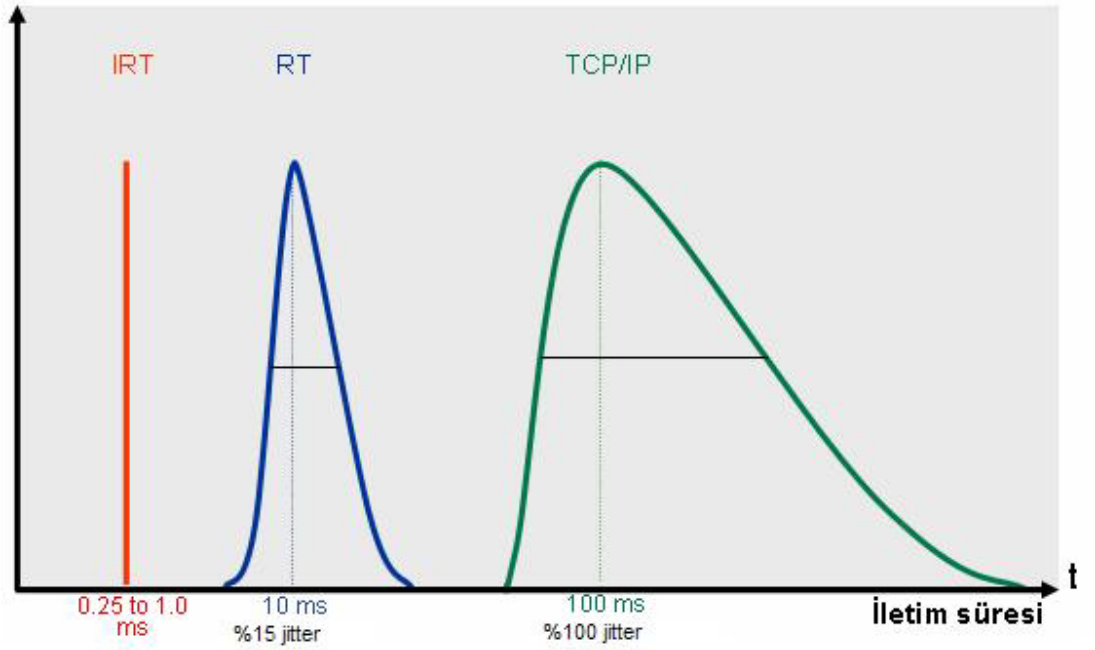
Anahtarlamalı Ethernet yerel alan ağlarda trafik yoğunluğunun artmasına karşı kullanılan bir çözümdür. Anahtarlar kullanılarak çift yönlü (full-duplex) iletişim yapılabilir ve bunun sonucu olarak veri akışında tek yönlü iletişime oranla 2 kat fazla miktarda veri iletilebilir. Anahtar cihazlar, gelen çerçeveyi bir sonraki cihaza iletim yöntemindeki farklılığa göre ikiye ayrılır. Depola-Gönder (Store & Forward) ve Doğrudan Aktarmalı (Cut-through). Depola-Gönder anahtarları, gelen çerçeveyi göndermeden önce tamamını depolayıp bütünlüğünü kontrol ettikten sonra alıcıya iletirler. Bu yöntem çerçevenin bütünlüğünü korurken iletişim zamanının uzamasına neden olur. Doğrudan Aktarmalı yöntemine göre çalışan anahtarlar ise gelen çerçeveyi hiç bekletmeden alıcıya iletirler. Bu yöntem hızlı iletişime imkan sağlarken iletişim sırasında oluşabilecek hatalı çerçevelere karşı korunaksızdır [9].

Doğrudan Aktarmalı anahtarlar, Depola-Gönder anahtarlarına göre sadece çerçevenin iletim hızının artmasını sağlar. Ağdaki tıkanıklık durumunda ise, çıkış tamponunda depolanan çerçevelerin çoğalmasından dolayı hangi tür anahtar kullanıldığı farketmeden ağdaki iletişimde bir gecikme yaşanması doğaldır. Tıkanıklık anahtara gelen trafiğin çıkışta kullanılan band genişliğinden fazla olması durumunda oluşur. Tıkanıklık oluştuğu durumda çıkış tamponunda depolanan çerçeveler sırayla uygun band genişliği oluştukça iletilir, bu durum ağ üzerinde gecikmelere (latency) ve iletim zamanlarının farklılaşmasına (jitter) neden olur. Tamponun dolması ve gelen çerçevelerin silinmesi de olasıdır [9].

Ethernet'in endüstriyel alanda kullanılmaya başlaması yani saha seviyesine inmesi için gerçek zamanlı veri iletişimine mümkün olduğunca imkan vermesi gerekmektedir. Birçok otomasyon üreticisi kendi gerçek zamanlı endüstriyel Ethernet çözümlerini üretmektedir. Birçok gerçek zamanlı endüstriyel Ethernet çözümünün olması her ne kadar belirli uygulamalar için en iyi çözümü üretse de diğer üretici

firmaların ürünleri ile uyum sorunları yaşanmaktadır. Hemen hemen bütün üretici firmaların ortak noktaları sadece 100Mbps tamamen çift yönlü (full-duplex) anahtarlı Ethernet kullanmaları ile sınırlı kalmaktadır. Bazı çözümler ağ üzerinde gerçek zamanlı veriyi iletirken TCP/IP vb. diğer protokollerin trafiğini sınırlandırmak gibi Ethernet standartına uymayan yapılara sahiptirler [9].

Endüstriyel Ethernet üzerinde kullanılması planlanan bir protokolun uygun olup olmadığını ölçmek için iki önemli faktöre dikkat edilir (Şekil 4.3). Bunlar, mesajların iletimi için geçen süre ve protokolün ortalama mesaj iletim süreleri arasındaki farkın büyüklüğüdür (jitter) [9].



Şekil 4.3. IRT, RT ve TCP/IP için iletim süreleri ve jitter

1. Kategoriye ait IRT protokolünde iletim süresi 0.25 ms. ile 1 ms. aralığındadır ve jitter değeri 0'dır.
2.Kategoriye ait RT protokolünde iletim süresi ortalama 10 ms. civarındadır ve jitter değeri yaklaşık %15'dir.
3.Kategoriye ait TCP/IP protokolünde ise iletim süresi ortalama 100 ms. civarındadır ve jitter değeri %100 seviyesindedir.

Bir sonraki bölümde, her 3 kategorideki iletişim gereksinimlerini destekleyen Profinet protokolleri ve mimarisi ile ilgili bilgiler verilecektir.

4.1. Profinet Katmanlı Mimari Yapısı

Otomasyon sistemlerinde Ethernet teknolojisinin kullanılması, saha seviyesindeki otomasyon iletişim ağının kurulabilmesi için gerekli olan:

- Çok küçük miktarlarda ve yüksek frekansta oluşan verinin iletilmesi,
- Gerçek zamanlı haberleşme,
- İstasyonlar arasında senkron halde iletişim,
- Topoloji gibi,

sistem özelliklerinin gerçeklenebilmesini sağlar. Günümüz sistemlerinin ihtiyacı olan bu özellikler 100-Mbit veri iletişim hızına sahip bir Ethernet'in belirli özelliklerinin geliştirilmesi ile elde edilebilir. Ethernet teknolojisi ile çalışan otomasyon sistemlerinde bant genişliğinin yanı sıra, istasyonlardaki yazılım gerçeklemelerinin davranışları, ağ topolojisi veya ağ elemanlarının iç özellikleri sistemin çalışma verimliliğini doğrudan etkiler [6].

Ethernet teknolojisi ile çalışan otomasyon sistemlerinde gerçek zamanlılık özelliğinin sağlanması gerekmektedir. Bir sistemin gerçek zamanlı olarak adlandırılabilmesi için sistemin tüm gerçek zamanlılık isterlerini karşılaması gerekmektedir. Otomasyon sistemi her türlü koşullar altında açık ve önceden tanımlı tepki süresini garanti etmesiyle gerçek zamanlı olarak adlandırılabilir. Gerçek zamanlı sistemin aşağıda tanımlı dört kriteri sağlayabilmesi gerekmektedir [6]:

Çalışma zamanı (runtime), çevrim zamanı (cycle time), tepki zamanı (response time): Bu parametreler için tanımlanmış üst limitlerin sistemin çalışması sırasında aşılmaması gerekmektedir.

İletim zamanlarının farklılığı (jitter): Sistemde hız ve hassasiyet ile ilgili isterler arttıkça, iletim zamandaki farklılıkların ve sapmaların azalması gerekmektedir.
Senkronizasyon: Sistemdeki cihazların birbirleri ile eş zamanlı olarak çalışması ve mümkün olan en yüksek doğruluğun sağlanması gerekmektedir.
Üretim miktarı (throughput): Önceden tanımlanmış bir zamanda, önceden tanımlı miktarda ürünün her türlü koşullarda üretilmesi gerekmektedir.

Ethernet yukarıda tanımlı dört isteri sağlayarak gerçek zamanlı olarak adlandırılabilmesi için çeşitli geliştirmelere ihtiyacı vardır. Gerçek zamanlı Ethernet'in sorunsuz bir şekilde çalışabilmesi için aşağıdaki özelliklerin sistemde sağlanmış olması gerekmektedir [6]:

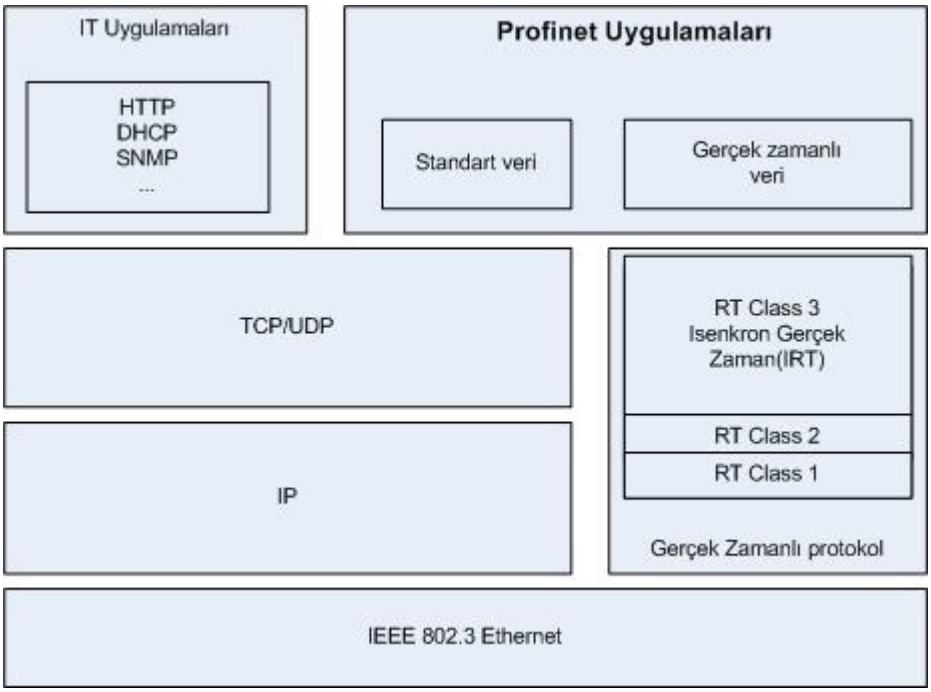
Bölümlendirme/ayrıştırma (segmenting/separation): Sistemde yönlendirici gibi özel olarak tasarlanmış bir cihaz yardımı ile gerçek zamanlı ağ trafiği ile standart Ethernet ağ trafiğinin birbirinden ayrılmış olması gerekmektedir. Trafiklerin ayrılmaması durumunda, sistemin aşırı yüklenmesi gibi durumlarda tepki süresi tahmin edilemez bir hale gelebilmektedir.
Zaman slotları oluşturma: Gerçek zamanlı sistemler genellikle kendini tekrar eden çevrim zamanı aralıklarında tanımlı benzer görevlerden oluşur. Bu yapı iletim çevrimi zamanının belirli slotlara ayrılması ile sağlanır. Gerçek zamanlı verilerin iletimi için kullanılan slot ile her çevrimde aynı zamanda gerçek zaman verisinin iletimi sağlanır.
Zaman senkronizasyonu: Sistemde bir çok işlem aynı işi yapmak üzere aynı anda tetiklenir. Bunun için bütün istasyonların iç saatlerinin kabul edilebilir farklılıklar dışında aynı olması gerekmektedir.

Ethernet üzerinden gerçek zamanlı iletişimin kurulabilmesi için yöntemler vardır. Bunlardan biri TCP/IP, UDP/IP gibi stardart iletişim protokollerinin kullanılmasıdır. Ancak standart protokolleri kullanmak çerçevelerin eklerinin artması ve bunun sonucunda iletim zamanının uzamasına neden olmaktadır. Ayrıca bu protokollerin işlemci üzerindeki işlem sürelerinin büyük olması veri gönderim sırasında çevrim süresinin artmasına neden olmaktadır. Standart protokol yığınında yapılacak bazı

değişiklikler ile işlem süresinin azaltılması ve veri iletişim hızının arttırılması mümkündür. Ancak bu değişiklikler sonucunda standart protokol yığını artık standart bir ürün olmak özelliğini kaybeder [6].

Gerçek zamanlı iletişimde 3. ve 4. katman protokolleri sadece belirli miktarda çevrimsel veri değişimi için kullanılır, gerçek zamanlı iletişimin sağlanabilmesi için IEEE 802.1'e göre optimize edilmiş 2.katman protokolünün kullanılması tavsiye edilir. Burdaki en büyük eksiklik artık 3.katmandaki yönlendirme özelliğinin kullanılamaz olmasıdır [6].

Profinet, gerçek zamanlı iletişim için optimize edilmiş bir iletişim kanalı kullanır. Bu kanal ile istasyonlar arasındaki veri iletim süresinin önceden tanımlanmış sınırlar içinde olması garanti edilmiş olur. Gerçek zamanlı kanal standart Ethernet cihazları üzerinde özel bir yazılım yardımı ile veya özel tasarlanmış bir cihaz ile kurulabilir. Bu kanal ISO/OSI referans modelinin 2.katmanına dayanarak oluşturulur. Veri paketlerinin adreslenmesi artık IP adreslerine göre değil MAC adreslerine göre yapılır. Gerçek zamanlı olması için optimize edilmiş sistem üzerinde aynı anda TCP/IP gibi standart protokoller de problemsiz olarak çalıştırılabilir (Şekil 4.4).



Şekil 4.4. Profinet Katmanlı Mimarisi

Gerçek zamanlı protokol Şekil 4.4'te görüldüğü üzere 3 sınıfa (class) ayrılmıştır:

Gerçek zaman sınıf 1 (real-time class 1): Çevrimsel verinin iletimi için kullanılır. Özel bir yazılım ve donanım ihtiyacı yoktur.
Gerçek zaman sınıf 2 (real-time class 2): Çevrimsel verinin ve kesmelerin (interrupt) iletimi için kullanılır. Özel anahtarların kullanımı gerekmektedir.
Gerçek zaman sınıf 3 (IRT): Hareket kontrollü uygulamalarda çevrimsel verinin iletimi için kullanılır. Özel bir iletişim planlaması ve özel anahtarların kullanılması gerekmektedir.

Profinet üzerinde iletilen ve gerçek zamanlı olmasına ihtiyac olmayan veriler (NRT) standart kanal üzerinden iletilir [6].

4.2. Profinet İletişim Profilleri ve Protokolleri

Profinet otomasyon uygulamalarının gereksinimleri için Profinet CBA ve Profinet IO olmak üzere 2 farklı iletişim profili sunar. Profinet IO dağıtık olarak bulunan veri giriş çıkış cihazlarının entegrasyonu amacıyla kullanılırken, Profinet CBA sahada dağıtık olarak çalışan otomasyonda modüler noktalar kurulmasına imkan sağlar [6].

4.2.1. Profinet CBA (Komponent Tabanlı Otomasyon)

Profinet CBA'nın temel amacı, otomasyon sahasının değişik şekillerde alt modüllere bölünmesi ve bu modüllerin birbirleriyle dağıtık olarak haberleşmesidir. Bu alt modüller birbirleriyle aynı ve birbirlerinden farklı formlarda olabilirler.

Modüller uzaktan kontrol edilebilen değişken sayıda giriş sinyali tarafından kontrol edilirler. Kullanıcı tarafından geliştirilen yazılımdaki fonksiyonlara ve bu fonksiyonların yerine getirdiği işlevlere sahiptirler. Modüllerin ürettiği çıkış sinyalleri başka bir modül tarafından kullanılır [6].

Modüller arası iletim çevrimi 10 ms seviyesindedir ve bu seviye kontrolörler arasında oldukça elverişli bir iletim imkanı sunmaktadır. Modülerliğin anlamı bir görevin birden fazla alt göreve bölünüp, bu görevlerin değişik modüller aracılığıyla gerçekleştirilebilmesidir. İdeal durumda bir problemi çözmek için hazırlanan modül daha sonra başka bir problem içinde kullanılabilmelidir.

Profinet CBA standart DCOM ve RPC teknolojileri üzerine kurulmuştur. Profinet CBA bir modülün üzerine düşen erişim, konfigürasyon ve kontrol gibi görevleri yerine getirmesinde geleneksel Ethernet donanım ve yazılımlarını kullanır.

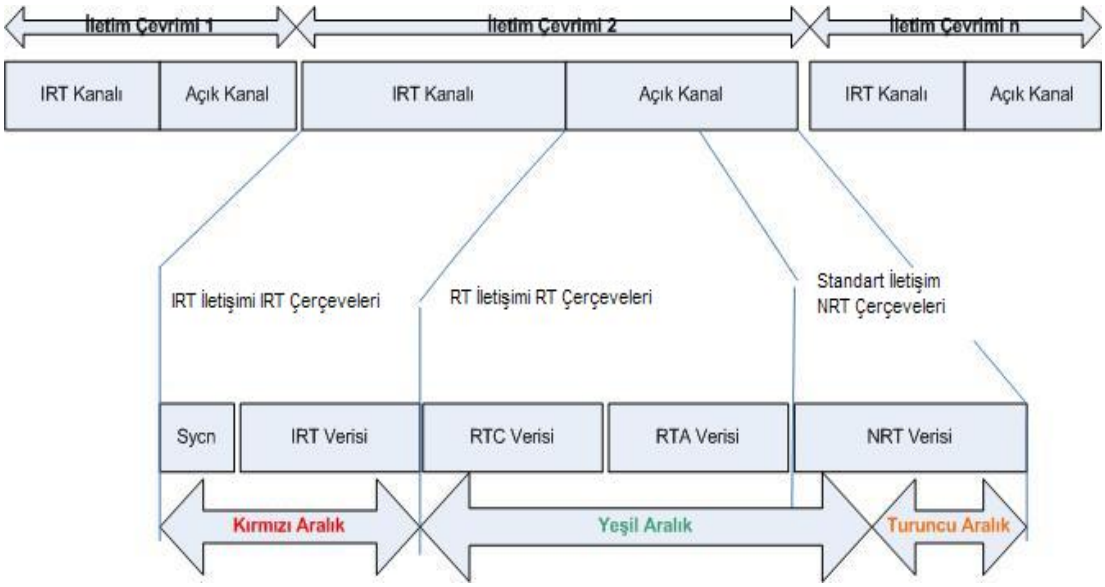
Profinet CBA bir otomasyon sistemini "Teknolojik modüllerin" bir birleşimi olarak görür. Bütün modüller birbirlerinden bağımsız olarak görevlerini koordine ederek

bütünleşik otomasyon sistemini oluştururlar. DCOM böyle bir sistem için uygun bir çözüm sunar. Nesneye dayalı olarak çalışan bu sistemde istemciler belirli bir veri kaynağından (sunucu) veri transfer ederek görevlerini yerine getirirler. Bu iletişim için çalışma zamanında TCP/IP üzerinde COM, DCOM modelleri ile RPC yapısı kullanılur.

4.2.2. Profinet IO

Profinet IO, Ethernet tabanlı saha seviyesi cihazları arasında yüksek hızlı veri iletişimi amacıyla tasarlanmaştır. Kontrolörler ve cihazlar arasında veri iletişimi çalışma parametrelerinin ve çalışma sırasında oluşan mesajların iletimini tanımlar. Profinet IO üretici/tüketici (producer/consumer) iletişim modeline göre çalışır.

Profinet spesifikasyonu hem gerçek zamanlı hem de gerçek zamanlı olmayan iletişimi tanımlar. Gerçek zamanlı iletişim ikiye ayrılır. İlki, öncelik tabanlı gerçek zamanlı iletişim olan RT (RT Class 1)'dir. İkincisi ise, saat senkronizasyonu ile çalışan isenkron gerçek zamandır (IRT, RT Class 2, RT Class 3). Spesifikasyon gerçek zamanlı olmayan iletişim için herhangi bir kısıtlama getirmez. Şekil 4.5'te görüldüğü gibi gerçek zamanlı trafik iletildikten sonra geriye kalan band genişliği gerçek zamanlı olmayan (NRT) iletişim için kullanılabilir (ör: TCP/IP vb.) [9].



Şekil 4.5. Profinet Kanalları İletim Çevrimi

Kırmızı Aralık: IRT çerçevelerinin iletimi amacıyla kullanılır. Sistemdeki istasyon sayısına ve çevrimsel verinin kalitesine göre boyutu değişir. Zaman açısından kritik olmayan veriler ASIC üzerinde yeşil aralığın başlangıç zamanına kadar tamponda saklanır.

Yeşil Aralık: RT çerçevelerinin ve 802.1Q'ya göre öncelik değeri verilmiş NRT çerçevelerinin iletimi amacıyla kullanılır. NRT çerçevelerinin iletim süresi turuncu aralığa geçmemelidir.

Turuncu Aralık: Sadece NRT çerçevelerinin iletimi amacıyla kullanılır. Bu çerçeveler çevrim sonunda yok edilir. Aralığın boyutu en az bir maksimum uzunluktaki Ethernet çerçevesinin iletimine imkan sağlayacak kadar olmalıdır.

4.2.2.1. Profinet IRT

Profinet IRT, Profinet'in saat senkronizasyonu ile çalışan öncelik tabanlı gerçek zamanlı iletişimini ifade eder. IRT'nin ihtiyaç duyduğu saat senkronizasyonu PTCP ile sağlanır.

IRT (Isochronous Real-Time Communication)

Yüksek performansa ve gerçek zamanlı çalışmaya ihtiyaç duyan hareket kontrolü gibi uygulamalar için Profinet'in IRT çözümü kullanılır [6].

Profinet IRT ile çok düşük gecikme ile (cycle time < 1ms ve jitter < 1 mikrosaniye) uygulamalar çalıştırılabilir. Bunun için özel bir cihaza (ASIC) ihtiyaç duyulmaktadır.

Profinet IRT, Şekil 4.6'te verilen çerçeve yapısına sahiptir. Bu çerçeve yapısında 0x8892 değerli Ethernet tip alanı ve 0x100-0x7FF değerli FrameID alanı belirleyici alanlardır.

Preamble 7 Byte	SFD 1 Byte	Hedef Adresi 6 Byte	Kaynak Adresi 6 Byte	EtherType 2 Byte	Frame ID 2 Byte	IRT Verisi 36-1490 Byte	FCS 4 Byte

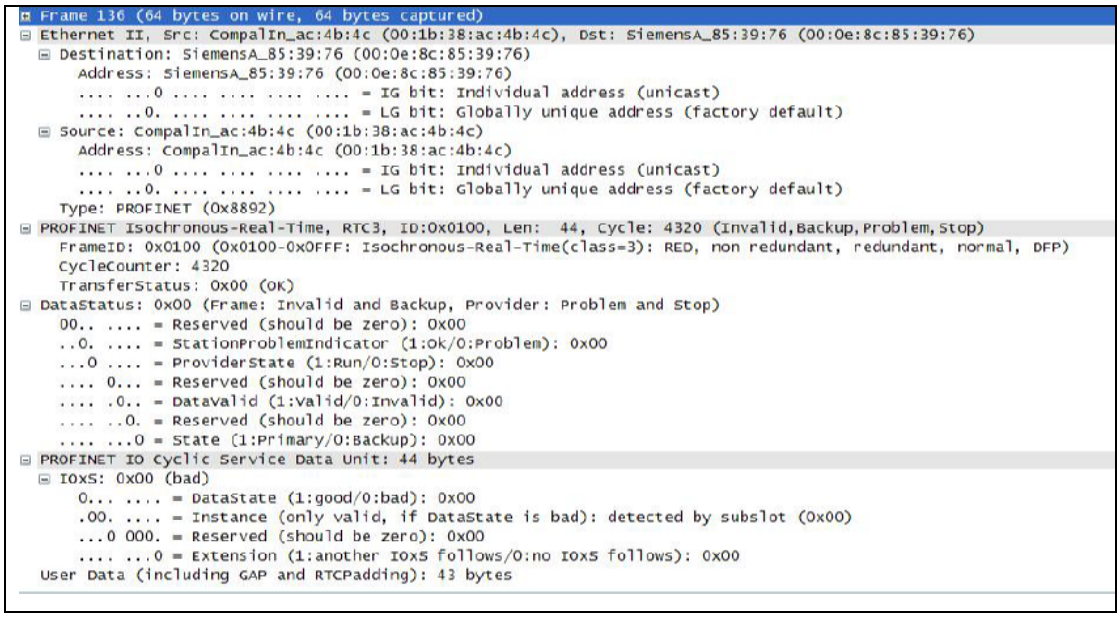
Şekil 4.6. IRT Çerçeve Yapısı

Tablo 4.1. IRT Protokol Tablo 4.1'te, PROFINET IRT çerçeve yapısındaki alanların detayları verilmiştir.

Tablo 4.1. IRT Protokol Alanları

Protokol Alanı	**Açıklama**
Preamble (7 Byte)	Paketin başladığını belirtir. Alıcının senkronizasyonu sağlar.
SFD (1 Byte)	Çerçeve içeriğinin başlangıcını gösterir. Değeri (10101011)'dir. En sondaki iki adet 1 biti Kaynak Adresinin başlangıcını gösterir.
Hedef Adresi (6 Byte)	Veri paketinin hedef adresini gösterir. 6 Byte'lık verinin ilk 3 Byte'ı üretici firmayı belirtir. Siemens: "08.00.06…"
Kaynak Adresi (6 Byte)	Veri paketinin kaynak adresi
Ethertype (2 Byte)	Veri kısmının ait olduğu ağ protokolünü belirtir. 0x8892: Profinet
Frame ID (2 Byte)	IRT çerçeve tipini belirler. 0x0100-0x07FF: RealTime class 3 frame, cyclic
IRT Verisi	Asenkron Gerçek zamanlı veri. Veri uzunluğu: 36-1490 bayt
FCS (4 Byte)	Çerçeve Kontrol verisini gösterir. 32-bit checksum. Bütün bir Ethernet çerçevesi için CRC.

Wireshark, PROFINET IRT çerçevelerini yakalamakta ve çerçeve içerisindeki alanları ayrıştırıp şekilsel olarak gösterebilmektedir (Şekil 4.7).



Şekil 4.7. IRT Çerçeve Örneği

PTCP (Presicion Transparent Clock Protocol)

PTCP, ağ haberleşmesi sırasında ağın senkronizasyonuna ihtiyaç duyulduğu zaman kullanılır. Profinet'de IRT ile birlikte kullanılır. İletim linkinin bütün zaman parametreleri PTCP ile kaydedilir.

PTCP, OSI referans modelinde 2.katmanda bulunur ve bu yüzden yönlendirme yeteneği bulunmamaktadır. PTCP master ve slave'leri arasında mikrosaniyeler mertebesinde zaman senkronizasyonu yapılmasına imkan tanımaktadır [6].

PTCP'nin en önemli özellikleri [6]:

- Mikrosaniye ve daha düşük seviyelerde senkronizasyon,
- Düşük kaynak kullanımı,
- Ağ elemanlarının işlemci ve hafıza kullanımları için yük getirmemesi,
- Minimum band genişliği kullanımı,
- Düşük yönetici (administration) ihtiyacı

PTCP, Şekil 4.8'te verilen çerçeve yapısına sahiptir. Bu çerçeve yapısında protokol elemanına bağlı olarak Hedef Adresi, Ethernet Tip alanı ve Frame ID alanı belirleyici alanlardır.

Preamble 7 Byte	SFD 1 Byte	Hedef Adresi 6 Byte	Kaynak Adresi 6 Byte	EtherType 2 Byte	VLAN TPID 2 Byte	EtherType 2 Byte	Frame ID 2 Byte	PTCP Başlık 16 Byte	PTCP Data	FCS 4 Byte

Şekil 4.8. PTCP Çerçeve Yapısı

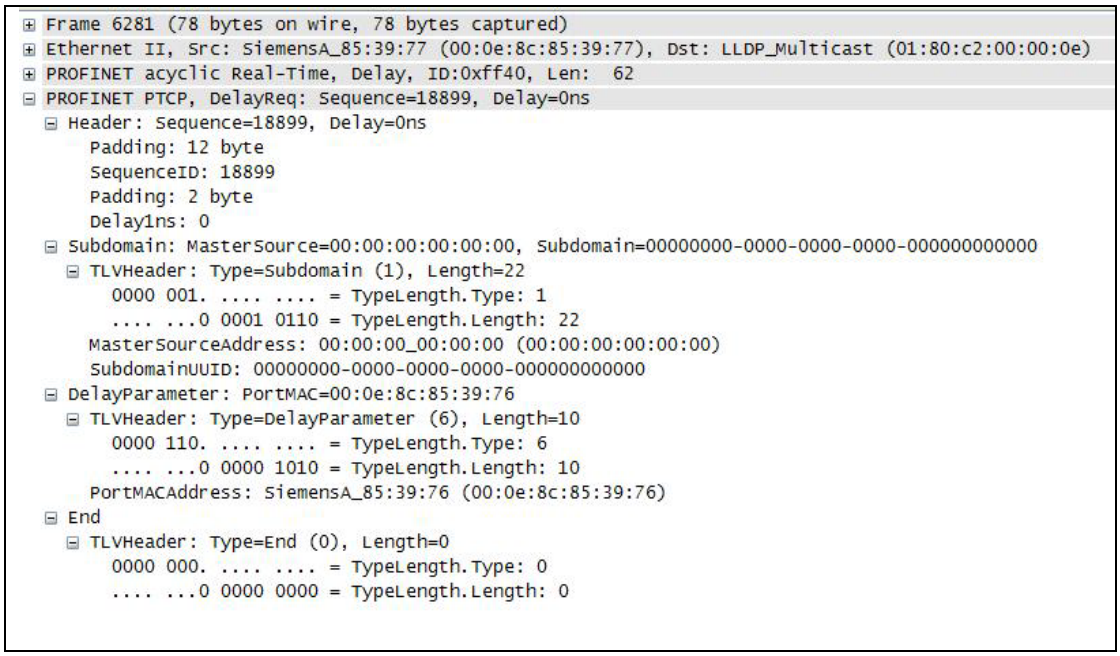
Tablo 4.2'te, PTCP çerçeve yapısındaki alanların detayları verilmiştir.

Tablo 4.2. PTCP Protokol Alanları

Protokol Alanı	**Açıklama**
Preamble (7 Byte)	Paketin başladığını belirtir. Alıcının senkronizasyonu sağlar.
SFD (1 Byte)	Çerçeve içeriğinin başlangıcını gösterir. Değeri (10101011)'dir. En sondaki iki adet 1 biti Kaynak Adresinin başlangıcını gösterir.
Hedef Adresi (6 Byte)	**Delay Req:** 01.80.C2.00.00.0E (Frame ID: 0xFF40) **Delay Res:** 01.80.C2.00.00.0E (Frame ID: 0xFF41) **FollowUp Res:** 01.80.C2.00.00.0E (Frame ID: 0xFF42) **Acyclic RT (RTA):** **Sync**: 01.0E.CF.00.04.00 (Frame ID: 0x0000)- 01.0E.CF.00.04.1F (Frame ID: 0x001F) **Follow Up:** 01.0E.CF.00.04.20 (Frame ID: 0xFF20)- 01.0E.CF.00.04.1F (Frame ID: 0xFF1F) **Cyclic RT (RTC):** **Sync:** 01.0E.CD.00.01.02 (Frame ID: 0x0080) **Reserved:** 01.0E.CF.00.00.00-01.0E.CF.00.01.01 01.0E.CF.00.01.03-01.0E.CF.00.03.FF 01.0E.CF.00.04.40-01.0E.CF.FF.FF.FF
Kaynak Adresi (6 Byte)	**Sync, Follow Up:** PTCP Master'ın Kaynak veya Gönderen Adresi **DelReq, DelRes:** Kaynak veya Gönderen Adresi
EtherType (2 Byte)	Veri paketinin uzunluğunu veya tip bilgisini belirtir. Ethertype < 0x0600: IEEE802.3 uzunluk bloğu Ethertype = 0x0600 Ethertype II tip bloğu Ethertype = 0x8100 Veri paketinin VLAN TPID alanına sahip olduğunu belirtir.
VLAN TPID (2 Byte)	VLAN tag protokolünü belirtir.
Öncelik Değeri (User Priority) (3 Bit)	Veri Paketinin önceliğini gösterir. 0x00-0x05: Reserved 0x06: Cyclic RT Sync Frame 0x07: Acyclic RT Sync Frame Acyclic RT FollowUp Frame Acyclic RT DelayReq Frame Acyclic RT DelayRes Frame Acyclic RT DelayFollowUpRes Frame
CFI (1 Bit)	0: Ethernet 1: Token Ring
VLAN ID (12 Bit)	VLAN tanımlamasını belirtir. 0x000: Öncelik değeri olan verinin iletildiğini belirtir. 0x001: Standart Ayar 0x002-0xFFE: Kullanım için serbest 0xFFF: Reserved
Ethertype (2 Byte)	Veri kısmının ait olduğu ağ protokolünü belirtir.

	0x8892: Profinet
Frame ID (2 Byte)	RT çerçeve tipini belirler. PTCP Senkronizasyonu 0x0000-0x001F: Acyclic RT Sync Frame 0x0080: Cyclic RT Sync Frame 0xFF00: Acyclic RT Sync Frame(cycle) 0xFF01: Acyclic RT Sync Frame(time) 0xFF20: Acyclic RT FollowUp Frame(cycle) 0xFF21: Acyclic RT FollowUp Frame(time) 0xFF22-0xFF3F: Acyclic RT FollowUp Frame 0xFF40: Acyclic RT DelayReq Frame 0xFF41: Acyclic RT DelayRes Frame 0xFF42: Acyclic RT FollowUpRes Frame 0x0020-0x007F: Reserved 0x0081-0xFEFF: Reserved 0xFF02-0xFF1F: Reserved 0xFF43-0xFFFF: Reserved
PTCP Header (16 Byte)	
Reserved_1	Unsigned32 value
Reserved_2	Unsigned32 value
Delay 10ns	Sync, FollowUp, DelayFollowUpRes: 10-ns bekleme zamanı DelayReq, DelayRes: 0x00000000
SequenceID	0x0000-0xFFFF: Sequence Numarası
Delay Ins	Sync, FollowUp, DelayFollowUpRes: 1-ns bekleme zamanı(0…9) DelayReq, DelayRes: 0x00
Gap	0x00
PCTP Data	PTCP çerçevesinin tipinin yapısına göre PTCP verisi
FCS (4 Byte)	Çerçeve Kontrol verisini gösterir. 32-bit checksum. Bütün bir Ethernet çerçevesi için CRC.

Wireshark, PTCP çerçevelerini yakalamakta ve çerçeve içerisindeki alanları ayrıştırıp şekilsel olarak gösterebilmektedir (Şekil 4.9).



Şekil 4.9. PTCP Çerçeve Örneği

4.2.2.2. Profinet RT

RT (Real-Time)

RT kanalı, çevrimsel (cyclic) veri transferi ve alarmların iletimi için kullanılır. RT haberleşmesi standart TCP/IP arayüzününün yerine geçen özel protokol yapısına sahiptir. Bu arayüz sayesinde klasik Ethernet'in veri gecikmesinin önüne geçilmiş olur (cycle times > 10ms).

Jitter süresinin kısa tutulması amacıyla çerçeveler içinde öncelik değeri bulunmaktadır. Bu öncelik yapısı VLAN çerçeve formatı ile sağlanmaktadır. Öncelik değerleri 0 (düşük öncelik) ile 7 (yüksek öncelik) arasında verilmektedir. Profinet RT çerçevelerinin öncelik değerleri 6 veya 7'dir [6].

IEEE 802.1Q'ya göre VLAN etiketi Ethernet çerçevesini, kaynak adres ve veri bölümleri arasında olacak şekilde, 2 tane 2 bayt olmak üzere toplam 4 bayt büyütür. Ethernet Tipi:0x8100 VLAN etiketinin varlığını göstermektedir.

Profinet RT bazı gerçek zamanlı uygulamalarda kullanılmak üzere tasarlanmıştır, veri haberleşmesini birkaç milisaniye mertebesine ve jitter değerini %15 seviyesine çekebilmektedir. VLAN çerçeve yapsını kullanarak Ethernet çerçevelerinin öncelik sırasına göre gönderilmesini sağlayarak daha kısa iletişim sürelerinin elde edilmesi sağlanabilmektedir [9].

Profinet RT için tip alanı değeri 0x8892'dir. Profinet çerçevesi için diğer bir önemli alan ise FrameID alanıdır. FrameID alanı birbiriyle haberleşen iki cihaz arasındaki iletişim kanalının adreslenmesi amacıyla kullanılır. RT cihazları arasındaki iletişim uygulama katmanı protokolleri ile sağlanır. Diğer bir alan ise çevrim sayacıdır (cycle counter). Çevrim sayacı eski paketlerin tespit edilmesi amacıyla kullanılır. 16 bit uzunluğundadır ve her 31.25µs'de değeri 1 artar. Kayıp çerçevelerin olup olmadığının anlaşılması amacıyla da kullanılabilir. Profinet RT anahtarlar Depola-Gönder (Store & Forward) yöntemini kullanırlar [9].

Her bir döngüde öncelikle çevrimsel RT çerçeveleri (cyclic RT Frames), ardından çevrimsel olmayan RT çerçeveleri (acyclic RT Frames, RTA) ve kalan sürede de gerçek zamanlı olmayan RT çerçeveleri (NRT) gönderilir. RT çerçevelerinin uzunluğu her bir çevrimde band genişliğinin %50'sini aşmamalıdır.

Profinet RT, Şekil 4.10'te verilen çerçeve yapısına sahiptir. Bu çerçeve yapısında 0x8100 değerli Ethernet tip alanı, 0x05-0x06 değerli VLAN öncelik (priority) alanı, 0x8892 değerli VLAN Tip alanı FrameID alanı belirleyici alanlardır.

Preamble 7 Byte	SFD 1 Byte	Hedef Adresi 6 Byte	Kaynak Adresi 6 Byte	EtherTyp e 2 Byte	VLAN TPID 2 Byte	EtherType 2 Byte	Frame ID 2 Byte	RT Verisi 40-1440 Byte	APDU Status 4 Byte	FCS 4 Byte

Şekil 4.10. RT Çerçeve Yapısı

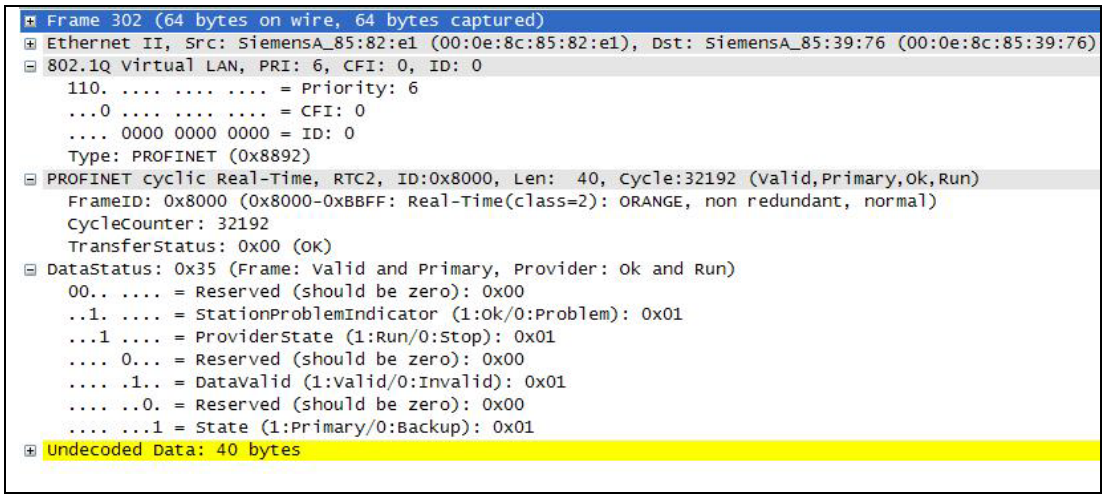
Tablo 4.3'te, RT çerçeve yapısındaki alanların detayları verilmiştir.

Tablo 4.3. RT Protokol Alanları

Protokol Alanı	**Açıklama**
Preamble (7 Byte)	Paketin başladığını belirtir. Alıcının senkronizasyonu sağlar.
SFD (1 Byte)	Çerçeve içeriğinin başlangıcını gösterir. Değeri (10101011)'dir. En sondaki iki adet 1 biti Kaynak Adresinin başlangıcını gösterir.
Hedef Adresi (6 Byte)	Veri paketinin hedef adresini gösterir. 6 Byte'lık verinin ilk 3 Byte'ı üretici firmayı belirtir. Siemens: "08.00.06…"
Kaynak Adresi (6 Byte)	Veri paketinin kaynak adresi
EtherType (2 Byte)	Veri paketinin uzunluğunu veya tip bilgisini belirtir. Ethertype < 0x0600: IEEE802.3 uzunluk bloğu Ethertype = 0x0600 Ethertype II tip bloğu Ethertype = 0x8100 Veri paketinin VLAN TPID alanına sahip olduğunu belirtir.
VLAN TPID (2 Byte)	VLAN tag protokolünü belirtir.
Öncelik Değeri (User Priority) (3 Bit)	Veri Paketinin önceliğini gösterir. 0x00: IP (RPC) 0x01-0x04: Reserved 0x05: Acyclic RT Data Low Acyclic RT UDP Frame Low 0x06: Cyclic RT Data Acyclic RT Data high Cyclic RT UDP frame Acyclic RT UDP frame high 0x07: Reserved
CFI (1 Bit)	0: Ethernet 1: Token Ring
VLAN ID (12 Bit)	VLAN tanımlamasını belirtir. 0x000: Öncelik değeri olan verinin iletildiğini belirtir. 0x001: Standart Ayar 0x002-0xFFE: Kullanım için serbest 0xFFF: Reserved
Ethertype (2 Byte)	Veri kısmının ait olduğu ağ protokolünü belirtir. 0x8892: Profinet
Frame ID (2 Byte)	RT çerçeve tipini belirler. 0x0000-0x00FF: Reserved 0x0100-0x7FFF: Reserved 0x8000-0xBEFF: RT Class 2 unicast(RT)

	0xBF00-0xBFFF:	RT Class 2 multicast(RT)
	0xC000-0xFAFF:	RT Class 1 unicast(RT and RT OverUPD)
	0xFB00-0xFBFF:	RT Class 1 multicast(RT and RT OverUDP)
	0xFC00:	Reserved
	0xFC01:	Alarm High(RT and RTOverUDP)
	0xFC02-0xFE00:	Reserved
	0xFE01:	Alarm low(RT and RTOverUDP)
	0xFE02-0xFFFF:	Reserved
RT Verisi	Gerçek zamanlı veri.	
	Profinet CBA:	Byte akışı ile aynı QoS değerine sahip iletişim verisi.
	Profinet IO:	I/O verisi
APDU Status	Uygulama protokolu veri durumunu gösterir. Gerçek zamanlı veri çerçevesinin durumunu belirtir.	
Cycle Counter	Döngü sayacı her iletim döngüsünde arttırılır. Tüketici uzun süren işlemleri sayacı kontrol ederek anlayabilir. 1 bitlik artış gerçek zamanda 31.25µs'lik bir artışa denk gelir.	
Veri Durumu	Bit0: 0: İkincil. Gereksiz durumda bulunan ikincil kanal tanımlanır. 1: Birincil. Gereksiz durumda bulunan birincil kanal tanımlanır. Bit1: 0 Bit2: 1: Veri geçerli 0: Veri geçersiz Bit3: Kullanılmıyor. Bit4: 0: Veriyi üreten süreç aktif değil. 1: Veriyi üreten süreç aktif. Bit5: 0: Problem var.Sistemde uyarı mevcut. 1: Bilinen bir problem yok Bit6: 0 Bit7: 0	
Transfer Durumu	Bit0-7: 0	
FCS (4 Byte)	Çerçeve Kontrol verisini gösterir. 32-bit checksum. Bütün bir Ethernet çerçevesi için CRC.	

Wireshark, RT çerçevelerini yakalamakta ve çerçeve içerisindeki alanları ayrıştırıp şekilsel olarak gösterebilmektedir (Şekil 4.11).



Şekil 4.11. RT Çerçeve Örneği

LLDP (Link Layer Discovery Protocol)

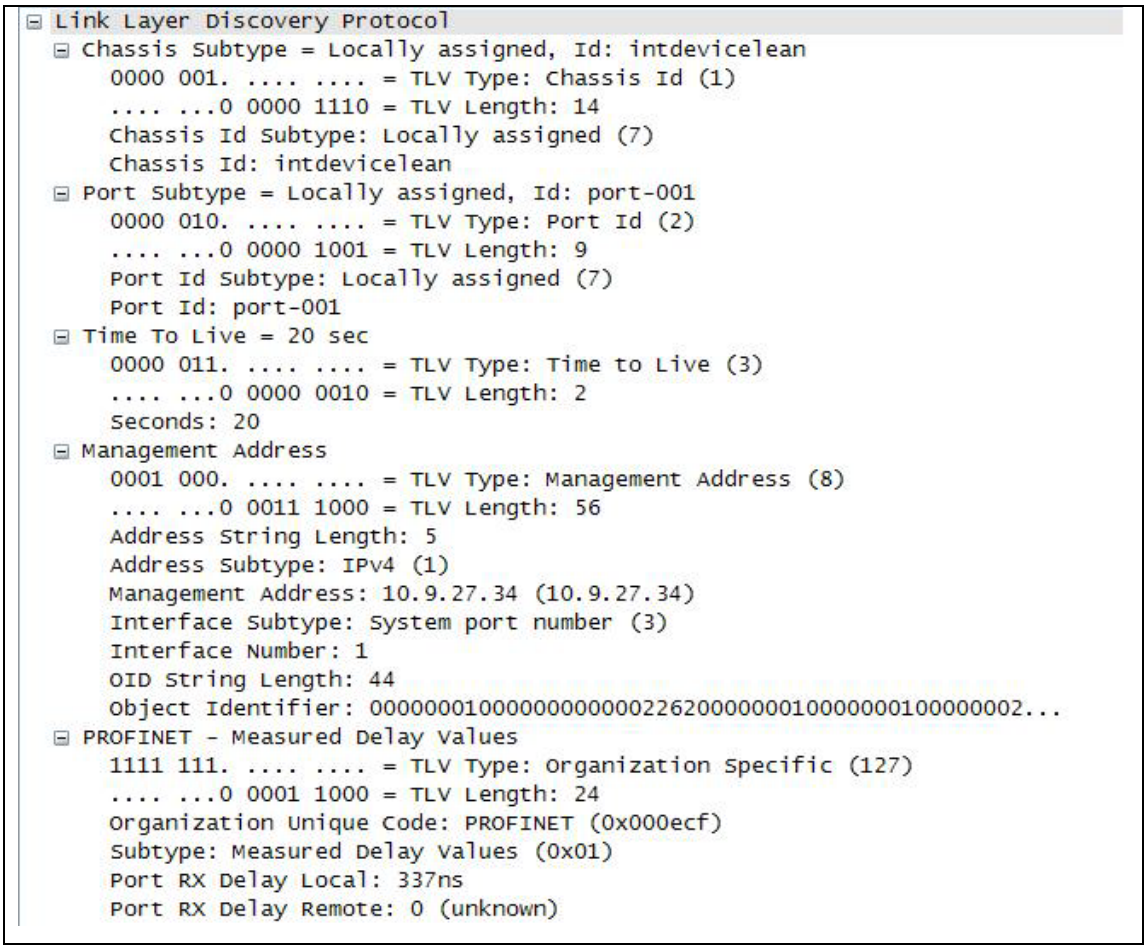
2.katman protokolüdür. Komşu cihazlar arasında veri iletimi amacıyla kullanılır. Periyodik olarak komşu cihazlarının bilgi gönderip alması amacıyla kullanılır. Hedef adresi: 01.80.C2.00.00.0E ve Ethernet tipi: 0x88CC'dir. Tek yönlü protokol olarak çalışır, gönderme ve alma işlemleri birbirinden bağımsız olarak yapılır.

LLDP, Şekil 4.12'te verilen çerçeve yapısına sahiptir. Bu çerçeve yapısında 01.80.C2.00.00.0E değerli Hedef Adresi ve 0x88CC değerli Ethernet Tip alanı belirleyici alanlardır.

Preamble	SFD	Hedef Adresi	Kaynak Adresi	EtherType	RT Verisi
7 Byte	1 Byte	6 Byte	6 Byte	2 Byte	40-1440 Byte

Şekil 4.12. LLDP Çerçeve Yapısı

Wireshark, LLDP çerçevelerini yakalamakta ve çerçeve içerisindeki alanları ayrıştırıp şekilsel olarak gösterebilmektedir (Şekil 4.13).



Şekil 4.13. LLDP Çerçeve Örneği

4.2.2.3. Profinet NRT

DCP (Dynamic Configuration Protocol)

DCP, Profinet ağındaki cihazların kendilerini tanıtmak ve sistem konfigürasyonun kurulması amacıyla kullanır. DCP ile ağdaki cihazların,

- İstasyon Adı,
- Cihaz Rolü,
- Üreticiye özel istasyon tipi,

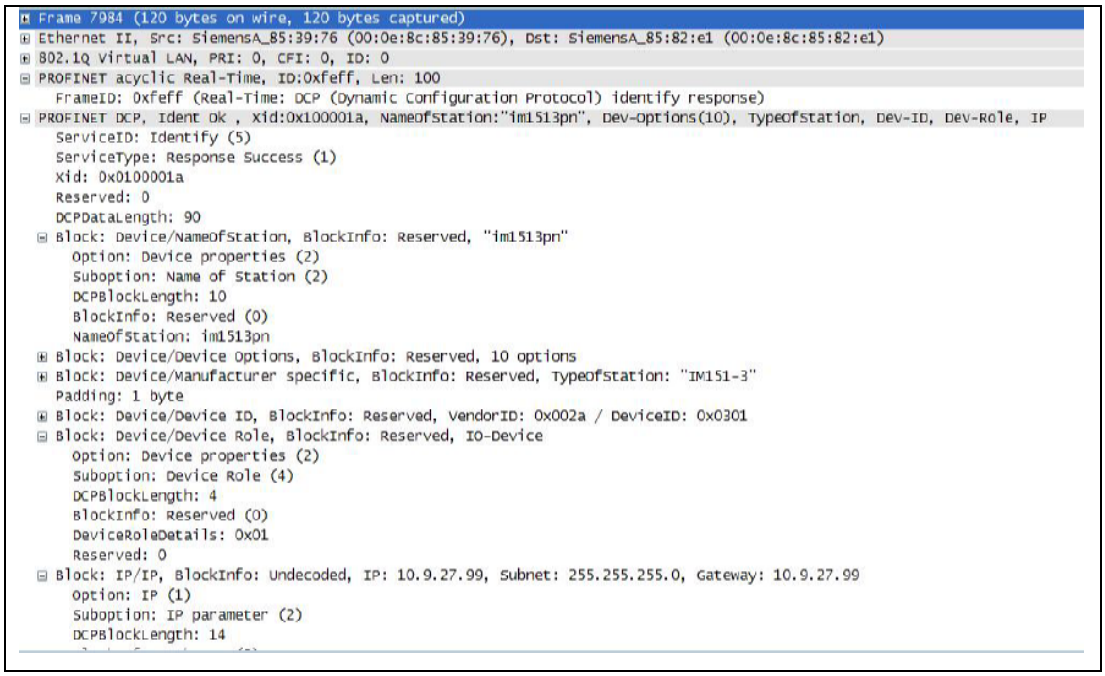
- CihazID,
- SağlayıcıID,
- IP, MAC, Subnet Mask ve Router adresleri elde edilebilir.

DCP, Şekil 4.14'te verilen çerçeve yapısına sahiptir. Bu çerçeve yapısında 0x8100 değerli Ethernet Tip alanı, 0x00 değerli VLAN öncelik alanı, 0x8892 değerli VLAN Tip alanı ve 0xFEFD-0xFEFF değerlerine sahip FrameID alanı belirleyici alanlardır.

Preamble 7 Byte	SFD 1 Byte	Hedef Adresi 6 Byte	Kaynak Adresi 6 Byte	EtherType 2 Byte	VLAN TPID 2 Byte	EtherType 2 Byte	Frame ID 2 Byte	DCP Verisi 40-1440 Byte

Şekil 4.14. DCP Çerçeve Yapısı

Wireshark, DCP çerçevelerini yakalamakta ve çerçeve içerisindeki alanları ayrıştırıp şekilsel olarak gösterebilmektedir (Şekil 4.15).



Şekil 4.15. DCP Çerçeve Örneği

NRT (Non Real-Time)

Gerçek zamanlı olmayan uygulamalar için kullanılır. Standart OSI referans modelindeki IP, UDP, DHCP, DNS, SNMP, ICMP, ARP protokollerini ifade eder (Şekil 4.16). Parametre değişimi, konfigürasyon ayarları, döngüsel olmayan yazma okuma uygulamaları için kullanılır (cycle times > 100 ms).

Yukarıdaki standart protokollere ek olarak RT çerçeve formatında öncelik değerinin 0 olması da çerçevenin NRT'ye ait olduğunu ifade eder.

Preamble 7 Byte	SFD 1 Byte	Hedef Adresi 6 Byte	Kaynak Adresi 6 Byte	EtherType 2 Byte	Ethernet Verisi 46-1500 Byte	CRC 32 4 Byte

Şekil 4.16. Standart Ethernet Çerçevesi

BÖLÜM 5. PROFINET IO AĞ ÇÖZÜMLEYİCİ GERÇEKLEMESİ

Bu tez çalışması kapsamında temel olarak 4 uygulama üzerinde çalışılmıştır. Uygulamalar 2 ayrı ana başlık altında toplanmıştır.

İlk grup, Profinet IO Ağ Çözümleyici uygulamalarıdır. Profinet IO Ağ Çözümleyici uygulamaları, Profinet IO ağı üzerindeki Profinet çerçevelerini paket koklama yöntemlerine göre toplayarak, tez çalışması kapsamında geliştirilen Profinet IO Ağ Çözümleyici algoritmasına göre sınıflandırırlar. Bu sınıflandırma esnasında Profinet DCP protokolüne ait çerçevelerin ayrıca ayrıntılı olarak çözümlenmesi ile ağ üzerindeki cihazların oluşturduğu ağ topolojisi üretilir ve görsel olarak sergilenir.

İkinci grup uygulamalar ise, Profinet IO Ağ Oynatıcı (Player) başlığı altında 6.bölümde incelenmiştir. Profinet IO Ağ Oynatıcı uygulamasında, ağ koklamaya göre sistemdeki çerçeveler ters yönde hareket ederler (bkz. Şekil 5.2 ve Şekil 6.1). Ağ çözümleyici uygulamalarında Profinet çerçeveleri ağdan alınırken, ağ oynatıcı uygulamarında görsel bir arayüzden oluşturulan veya daha önce ağdan alınarak bir dosyaya kaydedilmiş çerçeveler ağa gönderilir.

Uygulamaların oluşturulmasında Java programlama dilinin 1.6.0_17 versiyonu kullanılmıştır. Paket yakalama ve gönderme işlemleri için jNetPcap 1.3.a1-1 versiyonlu Java programlama kütüphanesinden yararlanılmıştır. Görsel arayüz ve programlama editörü olarak ise Java NetBeans IDE 6.8 kullanılmıştır.

Bu 4 uygulamanın birlikte kullanılması ile Profinet çerçevelerinin gönderildiği bir ağ üzerinde Profinet IO protokollerine ait çerçevelerin çözümlenmesi yapılabilir, ağ üzerinde Profinet IO protokollerini kullanan cihazların topolojisi ve çeşitli ağ

parametreleri sergilenebilir, ağa parametreleri kullanıcı tarafından tanımlanan veya daha önce kaydedilmiş çerçeveler tekrar gönderilebilir.

5.1. Profinet IO Ağ Çözümleyici Tasarımı

Profinet IO iletişim profilinin sahip olduğu birçok protokol vardır. Bu tez çalışmasında gerçeklediğimiz Ağ Çözümleyici uygulamamızın bu protokolleri anlayabilmesi için öncelikle çerçeve yapılarına bağlı olarak özet bir tablo oluşturulmuştur (Tablo 5.1). Bu tablo üzerinden de bir algoritma (Şekil 5.1) geliştirilerek programlama aşamasına geçilmiştir.

Tez kapsamında yapılan 4 uygulama da piyasada benzerleri olan uygulamaların teze uygun olarak Profinet IO çerçevelerine göre özelleştirilmiş uygulamalardır. Bu uygulamalarda Profinet'in desteklediği protokoller kullanılmıştır.

5.1.1. Profinet IO protokol çözümleme ve sınıflandırma algoritması

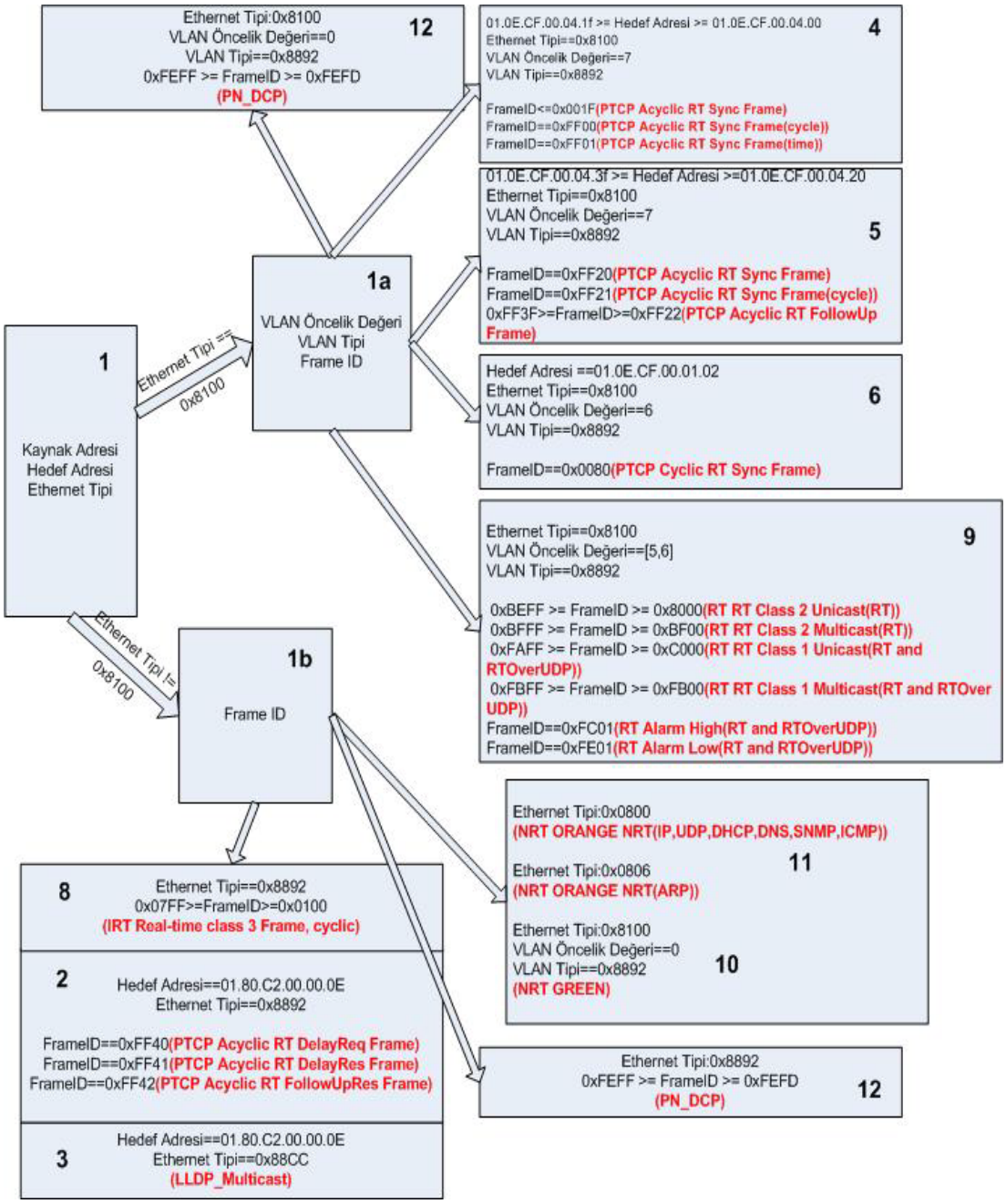
Profinet'in desteklediği protokoller ve çerçeve yapıları Tablo 5.1'te gösterildiği gibi özetlenebilir:

Tablo 5.1. Profinet IO Protokolleri ve Çerçeve Yapıları

Destination Address(6 BYTE)	**Ethertype(2 BYTE)**	**VLAN User Priority(3 Bit)**	**VLAN Ethernettype (2 BYTE)**	**Frame ID (2 BYTE)**	**Açıklama**
01.80.C2.00.00.0E	0x8892	NOT EXISTS	NOT EXISTS	0xFF40	Acyclic RT DelayReq Frame
01.80.C2.00.00.0E	0x8892	NOT EXISTS	NOT EXISTS	0xFF41	Acyclic RT DelayRes Frame
01.80.C2.00.00.0E	0x8892	NOT EXISTS	NOT EXISTS	0xFF42	Acyclic RT FollowUpRes Frame
01.0E.CF.00.04.00-01.0E.CF.00.04.1F	0x8100	0x07	0x8892	0x0000-0x001F	Acyclic RT Sync Frame
01.0E.CF.00.04.00-01.0E.CF.00.04.1F	0x8100	0x07	0x8892	0xFF00	Acyclic RT Sync Frame(cycle)
01.0E.CF.00.04.00-01.0E.CF.00.04.1F	0x8100	0x07	0x8892	0xFF01	Acyclic RT Sync Frame(time)
01.0E.CF.00.04.20-01.0E.CF.00.04.3F	0x8100	0x07	0x8892	0xFF20	Acyclic RT FollowUp Frame(cycle)
01.0E.CF.00.04.20-01.0E.CF.00.04.3F	0x8100	0x07	0x8892	0xFF21	Acyclic RT FollowUp Frame(time)
01.0E.CF.00.04.20-01.0E.CF.00.04.3F	0x8100	0x07	0x8892	0xFF22-0xFF3F	Acyclic RT FollowUp Frame
01.0E.CF.00.01.02	0x8100	0x06	0x8892	0x0080	Cyclic RT Sync Frame
Destination Addr.	0x8892	NOT EXISTS	NOT EXISTS	0x0100-0x7FFF	Real-time class 3 Frame, cyclic IRT
Destination Addr.	0x8100	0x05-0x06	0x8892	0x8000-0xBEFF	RT Class 2 Unicast(RT)
Destination Addr.	0x8100	0x05-0x06	0x8892	0xBF00-0xBFFF	RT Class 2 Multicast(RT)
Destination Addr.	0x8100	0x05-0x06	0x8892	0xC000-0xFAFF	RT Class 1 Unicast(RT and RTOverUDP)
Destination Addr.	0x8100	0x05-0x06	0x8892	0xFB00-0xFBFF	RT Class 1 Multicast(RT and RTOver UDP)
Destination Addr.	0x8100	0x05-0x06	0x8892	0xFC01	Alarm High(RT and RTOverUDP)
Destination Addr.	0x8100	0x05-0x06	0x8892	0xFE01	Alarm Low(RT and RTOverUDP)
Destination Addr.	0x8100	0x00	0x8892	-	NRT
Destination Addr.	0x0800	NOT EXISTS	NOT EXISTS	-	NRT(IP,UDP,DHCP,DNS,SNMP,ICMP)
Destination Addr.	0x0806	NOT EXISTS	NOT EXISTS	-	NRT(ARP)
01.80.C2.00.00.0E	0x88CC	NOT EXISTS	NOT EXISTS	-	LLDP_Multicast
Destination Addr.	0x8892	NOT EXISTS	NOT EXISTS	0xFEFE	PN_DCP

Destination Addr.	0x8100	0x00	0x8892	0xFEFE	PN_DCP

Yaptığımız çalışma ile yakalanan çerçevelerin PTCP, IRT, RT, NRT, LLDP veya DCP protokollerinden hangisine ait olduğu ve bu protokollerin içinde tanımlanan hangi işleve ait olduğu bulunmaya çalışılmıştır. Bu işlem sırasında Tablo 5.1 ve [6] referans olarak kullanılmıştır. Ağ üzerinden alınan çerçeveler byte dizisi (array) şeklindedir. Kullandığımız algoritma içinde bu byte dizileri String haline dönüştürülmüş ve çerçevelerin alanlarının kontrolleri String karşılaştırma yöntemiyle yapılmıştır. Aşağıda maddeler halinde yazılmış algoritma adımlarının numaraları Şekil 5.1 üzerindeki kutucukların numaralarına denk gelmektedir.



Şekil 5.1. Profinet IO Çerçeve Çözümleyici Algoritması

1. Öncelikle yakalanan çerçevenin aşağıdaki parametleri bulunur. Sınıflandırma için bu parametreler yeterlidir.

– Kaynak Adresi,

– Ethernet Tipi,

– Eğer Ethernet Tipi VLAN tag'ini gösteriyorsa (EthernetType=0x8100)

1a. VLAN öncelik değeri,

1a. VLAN Ethernet Tipi,.

- **1a** ve **1b**. Frame ID

2. Eğer Kaynak Adresi: 01.80.C2.00.00.0E ve EthernetTipi: 0x8892 ise aşağıdaki kriterlere göre protokol ve açıklaması bulunur.

- FrameID: 0xFF40 (PTCP Acyclic RT DelayReq Frame),
- FrameID: 0xFF41 (PTCP Acyclic RT DelayRes Frame),
- FrameID: 0xFF42 (PTCP Acyclic RT FollowUpRes Frame)

3. Eğer Kaynak Adresi: 01.80.C2.00.00.0E ve EthernetTipi: 0x88CC ise bu çerçeve LLDP_Multicast çerçevesidir.

4. Eğer Kaynak Adresi: 01.0E.CF.00.04.00-01.0E.CF.00.04.1F aralığında ve Ethernet Tipi: 0x8100 ve VLAN öncelik değeri: 7 ve VLAN Ethernet Tipi: 0x8892 ise bu çerçeve PTCP Acyclic RT Sync Frame çerçevesidir.

5. Eğer Kaynak Adresi: 01.0E.CF.00.04.20-01.0E.CF.00.04.3F aralığında ve Ethernet Tipi: 0x8100 ve VLAN öncelik değeri: 7 ve VLAN Ethernet Tipi: 0x8892 ise bu çerçeve PTCP Acyclic RT FollowUp Frame çerçevesidir.

6. Eğer Kaynak Adresi: 01.0E.CF.00.01.02 ve Ethernet Tipi: 0x8100 ve VLAN öncelik değeri: 6 ve VLAN Ethernet Tipi: 0x8892 ve FrameID değeri: 0x0080 ise bu çerçeve PTCP Cyclic RT Sync Frame çerçevesidir.

7. IRT, RT, NRT ve DCP protokollerine ait çerçeveler için kaynak adresi bir önem taşımaz.

8. Eğer Ethernet Tipi: 0x8892 ise ve Frame ID: 0x0100 ile 0x07FF aralığında ise bu çerçeve IRT Real-time class 3 Frame, cyclic çerçevesidir.

9. Eğer Ethernet Tipi: 0x8100 ise ve VLAN öncelik değeri: 5 veya 6 ise ve VLAN Ethernet Tipi: 0x8892 ise RT Class 1, RT Class 2, RT Alarm çerçevesidir.

10. Eğer Ethernet Tipi: 0x8100 ise ve VLAN öncelik değeri: 0 ise ve VLAN Ethernet Tipi: 0x8892 ise NRT Green çerçevesidir.

11. Eğer Ethernet Tipi: 0x0800 (IP) veya 0x0806 (ARP) ise NRT Orange çerçevesidir.

12. Eğer Ethernet Tipi: 0x8892 ve FrameID: 0xFEFD ile 0xFEFF aralığında ise bu çerçeve PN_DCP protokolune ait bir çerçevesidir.

5.2. Geliştirme Ortamı

Tez çalışması sırasında 4 uygulama da dahil olmak üzere Ethernet çerçeveleri ile ilgili geliştirilecek uygulamalar öncelikle işletim sistemi seviyesinde paket yakalama işlemini yapan hazır bir kütüphaneye ihtiyaç duyarlar. En yaygın olarak kullanılan Wireshark programı da çalıştığı ortama göre libpcap veya winpcap kütüphanesini kullanır.

Aşağıda Ethernet çerçeveleri ile ilgili yapılacak uygulamalarda bilinmesi gereken temel kavramlar listelenmiştir:
pcap (**P**acket **Ca**pture): Ağ üzerindeki trafiği yakalamak için kullanılan uygulama programı arayüzüdür.

libpcap: Unix tabanlı sistemlerde pcap arayüzünün gerçeklendiği kütüphanedir.

winpcap: Windows tabanlı sistemlerde libpcap'in Windows işletim sistemine uyarlanmış halidir.

Jpcap: Ağ üzerindeki paketleri yakalamak üzere geliştirilmiş java kütüphanesidir. Jpcap kullanarak ağ üzerinde paketler yakalanarak analiz edilir ve görüntülenir. Yakalanan paketler daha sonra inceleme amaçlı dosyalara kaydedilip, tekrar bu dosya ile ağ üzerinden geliyormuş gibi jpcap ile tekrar yakalanabilir. Jpcap kütüphanesi kendi içerisinde winpcap kütüphanesini kullanmaktadır. Bu kütüphanenin üzerine hazır protokol implementasyonları geliştirmiş ve Java programlama dili uygulama geliştiricelere bir arayüz sunar.

Örnek paket koklayıcı için gerekli olan yazılımlar ve kütüphaneler:

- Java programlama dili ve JavaSDK (1.6.0_17),
- NetBeans IDE 6.8
- Jpcap 0.01.16 (Java programlama dili için ağ paket yakalama kütüphanesi, http://sourceforge.net/projects/jpcap/)

– Jpcap ver.0.7 (Java programlama dili için paket oluşturma ve paket yakalama kütüphanesi, http://netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html)

– JEthereal, Netresearch Jpcap ile hazırlanmış, Wireshark benzeri ve java applet olarak da çalışabilen program http://yuba.stanford.edu/JEthereal/.

Tez çalışması sırasında oluşturduğumuz paket koklayıcı içinde basit anlamda bir paketin yakalanması için jNetPcap kütüphanesi ile aşağıdaki kod kullanılabilir:

```
// Sistemdeki ağ arayüzlerinin listesinin alınması için gerekli tanımlamalar.
private List<PcapIf> mDeviceList = new ArrayList<PcapIf>();
private StringBuilder errbuf = new StringBuilder();

Pcap.findAllDevs( mDeviceList, errbuf );



// Kullanıcı tarafından seçilen ağ arayüzü üzerinden Ethernet çerçevelerinin yakalanması
private Pcap mPcap = null;

int tSnapLen = 64 * 1024;
int tFlags = Pcap.MODE_PROMISCUOUS;
int tTimeOut = 10 * 1000;

mPcap = Pcap.openLive( mOpenDevice.getName(), tSnapLen, tFlags, tTimeOut, errbuf);

// Ağ üzerinde yakalanan paketlerin incelenmesi amacıyla PacketHandler tipinde sınıfın
// çağrılması
PacketHandler<String> tPacketHandler = new PacketHandler<String>();
mPcap.loop( -1, tPacketHandler, "" );

// PacketHandler sınıfı
public class PacketHandler<String> implements PcapPacketHandler<String>
{
// jNetPcap kütüphanesi tarafından otomatik olarak çağırılan nextPacket metodu
    public void nextPacket(PcapPacket pPcapPacket, String t)
    {
// Ethernet paketleri parsePcap metodu içinde ilgili alanlarına ayrıştırılara Profinet
paketleri
// oluşturulur.
            ProfinetPacket tProfinetPacket =
            PcapParser.parsePcap(pPcapPacket, mPacketNumer++);
    }
}
```

5.3. Uygulama Gerçeklemesi

Bir paket koklayıcı oluştururken temel olarak aşağıdaki adımların takip edilmesi gereklidir [1].

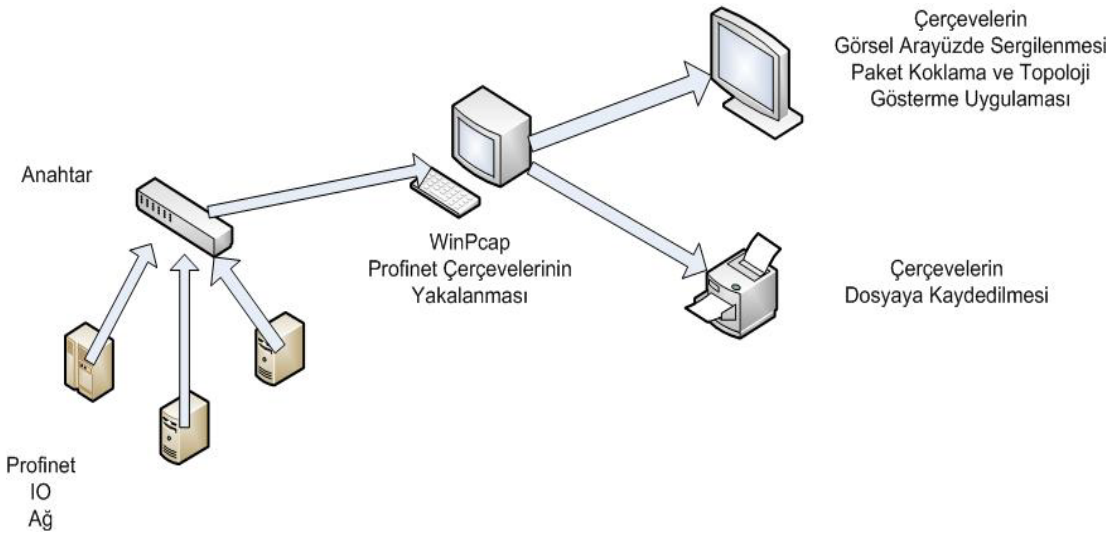
– Soket akışı oluşturma: UNIX ve klonlarında iletişim noktaları soket olarak adlandırılır. Bir soket oluşturulduğunda, ağ üzerindeki verilerin okunduğu bir soket akışı oluşturulur.

– Ağ arayüz kartını "promiscuous" moda alma,

– Açılan soket akışından gelen verileri okuma,

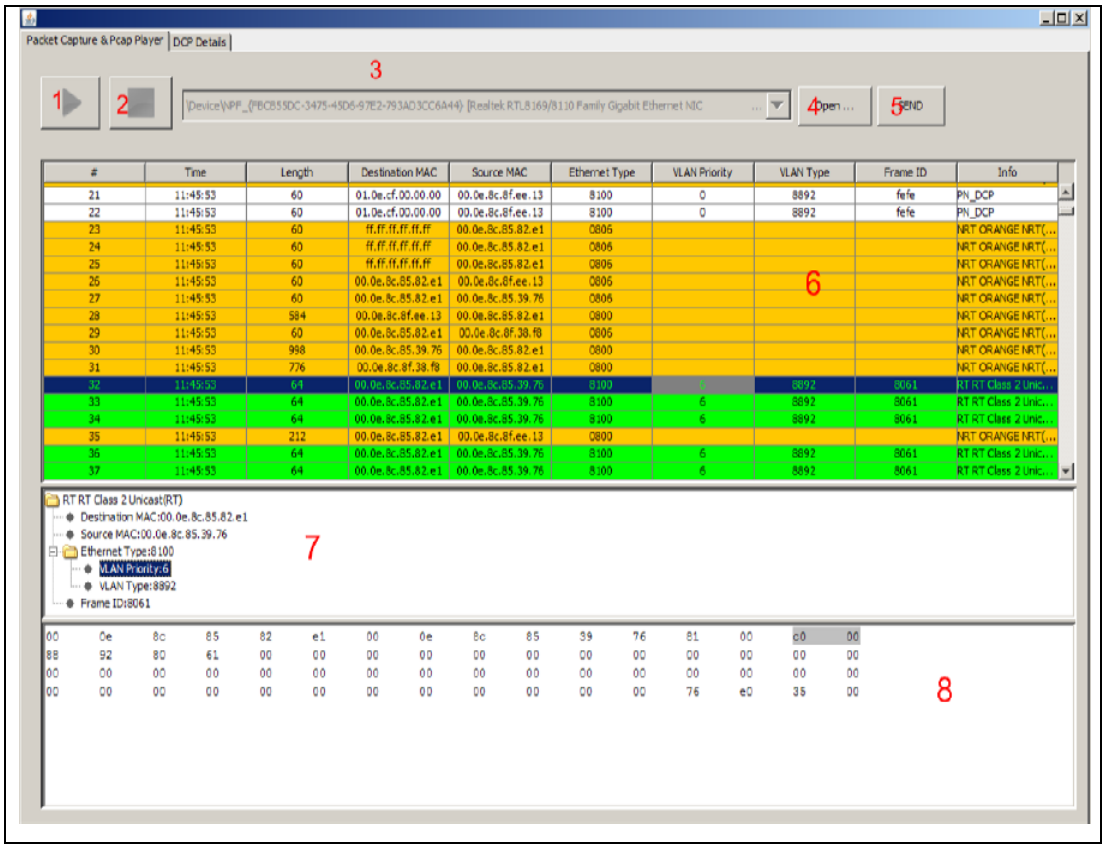
– Verileri başlık ve veri olarak ayrıştırıp bunları inceleme. Bu kısım paket koklayıcı oluşturmadaki asıl zorlu bölümü oluşturur. Kullanıcı bu adımı yapabilmesi için, incelemek istediği protokolün paket yapısı hakkında temel bilgiye sahip olması gereklidir. Protokol içeriği, alanları ve alan uzunlukları ile bu alanların içinde kullanıcının olmasını beklediği verilerin içeriğinin bilinmesi gereklidir. Örneğin Linux işletim sistemi çalışan bir makinede <linux/ip.h> ve <linux/tcp.h> başlıklarının eklenmesi ile IP ve TCP protokollerine ait yapılar otomatik olarak tanınabilir. Ancak kullanışlı bir paket koklayıcının sadece belli protokole ait paketleri değil bütün paketleri tanıyabilmesi önemlidir.

Tez çalışması sırasında gerçeklenen Profinet IO Ağ Çözümleyici uygulaması için Şekil 5.2'teki gibi bir ağ kullanılmıştır. Profinet IO ağına bir anahtar aracılığı ile bağlanan uygulama ağdaki çerçeveleri alıp yukarıda belirtilen çözümleme algoritmasını kullanarak çerçevenin hangi Profinet IO protokolüne ait olduğunu bulur ve görsel olarak listeler. Ayrıca yakalanan çerçeveler String formatında bir metin dosyasında kaydedilir (Şekil 5.3).



Şekil 5.2. Ağ Çözümleyici ağ yapısı ve çerçevelerin hareket yönü

Program tarafından yakalanan çerçeveler bir tablo içerisinde listelenir. Tabloda bir çerçeveye ait satıra tıklandığında alt bölümde ilgili çerçevenin byte dizisi olarak ayrıntısı ve seçilen alan üstü işaretli olarak görüntülenir (Şekil 5.3).



Şekil 5.3. Paket Koklayıcı Arayüzü

1. Seçili olan ağ arayüzünden canlı olarak paket yakalama işlemini başlatma tuşu.

2. Paket yakalama işlemini sona erdirme tuşu.

3. Sistemde paket yakalama veya gönderme amacıyla kullanılabilecek ağ arayüzlerini listeme ve seçme menüsü.

4. Pcap Oynatıcısında kullanılan ve sistemde kayıtlı çerçeveleri içeren dosyayı seçmek için kullanılan tuş.

5. Pcap Oynatıcısı uygulamasında kullanılan ve daha önceden dosyaya kaydedilmiş ve tekrar göndermek için seçilmiş çerçeveleri, sistemde kullanıcı tarafından belirlenmiş ağ arayüzüne tekrar aynı sıra gönderilmesini sağlayan tuş.

6. Ağ üzerinden yakalanmış veya dosyadan okunmuş çerçevelerin Ağ Çözümleyicisi algoritmasından geçtikten sonraki çözümlenmiş halde gösterimi. Her satır bir çerçeveyi göstermektedir. Ağ Çözümleme algoritması ile;

- Çerçevenin yakalanma sırası,
- Yakalanma zamanı,
- Bayt olarak uzunluk değeri,

- Hedef MAC adresi,
- Kaynak MAC adresi,
- Ethernet Tipi,
- VLAN öncelik değeri,
- VLAN Ethernet Tipi,
- Çerçeve ID'si ve
- Profinet Protokol bilgisi

çözümlenebilir.

7. Kullanıcı tarafından tablo üzerine tıklama sonucunda seçili olan çerçevenin ağaç yapısında gösteriimi.

8. Kullanıcı tarafından tablo üzerine tıklama sonucunda seçili olan çerçevenin tüm içeriğinin Bayt olarak gösterimi.

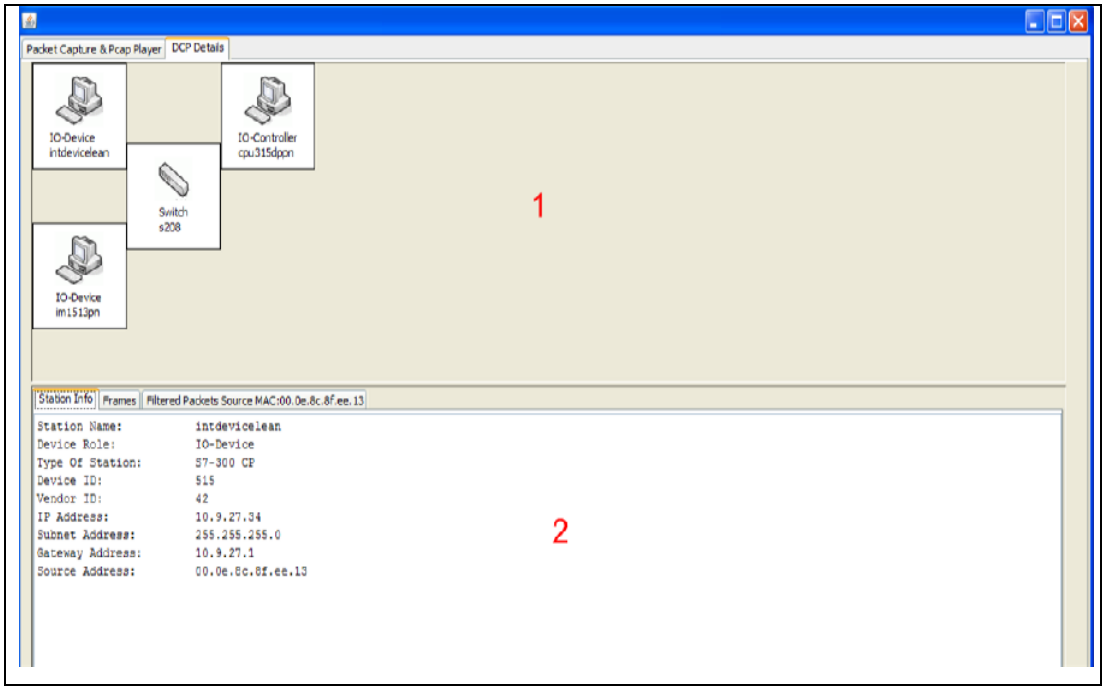
Paket yakalama uygulaması özellikle Profinet çerçevelerinin dolaştığı ağlar üzerinde, ağ yöneticisi tarafından sistemin hatalarını, performansını ve genel durumunu incelemek amaçlı olarak kullanılabilir. Sistemde hangi protokollere ait çerçevelerin daha sıklıkla gönderildiğini inceleyerek sistemin o anki durumu hakkında yorum yapmasına yardım edici bir gözlem aracı olarak kullanılabilir.

Paket yakalama uygulamasının devamı olarak geliştirilen diğer bir uygulama, Profinet ağı üzerindeki DCP protokolüne ait çerçevelerin yakalanarak sistemdeki cihazlara ait çeşitli parametrelerin gösterilmesini sağlayan topoloji gösterme uygulamasıdır. Bu uygulama Paket Yakalama veya Pcap Oynatıcı uygulaması ile yakalanan ve listelenen çerçevelerin arasındaki DCP protokolüne ait çerçeveleri inceler. Bu uygulama aslında Paket Yakalama ve Pcap Oynatıcı uygulamalarına bağımlı olarak çalışır. Yakalanan çerçeveler arasında DCP protokolüne ait olan ve FrameID değeri 0xFEFF olan çerçeveler cihazlara ait;

- İstasyon Adı (Station Name),
- Cihaz Rolü (Device Role),
- İstasyon Tipi (Type Of Station),
- Cihaz ID (Device ID),
- Üretici ID (Vendor ID),

– IP Adresi,
– Alt ağ adresi (Subnet),
– Geçit Adresi (Gateway),

parametlerini diğer cihazlara aktarırlar. Topoloji gösterme uygulaması sistemdeki cihazları listeler ve arayüz üzerinden seçilen cihaza ait parametleri (Şekil 5.4) ve parametrelerin elde edildiği DCP paketinin içeriğini gösterir (Şekil 5.5).



Şekil 5.4. Cihaz Listesi ve Parametleri

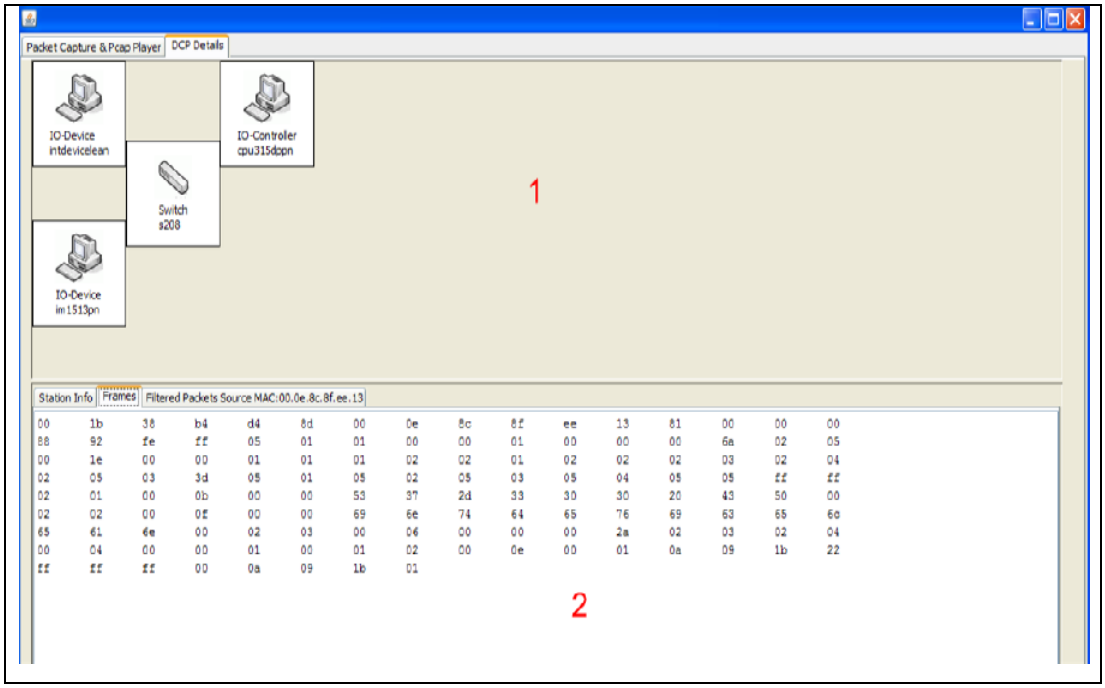
1. Ağ üzerindeki cihazların listesi. Her simge bir cihaza karşılık gelmektedir. Şekil 5.4'teki ağda 4 adet cihaz vardır. Bu cihazlardan anahtar görevindeki ortada gösterilmiş, diğer cihazlar bu anahtara bağlanarak yıldız topolojisi oluşturmuşlardır.

2. Cihaza ait simegeye tıklanarak seçilen cihaza ait bilgiler. Bu alanda;

– İstasyon Adı (Station Name),
– Cihaz Rolü (Device Role),
– İstasyon Tipi (Type Of Station),
– Cihaz ID (Device ID),
– Üretici ID (Vendor ID),
– IP Adresi,
– Alt ağ adresi (Subnet),

– Geçit Adresi (Gateway),
– Kaynak MAC Adresi

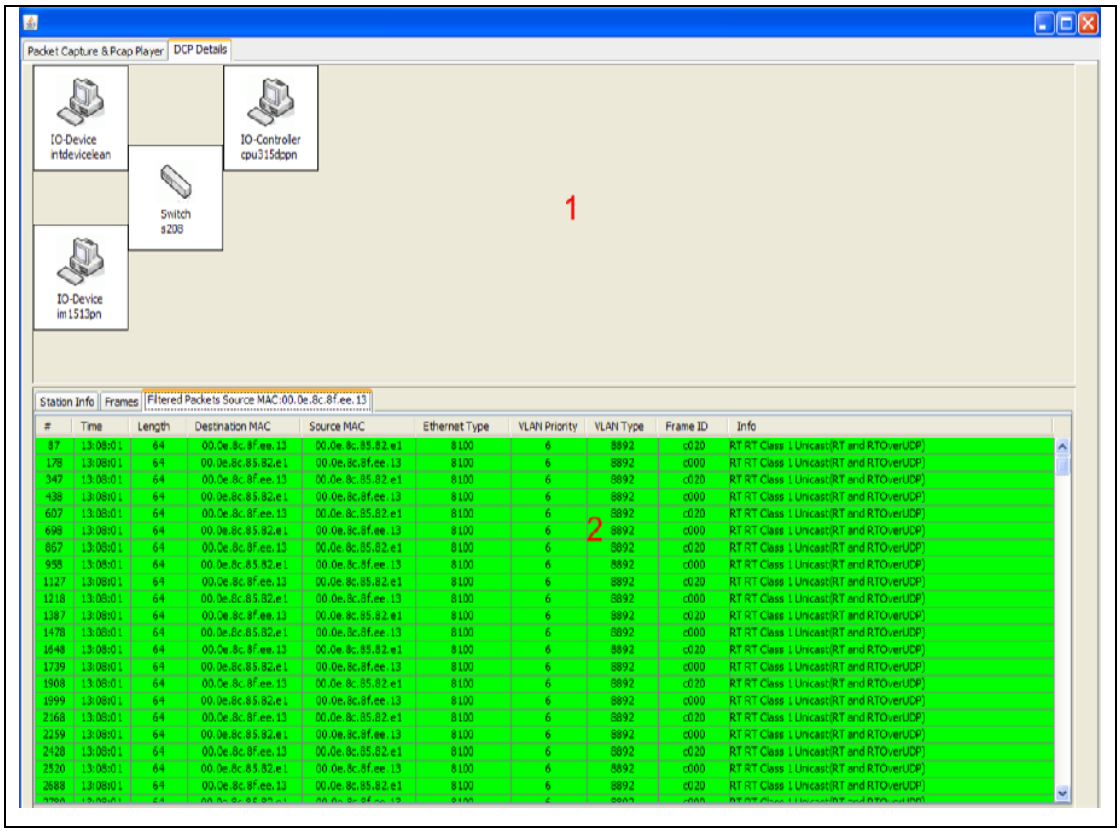
bilgileri sergilenir.



Şekil 5.5. Cihaza ait DCP Çerçevesi

1. Ağ üzerindeki cihazların listesi. Her simge bir cihaza karşılık gelmektedir.

2. Cihazı ait bilgileri çözümlemek amacıyla kullanılan DCP protokolüne ait verinin Bayt cinsinden gösterimi.



Şekil 5.6 Paket Filtreleme Ekranı

1. Ağ üzerindeki cihazların listesi. Her simge bir cihaza karşılık gelmektedir.

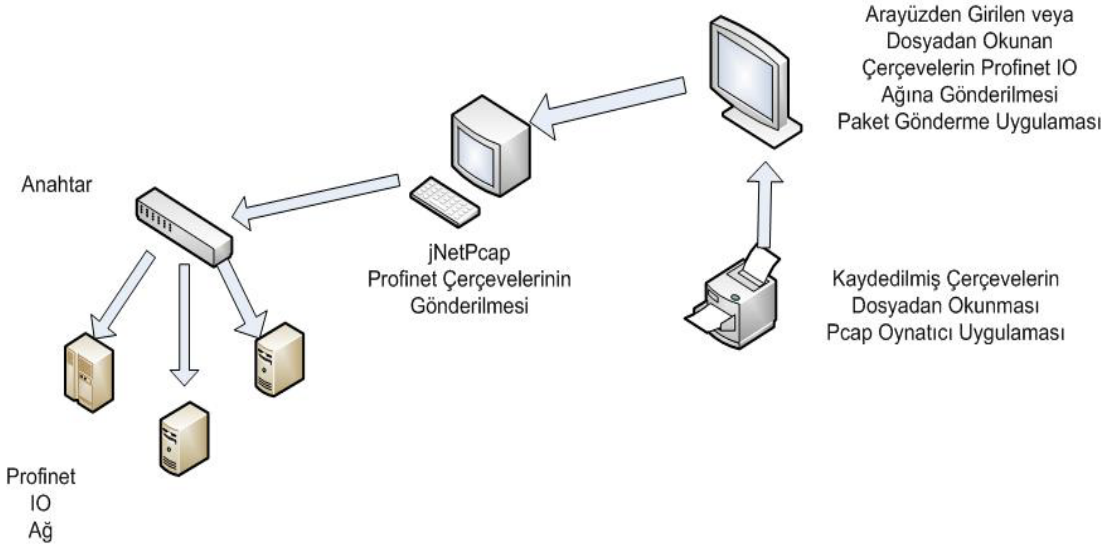
2. Seçilen cihaza ait trafik akışını gösteren çerçevelerin gösterimi. Bu alanda ana yakalama tablosundaki çerçevelerden sadece seçilen cihaza gelen veya seçilen cihazdan gönderilen çerçeveler görüntülenir. Bu alandaki çerçevelerin kaynak veya hedef adreslerinden birisi seçilen cihazın kaynak adresine denk gelmektedir.

Topoloji gösterme uygulaması ile çok sayıda ve fiziksel olarak birbirinden uzakta veya gözle görülemeyen yerlerde bulunan cihazların çalıştığı Profinet ağında cihazların birbiriyle haberleşme sıklığı, cihazların genel çalışma parametre hakkında bilgi sahibi olunabilir ve sistemin genel ağ topoloji görüntülenerek sistemin çalışma şekli hakkında fikir sahibi olunabilir.

BÖLÜM 6. PROFINET IO AĞ OYNATICI GERÇEKLEMESİ

Önceki bölümde, Profinet IO ağı üzerindeki çerçeveleri yakalayan ve bu çerçeveleri çözümleme algoritmasına göre inceleyen ve sınflayan Ağ Çözümleyici ve Topoloji Gösterme uygulamalarından bahsedilmiştir. Bu uygulamalar bir ağı pasif olarak sadece dinlemeye yönelik uygulamalardır. Bu bölümde ise kendi oluşturduğumuz veya daha önceden yakalanmış çerçeveleri ağa göndermeyi sağlayan Profinet IO Ağ Oynatıcı (Player) uygulamalarından bahsedilecektir. Bu tez çalışmasında Ağ Oynatıcı uygulamaları olarak Paket Gönderme ve Pcap Oynatıcı (Player) uygulamarı üzerinde çalışılmıştır. Bu uygulamalar ağa direkt olarak müdahale eden ve ağ trafiğini etkileyen uygulamalardır.

Paket Gönderme ve Pcap Oynatıcı uygulamarı için kullanılan geliştirme ortamı ve protokol çözümleme tablosu (Bkz. Tablo 5.1) önceki bölümde anlatıldığı gibidir. Sadece çerçevelerin akış yönü açısından farklılık vardır. Bu 2 uygulama için Şekil 6.1'teki gibi bir ağ yapısı kullanılmıştır.



Şekil 6.1. Paket Gönderme ağ yapısı ve çerçevelerin hareket yönü

6.1. Paket Gönderme Uygulaması

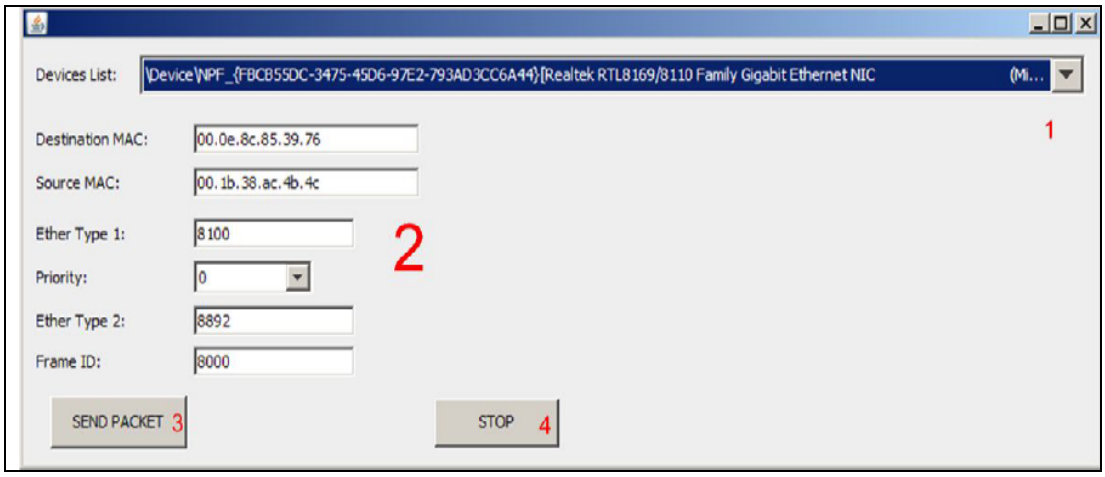
Paket Gönderme uygulaması kullacının seçilen bir ağ arayüzü üzerinden kendi oluşturduğu Ethernet çerçevesini göndermesine imkan sağlar. Kullanıcı bu uygulama ile bir Ethernet çerçevesinin;

- Hedef MAC adresini,
- Kaynak MAC adresini,
- Ethernet tip alanını,
- VLAN öncelik değerini,
- VLAN ethenet tip alanını,
- Frame ID ,

Alanlarının değerlerini kendisi belirleyerek bu çerçeveyi sisteme gönderebilir. Paket Gönderme uygulaması, Ağ Çözümleyici uygulamasında olduğu gibi jNetPcap kütüphanesi ile hazırlanmıştır. Bir çerçevenin alanlarının girilerek ağa gönderilmesi için aşağıdaki kod kullanılabilir:

```
// Ethernet paketi byte dizisi olarak ifade edilir.
byte[] dataFieldByteArray = new byte[64];
int snaplen = 64 * 1024;
int flags = Pcap.MODE_PROMISCUOUS;
int timeout = 10 * 1000;
StringBuilder errbuf = new StringBuilder();
Pcap pcap = Pcap.openLive(device.getName(), snaplen, flags, timeout, errbuf);
pcap.sendPacket(dataFieldByteArray);
```

Uygulama arayüzünden girilen parametreler byte dizisine çevrilir ve bu byte dizisi sendPacket metodu ile seçili olan ağ arayüzüne gönderilir (Şekil 6.2).



Şekil 6.2. Paket Gönderme Arayüzü

1. Çerçevenin gönderileceği ağ arayüzünü gösteren menü.

2. Çerçeveyi oluşturmak için kullanıcı tarafından girilen parametreler.

– Hedef Adresi,

– Kaynak Adresi

– Ethernet Tipi,

– VLAN Öncelik değeri,

– VLAN Ethernet Tipi,

– Çerçeve ID

3. Oluşturulan çerçeveyi gönderme tuşu.

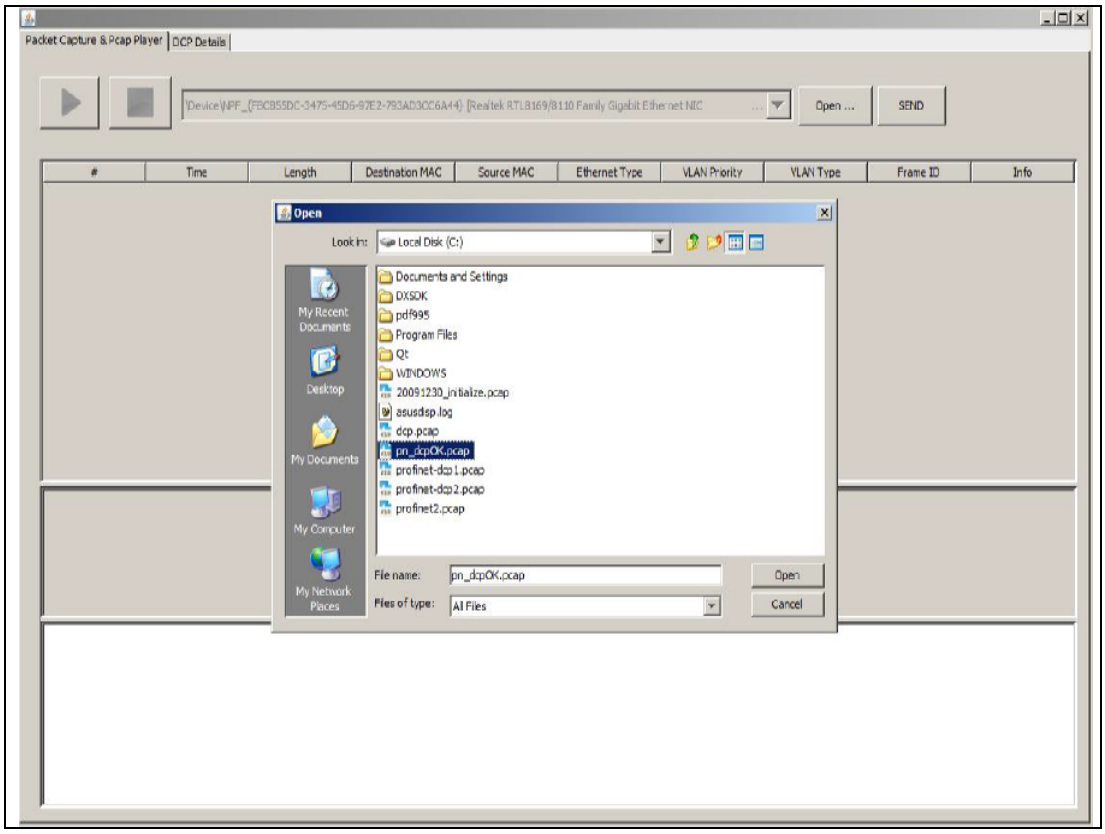
4. Çerçevenin döngüsel olarak sürekli gönderilmesi durumunda bu gönderimi durdurma tuşu.

Paket gönderme uygulaması ile Profinet ağına test amaçlı olarak çerçeveler gönderilerek, cihazların belirli komutlara ve tanımlı cihazlardan gelen belirli girdilere karşı tepkileri ölçülebilir, cihazların çeşitli özellikleri test edilebilir.

6.2. Pcap Oynatıcı (Player) Uygulaması

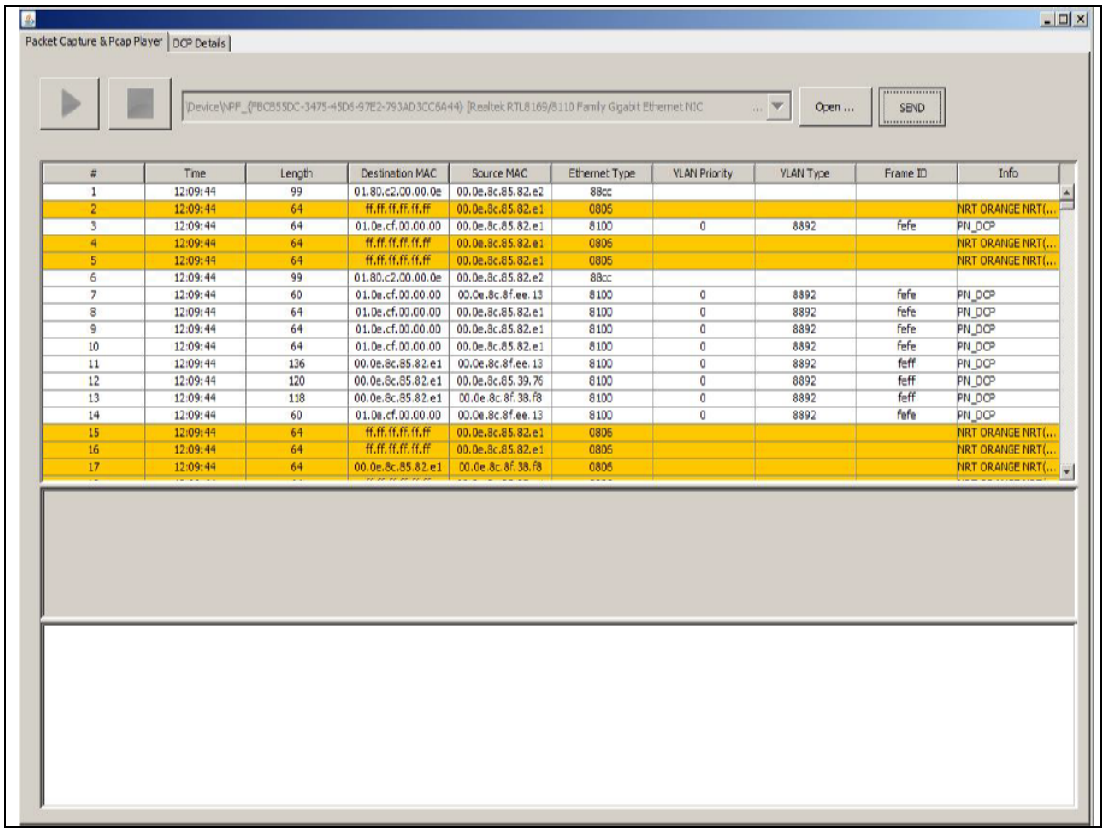
Tezin paket göndermeye dayalı bir diğer uygulaması Pcap Oynatıcı uygulamasıdır. Wireshark ile daha önceden yakalanan ve .pcap dosyası şeklinde kaydedilen çerçeveler, Pcap Oynatıcı uygulaması ile dosyadan okunur ve Ağ Çözümleme uygulamasında olduğu gibi Profinet çerçeveleri olarak sınıflandırılır ve Paket Gönderme uygulamasındaki gibi byte dizilerine dönüştürülerek tekrar ağa gönderilir. Pcap Oynatıcı uygulaması aslında Ağ Çözümleyici ve Paket Gönderme uygulamalarının birlikte çalışan halidir. Profinet IO çerçevelerinin çözümlenmesinde Ağ Çözümleme uygulamasında olduğu gibi Ağ Çözümleme algoritması kullanılır.

Pcap Oynatıcı uygulaması başlatıldığında öncelikle .pcap dosyası seçilir (Şekil 6.3).



Şekil 6.3. Pcap Oynatıcı Dosya Seçim Ekranı

Seçilen dosya içindeki çerçeveler ekranda listelenir ve "Send" tuşuna basılarak çerçeveler aynı sıra ile tekrar seçili olan ağ arayüzü üzerinden ağa gönderilir (Şekil 6.4).



Şekil 6.4. Send Tuşu ile Çerçevelerin Gönderilmesi

Pcap Oynatıcı uygulaması daha önce çalışan ve tekrarlanamayan hataların oluştuğu bir Profinet ağındaki hatalı çalışmaya neden olan çerçeveler, bir simülasyon ağ üzerinde tekrar ağa gönderilerek sistemdeki hataları görmek ve debug etmek amaçlı kullanılabilir.

BÖLÜM 7. SONUÇLAR

Bu tez çalışması ile öncelikli olarak Endüstriyel otomasyon haberleşme sistemlerinde kullanılan Ethernet tabanlı Profinet IO protokolleri ayrıntılı olarak incelenmiştir. Bu protokollerin çalıştığı bir ağ üzerine kolayca kurulabilen ve ağdaki trafiği yakalayarak Profinet IO protokollerine ait çerçeveleri çözümleyebilen Profinet IO Ağ Çözümleyici uygulaması, özel olarak geliştirilen Profinet IO Ağ Çözümleme algoritması ile ağdaki çerçeveleri sınıflandırabilmektedir. Bu sınıflandırma ile bir çerçeve içerisindeki alanların ne anlama geldiği ve hangi Profinet IO protokolüne ait olduğu görsel bir arayüz ile sergilenmektedir. Wireshark programı belirli bir derinliğe kadar çözümleme yapabilmekte ancak bu tez çalışmasında çözümlenen bazı alanlar için "Undecoded Data" uyarı vermektedir. Ağ Çözümlemesi ile Profinet IO protokollerinden DCP protokolüne ait çerçevelerin çözümlenmesi sonucunda ağın topolojisi görsel olarak kullanıcıya sunulabilmektedir.

Bu tez çalışması ile Profinet ağını pasif olarak dinlemek dışında ağa çerçeve gönderebilmek suretiyle ağa müdahale edebilmek de mümkündür. Profinet IO Ağ Oynatıcı uygulamaları sonucunda, kullanıcı arayüzünden girilen parametreler ile oluşturulan veya daha önceden yakalanmış ve bir dosyaya kaydedilmiş çerçevelerin tekrar aynı sıra ile ağa gönderilmesi mümkün hale gelmiştir.

Bu tez çalışmasında geliştirilen uygulamalar ve çözümleme algoritması Profinet IO protokollerine özel olarak geliştirilmiştir. Bu uygulamalar piyasada oldukça fazla sayıda bulunan ve genel Ethernet çerçeveleri için çalışan paket koklama ve paket gönderme uygulamalarının Profinet için özelliştirilmiş halleridir.

KAYNAKLAR

[1] ANSARİ, S., RAJEEV, S.G., CHANDRASHEKAR, H.S., Packet Sniffing, IEEE, Sayfa 17-20 Aralık 2002-Ocak 2003

[2] www.syngress.com, Analysis, Troubleshooting, and Packet Sniffing Chapter 7, Mayıs 2010

[3] Alder 2007 How to Cheat at Configuring Open Source Security Tools, Chapter 7: Introducing Wireshark: Network Protocol Analyzer

[4] http://www.wireshark.org/docs/wsdg_html_chunked/ChIntroPlatforms.html, Mayıs 2010

[5] PAXSON, V., ORTİZ, J., ANTHONY JOSEPH, D., Capturing & Analyzing Network Traffic: tcpdump/tshark and Wireshark, EE 122:Introl to Communication Networks

[6] PIGAN, R., METTER, M., Automating with PROFINET: Industrial communication based on Industrial Ethernet

[7] KAHVECİ, E., Endüstriyel Veri İletişim Sistemlerinde Bir Profinet Uygulaması, Y.Lisans Tezi, İstanbul Teknik Üniversitesi, Ocak 2007

[8] JASPERNEITE, J., SCHUMACHER, M., WEBER, K., Limits of Increasing the Performance of Industrial Ethernet Protocols, 2007

[9] SILVOLA, R., HANNİLA, J., SEPPALA, J., KOIVISTO, H., Experimentel Performance Evaluation of Profinet IO Real-Time version, 2008

[10] http://www.wireshark.org/docs/dfref/, Mayıs 2010

[11] http://wiki.wireshark.org/ProtocolReference, Mayıs 2010

ÖZGEÇMİŞ

1980 yılında İzmir'de doğdum. 1998 yılında İzmir Atatürk Lisesi'nden mezun olup aynı yıl İzmir Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği bölümünde lisans eğitimime başladım 2003 yılında bu bölümden mezun oldum. 2003 Temmuz ayından itibaren TÜBİTAK Marmara Araştırma Merkezi ve TÜBİTAK UEKAE'ye bağlı olan Bilişim Teknolojileri Enstitüsü'nde Yazılım Mühendisi Uzman Araştırmacı ünvanıyla görev yapmaktayım.