

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

KABLOSUZ AĞLARDAN ÇOKLU-ORTAM İLETİMİ

YÜKSEK LİSANS TEZİ

Elektronik ve Haberleşme Mühendisi Ercan YİĞİT

Anabilim Dalı: Elektronik ve Haberleşme Mühendisliği

Danışman: Prof. Dr. Sarp ERTÜRK

KOCAELİ, 2008

KOCAELİ ÜNİVERSİTESİ*FEN BİLİMLERİ ENSTİTÜSÜ

KABLOSUZ AĞLARDAN ÇOKLU-ORTAM İLETİMİ

YÜKSEK LİSANS TEZİ

Elektronik ve Haberleşme Mühendisi Ercan YİĞİT

Tezin Enstitüye Verildiği Tarih: 04.06.2008

Tezin Savunulduğu Tarih: 07.07.2008

Tez Danışmanı

**Prof. Dr.
Sarp ERTÜRK**


(.....)

Üye

**Yrd. Doç. Dr
Cabir VURAL**


(.....)

Üye

**Yrd. Doç. Dr
Oğuzhan URHAN**


(.....)

KOCAELİ, 2008

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında WiMAX türü kablosuz geniş-bant ağlarda çoklu-ortam aktarımını gerçekleştirilmiştir. Bu çalışmanın günümüz ihtiyaçlarına cevap verecek nitelikte olduğunu umut ederek, bundan sonra yapılacak olan çalışmalara kaynak olabildiğini dilerim.

Bu tezin hazırlanmasında özendirici ve titiz katkılarını esirgemeyen lisans, yüksek lisans ve tez çalışması süresince, derin görüşlerinden faydalanma imkânı sağlayan ve sürekli sabır gösterip destekleyen danışman hocam Prof. Dr. Sarp Ertürk' e, yazılım noktasında sorularıma cevaplar bulan mühendis arkadaşlarım Cüneyt Aksakallı' ya ve Melih Gürkan' a, test süreci boyunca yanımda bulunan mesai arkadaşım Levent Koç'a ve yüksek lisans çalışmalarımı sabırla destekleyen tüm V.O.S.S Varinak A.Ş. teknik servis ekibine teşekkür ederim. Ayrıca sevgiyle ve güvenle yanımda olan sevgili eşim Pelin Yiğit'e ve aileme minnettarım.

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ	iii
TABLolar DİZİNİ	iv
KISALTMALAR	v
ÖZET	vii
İNGİLİZCE ÖZET	viii
1. GİRİŞ	1
2. GENEL BİLGİLER	3
2.1. TCP / IP Referans Modeli	3
2.1.1. TCP/IP protokol katmanları	5
2.1.1.1. Ağ erişim katmanı (Network access layer)	5
2.1.1.2. Yönlendirme katmanı (Internet layer)	6
2.1.1.3. Taşıma katmanı (Transport layer)	8
2.1.1.4. Uygulama katmanı (Application layer)	11
2.2. Kablosuz İletişim ve WiMAX	12
2.2.1. Kablosuz ağ teknolojileri	12
2.2.2. WiMAX teknolojisi	13
2.2.2.1. WiMAX fiziksel katmanı ve teknolojiyi belirleyen değişkenler	15
3. ÇOKLU-ORTAM AKTARIMI UYGULAMASI	20
3.1. Uygulamaya Genel Bakış ve Bilgiler.....	20
3.1.1. Soket programlama ve .NET mimarisi	20
3.1.2. C# programlama dili ve Multi-Threading	21
3.2. Uygulama Aşamaları ve Analizi	22
3.2.1. Giriş	22
3.2.2. Program arayüzü ve değişkenleri	22
3.2.3. Uygulamada kullanılan mesaj formatları	24
3.3. Uygulama Akış Diyagramı ve Kod Parçacıkları	24
3.3.1. Akış diyagramı	24
3.3.2. Kod parçacıkları	26
4. SONUÇ VE ÖNERİLER	30
KAYNAKLAR	31
EKLER	33
ÖZGEÇMİŞ	54

ŞEKİLLER DİZİNİ

Şekil 2.1. TCP/IP modelinde veri iletimi	4
Şekil 2.2. Verinin Protokole göre değişen isimleri	5
Şekil 2.3. Paket Formatı	7
Şekil 2.4. UDP Paket Yapısı	9
Şekil 2.5. TCP Segment Yapısı	9
Şekil 2.6. SCTP Multi-homing Yapısı	10
Şekil 2.7. Çoklu Akış Yapısı	11
Şekil 2.8. Kablosuz Ağ Teknolojileri	12
Şekil 3.1. Program Arayüzü	23
Şekil 3.2. Uygulama Akış Diyagramı	25
Şekil 3.2. Uygulama Kullanımında	29

TABLÖLAR DİZİNİ

Tablo 2.1. OFDMA Parametreleri	16
Tablo 2.2. WiMAX Parametreleri	16
Tablo 2.3. Frekans ve Hız Değişkenleri Temelinde Kablosuz Ağ Teknolojileri	17
Tablo 2.4. WiMAX Sertifikasyon Profilleri	18
Tablo 2.5. Kullanılan WiMAX Frekans Aralığı ve Ülke Sayıları	18

KISALTMALAR

AES-CCM: Counter Mode with Cipher Block Chaining Message Authentication

AP: Access Point

ARP: Adress Resolution Protocol

CMAC: Cipher-based MAC

DHCP: Dynamic Host Configuration Protocol

DL: Downlink

DPCM: Differential Pulse Code Modulation

DSL: Digital Subscriber Line

EAP: Extensible Authentication Protocol

FDD: Frequency division duplex

FFT: Fast Fourier Transform

FPGA: Field Programmable Gate Array

FTP: File Transfer Protocol

GCP: Gateway Control Protocol

GSM: Global System For Mobile Communication

HTTP: Hyper Text Transfer Protocol

HMAC: Hash Message Authentication Code

ICMP: Internet Control Messaging Protocol

IFFT: Inverse Fast Fourier Transform

IGMP: Internet Group Management Protocol

IP: Internet Protocol

ISDN: Integrated Services Digital Network

ITU-T: ITU Telecommunication Standarts Department

JPEG: Joint Picture Experts Group

LAN: Local Area Network

LLC: Logical Link Control

MAC: Media Access Control

MIMO: Multiple-Input and Multiple-Output

MPEG: Motion Pictures Experts Group

MPLS: Multiprotocol Label Switching

MTU: Maximum Transmission Unit

NAT: Network Address Translation

NWG: Network Working Group

OFDM: Orthogonal Frequency-Division Multiplexing

OFDMA: Orthogonal Frequency-Division Multiplexing Access

OSI: Open System Interface

PC: Personal Computer

PCMCIA: Personal Computer Memory Card International Association

PSTN: Public Switched Telephony Network

QoS: Quality of Service

RARP: Reverse Adress Resolution Protocol

RF: Radio Frequency

S-OFDMA: Scalable Orthogonal Frequency-Division Multiplexing Access

SCTP: Stream Control Transmission Protocol

SMTP: Simple Mail Transfer Protocol
SNMP: Simple Network Management Protocol
TCP: Transmission Control Protocol
TDD: Time Division Duplex
UDP: User Datagram Protocol
UL: Uplink
VoIP: Voice Over Internet Protocol
WAN: Wide Area Network
WEP: Wired Equivalent Privacy
Wi-Bro: Wireless Broadband
Wi-Fi: Wireless Fidelity
WiMAX: Worldwide Interoperability for Microwave Access
WLAN: Wireless Local Area Network
WMAN: Wireless Metropolitan Area Network
WPA: Wi-Fi Protected Access
WPAN: Wireless Personal Area Network

KABLOSUZ AĞLARDAN ÇOKLU-ORTAM İLETİMİ

Ercan YİĞİT

Anahtar Kelimeler: Wimax, Multimedya aktarımı, Soket programlama, C# uygulaması, TCP/IP, SCTP.

Özet: Günlük yaşamımızda sıkça kullandığımız ve hayatımızın önemli bir parçası haline gelen kablosuz ağlar sürekli olarak gelişme göstermektedir. Bu tez çalışmasında yaygın olan kablosuz ağ (Wi-Fi Wireless Fidelity) üzerinden çoklu ortam (multi-media) aktarımı gerçekleştirilmiştir. Normal Wi-Fi standartlarından farklı olarak program yapısı, yeni gelişmekte olan ve Wi-Fi teknolojisine göre kapsama alanı ve bant genişliği noktasında büyük avantaj sağlayan WiMAX (Worldwide Interoperability for Microwave Access) yapısı göz önüne alınarak kurulmuştur. Uygulama Microsoft .NET platformu altında çalışmak üzere, Visual C# programlama dili kullanılarak geliştirilmiştir. Uygulama aynı yerel ağ üzerinde kablosuz veya kablo yardımı ile bir erişim noktasına bağlanan bilgisayarlarda, aynı anda karşılıklı 2 bilgisayarın haberleşmesini sağlayacak şekilde tasarlanmıştır.

Haberleşme sağlayacak olan iki bilgisayar birbirlerine ait IP (Internet Protocol) adreslerini ve veri aktarımını sağlayacak port adreslerini belirledikten sonra veri transferini başlatırlar. Program sunucu yapısı olmadan sadece bilgisayar – bilgisayar (Peer-to-Peer) yapısı üzerine kurulmuştur. Şu anda internet ortamında yaygın olarak kullanılan anlık mesajlaşma programlarına alternatif olarak kullanılabilir olan uygulamanın asıl önemli özelliği ise ikişer farklı IP adresleri üzerinden kurulacak olan iletişimde, herhangi bir hattın kopması durumunda iletişimin sürekliliğinin sağlanabilmesidir. Bu özellik, ağ yapısında kullanılan protokolleri sınıflayan OSI modelinde, taşıma katmanında (transport layer) yeralan TCP (Transmission Control Protocol) yerine, yine benzer bir protokol olan SCTP (Stream Control Transmission Protocol) multihoming özelliği kullanılarak gerçekleştirilmiştir.

Uygulamada tamamen TCP/IP protokol bütünü kullanılmıştır ve altyapısını oluşturan tüm aşamalar detaylı biçimde tez çalışması boyunca incelenmiştir.

A SOFTWARE APPLICATION FOR MULTIMEDIA TRANSMISSION OVER WIRELESS NETWORKS

Ercan YİĞİT

Keywords: Wimax, Multimedia transmission, Socket programming, C# application, TCP/IP, SCTP.

Abstract: Wireless Networks which are being widely used nowadays are growing rapidly. The multimedia transmission over Wi-Fi (Wireless Fidelity) has been realized by the software which was developed around the scope of this thesis. It has been developed by taking into consideration the new technologies like WiMAX (Worldwide Interoperability for Microwave Access) which has better coverage and much more available bandwidth than Wi-Fi. The application has been developed using C# programming language over .NET platform. It has been designed to enable two computers to communicate on the same network connected to an access point via wireless or cable.

Two computers, which will communicate, start data transfer after specifying the IP addresses and port numbers of each other. The application has been built on Peer-to-Peer structure without any server configuration and has the basic properties of programs which provide improved voice meetings and which is used intensively by internet users. But the most important property of this application is maintaining the data transfer between two computers which has more than one network interface. This property has been developed by using multihoming feature of Stream Control Transmission Protocol (SCTP) instead of TCP.

In this application, complete TCP/IP has been used and all the steps of this application are examined in detail under various headlines around the scope of this thesis.

1. GİRİŞ

Kablosuz haberleşme sistemlerinin son yıllarda göstermiş olduğu gelişime paralel olarak devamlı artan altyapı ihtiyacının karşılanması amacıyla yeni teknolojiler geliştirilmektedir. Bu teknolojiler arasında son zamanlarda göstermiş olduğu gelişmeler ile WiMAX teknolojisi göze çarpmaktadır. Yüksek bant genişliği ve geniş kapsama alanı ile diğer kablosuz ağ teknolojilerine sağladığı üstünlükler sebebiyle gelişme ivmesi artarak devam etmektedir. Bant genişliğinin artışı yüksek hızlarda veri iletimine ihtiyaç duyan uygulamalar için çözümler sunmaktadır. Şu anda mevcut bant genişliği sınırları içerisinde uygulanması mümkün olan ve gelecekte tamamen PSTN (Public switched telephone network) şebekelerinin yerini alacağına kesin gözüyle bakılan VoIP (Voice over IP) uygulaması, çoklu ortam kavramının tüm öğelerini kapsayacak bir uygulamaya dönüşme yolunda hızla ilerlemektedir. İletişim yatırım maliyetlerinin çok büyük oranda düşmesi, bakım onarım giderlerinin düşmesi, hareketliliği artırması ve dünyanın her yerinden erişilebilirlik, kurulum maliyetlerinin düşmesi, operasyon maliyetlerinin düşmesi, yüksek güvenilirlik, gelişme ve yeniliklere açıklık, var olan internet altyapısı üzerine kurulabilmesi, ses ve veri hizmetlerinin bir arada verilebilmesi gibi birçok avantaja sahip olması öne sürülen bu kestirimi destekler niteliktedir.

VoIP teknolojisinin gelişimine paralel olarak PSTN ağları gibi GSM ağları da popülerliğini yitirme noktasındadır. Güney Kore'de uygulanan ve WiBro (Wireless Broadband) olarak adlandırılan teknoloji ile 50 Mbit/s ve 5 km çapa sahip olması ve 120 km/h el değiştirme (handover) kapasitesi ile veri transferi sağlanmaktadır[1]. Eş zamanlı olarak ses ve görüntü iletimimin TCP/IP tabanlı protokoller bütünü üzerine kurulması ve internet erişiminin mümkün kılınması gelecek süreçte, mobil cihazlar üzerindeki çiplerin (FPGA) gelişimine bağlı olarak cep telefonları benzeri cihazlarda yüksek veri hızlarında iletimin sağlanması yolunu açacaktır[2].

Tez çalışması boyunca WiMAX benzeri bir ağ yapısında çoklu ortam aktarımının nasıl sağlanabileceği incelenmiştir. Genel bilgiler bölümünde TCP/IP protokolleri ve WiMAX yapısı karşılaştırılmıştır, WiMAX ve kablosuz ağ üzerinden ortam

aktarımının avantajları ve dezavantajları tartışılmıştır. Üçüncü bölümde ise kablosuz ağ üzerinden çoklu ortam aktarımını gerçekleştiren, NET tabanlı uygulama tanıtılmış ve mevcut uygulamanın sonuçları üzerine açıklamalar yapılmıştır.

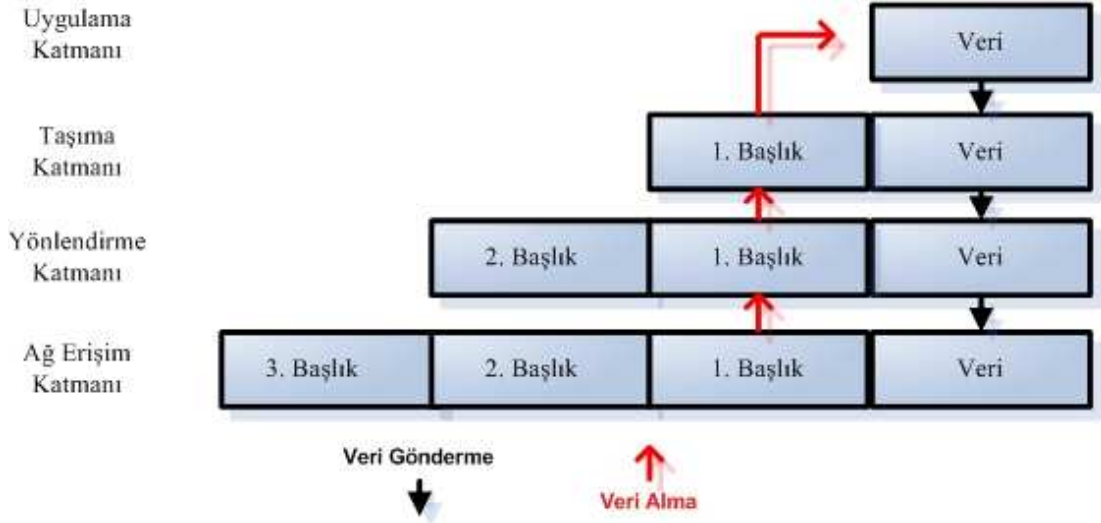
2. GENEL BİLGİLER

Tez çalışması için gerçekleştirilmiş olan uygulamanın daha iyi anlaşılabilmesi için gerekli bilgiler bu bölümde sağlanacaktır. TCP / IP protokolleri ve Wimax yapısı bu bölümde anlatılacaktır.

2.1 TCP / IP Referans Modeli

TCP/IP (Transmission Control Protocol/Internet Protocol) İletişim kontrol protokolü/Internet protokolü) aslında bir protokoller bütünüdür. İsmi bütünü içindeki iki popüler protokolden (TCP ve IP) almıştır. İnternet'in protokolü olan TCP/IP İnternet yayıldıkça daha popüler olmaya başladı. Fakat TCP/IP protokolünün önem kazanmasının asıl sebepleri şu şekilde özetlenebilir.

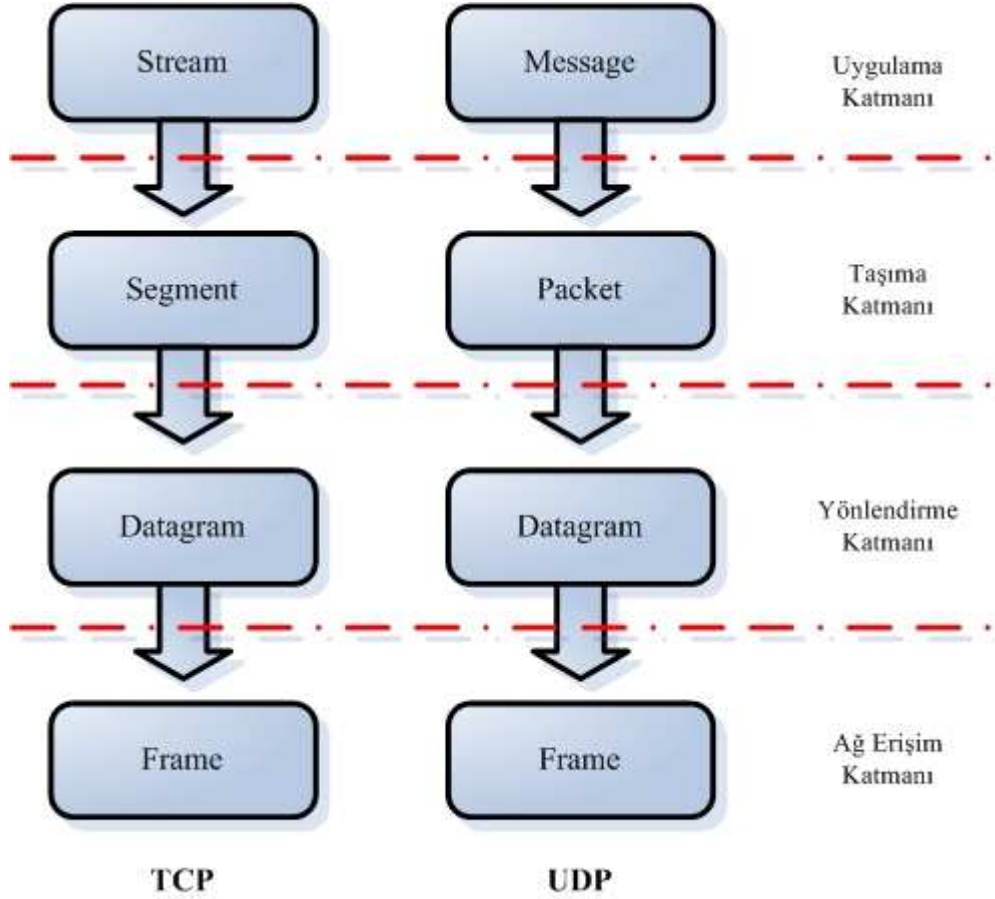
- a) Ücretsiz olarak, herhangi bir bilgisayar donanım ve işletim sisteminden bağımsız olarak geliştirilen açık bir standart olması. Bu özellik TCP/IP'nin çok fazla desteklenmesini ve İnternet bağlantısı olmasa bile farklı donanım ve yazılım bağlantısında kullanılmasını sağlamıştır.
- b) Özel bir network bağlantısı gerektirmemesi. Yani TCP/IP Ethernet, Token Ring ağlarda veya dial-up, X.25 bağlantılarda kullanılabileceği gibi hemen her tür ağ ortamında kullanılabilir[3].
- c) Adresleme sistemi sayesinde herhangi bir bilgisayara verilen adresin bütün dünya üzerinde tek olması sağlanmıştır.
- d) Son olarak kullanıcı hizmetleri için geliştirilmiş yüksek seviyeli protokolleri barındırması TCP/IP'nin popülerliğini artırmıştır.



Şekil 2.1: TCP/IP modelinde veri iletimi

TCP/IP’de OSI referans modeline benzemekte ve katman yapısında oluşturulmuştur. Bu katmanlar sırasıyla fiziksel katman (ağ erişim katmanı) - (Network Access Layer), yönlendirme katmanı (Internet Layer), taşıma katmanı (Transport layer) ve uygulama katmanıdır (Application Layer). Bu katman yapısında özetle iletişim şöyle gerçekleşmektedir[4][5]. Her katman kendi başlığını veriye ekleyerek bir alt katmana iletir. Alıcı tarafında alınan bilgi verici tarafında izlediği yolun tersini izleyerek - alt katmandan başlayıp en üst katmana doğru ilerleyerek – en üst katmana kadar ulaşır. Her katman geçişinde o katmana ait başlık verisi çıkarılarak uygulama katmanına sadece veri kalır. Şekil 2.1 ‘ de veri iletim yapısı görülmektedir.

TCP/IP protokolünde gönderilecek olan veriler taşıma katmanında kullanılan protokole göre isimlendirilmektedirler. Daha açık bir deyişle eğer taşıma katmanında TCP protokolü kullanılırsa veri segmentlerden oluşan “stream”, UDP protokolü kullanılırsa paketlerden oluşan “message” olarak adlandırılır. (Şekil 2.2)



Şekil 2.2: Verinin Protokole göre değişen isimleri

Şimdi alt-seviyeden başlayarak katmanlara ve her katmanda eklenen başlıklara bakalım. Örnekler verilirken daha çok UNIX sistemleri dikkate alınmıştır.

2.1.1. TCP/IP protokol katmanları

Bu bölümde TCP/IP Protokol Katmanları incelenmiştir.

2.1.1.1 Ağ erişim katmanı (Network access layer)

TCP/IP'nin en alt seviyesinde bulunan bu katman, veri iletilimde kullanılacak olan ağın yapısını belirler. Ağ yapısının detaylarını bilerek iletişim (IP adresi, MAC adresi) iletişim kurulmasını sağlar. Bunun sebebi aslında değişik ağ yapılarında (Ethernet, X.25 vs) farklı protokollerin kullanılmasıdır. Dolayısıyla teknoloji geliştikçe ve yeni ağ yapıları buldukça bu katmanda kullanılacak protokollerinde geliştirilmesi gerekir. Temel olarak bu katmanda yapılanlar IP datagramlarının ağ üzerinde iletilebilecek şekilde paketlere çevrilmesi ve IP adreslerinin o ağda kullanılan fiziksel adreslere (MAC) çevrilmesidir. Örnek olarak Ethernet yapısı

kullanıldığı zaman IP adresleri MAC adreslerine ARP (Address Resolution Protocol) protokolü yardımıyla, MAC adresleri de IP adreslerine RARP (Reverse Address Resolution Protocol) protokolü ile çevrilir[6].

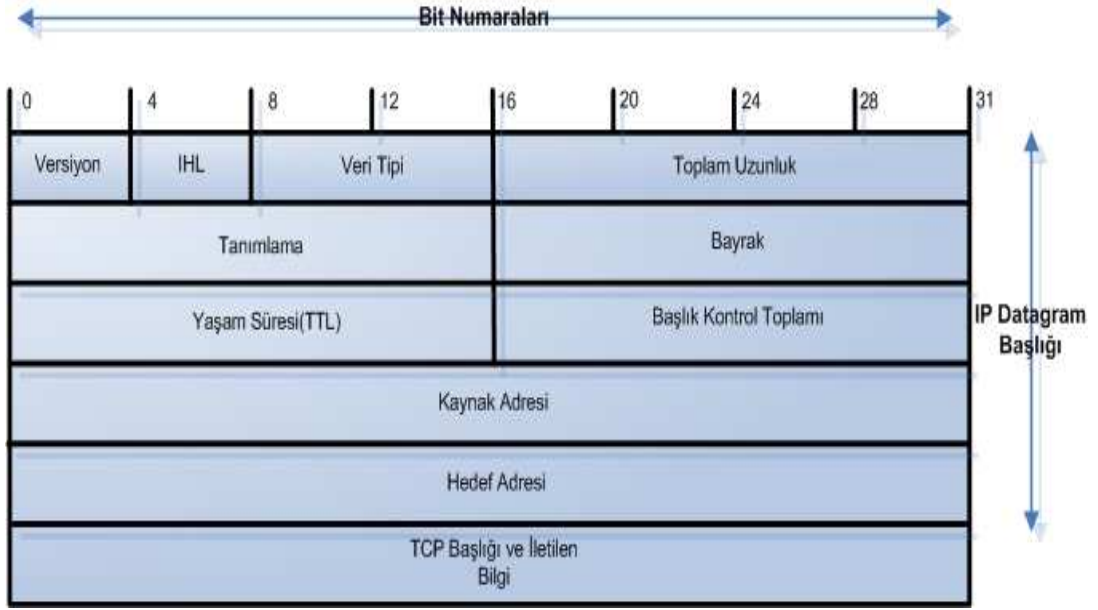
2.1.1.2. Yönlendirme katmanı (Internet layer)

Bu katmanda yer alan Internet Protokolü (IP) TCP/IP'nin en önemli protokolüdür [6]. Bu katmanların görevleri ise

- Verilerin ağ erişim katmanı ile taşıma katmanı arasında ilbağlantının sağlanması,
- Datagramların ağ üzerinde yönlendirilmesinin yapılması,
- Datagramların iletişimin sağlanması için frame' lere bölünmesi yada paketlerin oluşturulması olarak tanımlanabilir[6].

IP protokolü hata algılama (error detection) ve kaybolan datagramların tespiti işlemleri yapmaz. IP datagramları varış noktasına ulaşacak şekilde iletir, fakat karşı tarafın bunları alıp almadığını kontrol etmez. Kontrol işlemleri diğer katmanlar tarafından yapılır[4].

Paket anahtarlamalı bir ağda, paketler üzerlerindeki adres bilgisine göre iletirler. Her paket birbirinden bağımsız olarak iletir. Datagram Internet Protokol'ünce tanımlanan paket formatıdır. Şekil 2.3'de görülebileceği gibi ilk 5 yada 6 32-bitlik bölüm başlık olarak isimlendirilir. Bu başlıkta bir üst-seviyedeki katmanın başlığı ve daha sonra asıl bilgiler yer alır[7].



Şekil 2.3: Paket Formatı

Datagramlar varış noktasına ulaşmak için farklı fiziksel ağlardan geçtikçe o ağın iletebileceği paket büyüklüğüne göre parçalar ayrılabilir. Bu parçalar fragment olarak adlandırılır. Her ağın maksimum transmission unit (MTU) denilen yani maksimum iletebilecek birim tanımlamaları vardır. Datagramlar bunlara uyacak şekilde bölünürler. Her fragmentin başına, ait olduğu datagramın başlığı eklenir. Fakat birleştirme esnasında sırayı bilmek ve fragmentin ait olduğu datagramı bulmak için tanımlama ve fragment offset kısımlarına gerekli bilgiler yazılır.

Bu katmanda yer alan protokoller özet olarak aşağıda açıklanmıştır.

IP: Ağ üzerindeki herhangi bir cihaza ulaşacağı zaman sanal bir adres üretilir. Paket bağlaşmalı şebeke sistemlerinin birbirlerine bağlanabilmesi için tasarlanmıştır. IP paketlerine datagram denir. Bu protokolda paketlerin bölünmesi ve tekrardan birleştirilmesi de tanımlıdır[4].

ARP: Bir ağdaki makinelerin sabit MAC adreslerinin belirlenmesinde ve bu makinelerin birbirleriyle konuşurken MAC adreslerinin kullanımlarını sağlayan kurulları içeren bir protokoldür[8].

ICMP: Verilerin yerine ulaşp ulaşmamasıyla ilgili mesajların yollanmasından sorumlu olan protokoldür. İletim esnasında oluşacak sorunlarda ICMP protokolü

devreye girerek gelen verinin bozuk olduğunu belirtir. Böylece verinin tekrar gönderilmesi için gerekli olan mekanizma çalıştırılmış olur.

IGMP: Bu protokol bir veri paketinin birden çok bilgisayara gönderilmesi işleminden (multicasting) sorumludur.

2.1.1.3. Taşıma katmanı (Transport layer)

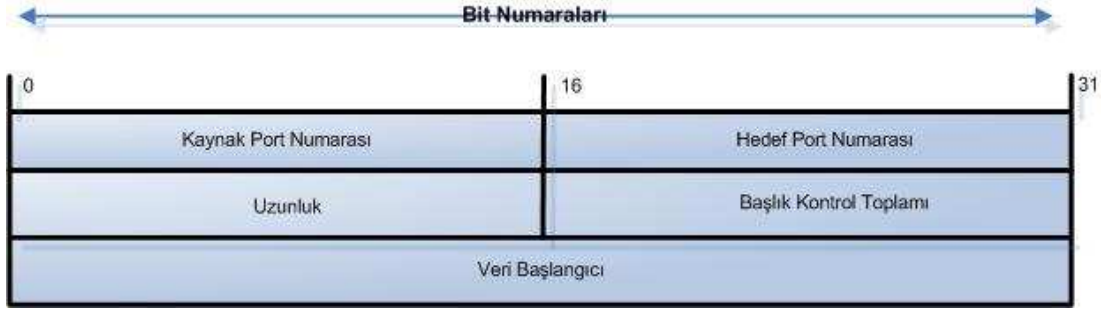
Bu katman veri transferinin yapılacağı iki bilgisayar arasında tüm iletişimin kurulması ve oturumun açılmasını sağlayan katmandır. İletişim Kontrol Protokolü (TCP: Transmission Control Protocol), Kullanıcı Datagram Protokolü (UDP: User Datagram Protocol) ve bu tezde iletişimin kurulmasında modellenen protokol olan Akış Kontrol İletişim Protokolü (SCTP: Stream Control Transmission Protocol) bu katmanda göze çarpan protokollerdir.

UDP, protokollerden kaynaklanan gecikmeleri en alt seviyeye indirerek iletişimin en üst hızda gerçekleşmesini sağlayan protokoldür.

UDP güvenilir olmayan, bağlantısız bir protokoldür. Güvenilir olmaması veri şifrenmesini ifade etmez, verinin karşı tarafa ulaşp ulaşmadığına dair bir kontrolün yapılmadığını anlatmaktadır. Protokolün daha hızlı olmasını sağlayan da bu özelliktir.

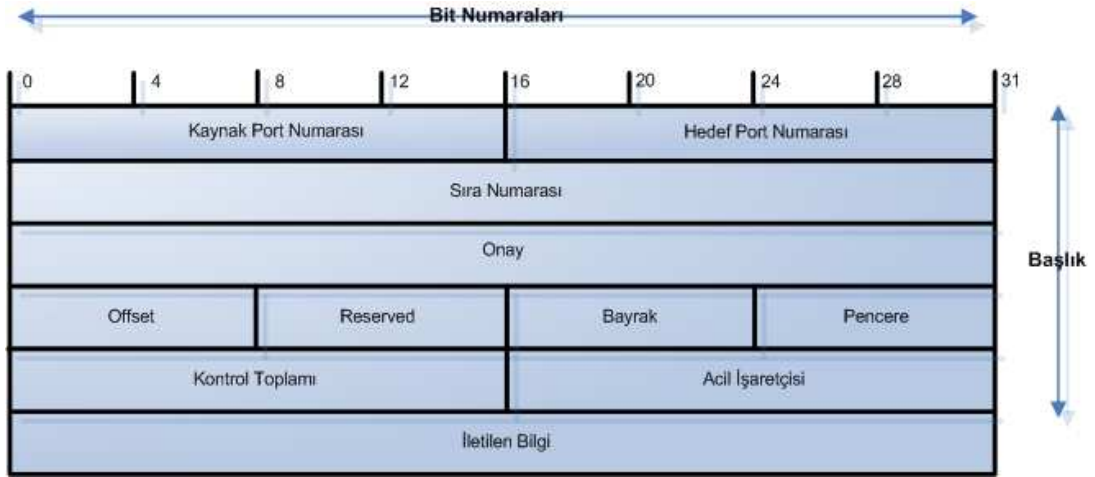
UDP'yi kullanmanın bazı sebepleri vardır. İletilecek veri miktarı azsa bu veriyi tekrar iletmek çoğu zaman bağlantı kurup güvenilir bir iletim yapmaktan daha kolay olmaktadır. Örneğin çoğu ses iletimi yazılımları UDP protokolünü kullanmaktadır. Bunun sebebi ise iletilen ses verisinin hedefe ulaşp ulaşmadığının ya da zamanlanmasının önemli olmayışdır. Kaybolan bir ses verisinin ait olmadığı farklı bir zamanda tekrar gönderilmesi iletişiminin verimsiz bir hale gelmesine yol açacaktır.

UDP mesaj içeriğine bakarak kullanılan port numaralarına göre verinin bir üst katmanda hangi protokole göre işleneceğine karar verir. Şekil 2.4'de bir UDP paketi görülmektedir.



Şekil 2.4: UDP Paket Yapısı

TCP ise güvenilir veri iletimine ihtiyaç duyan uygulamalar tarafından kullanılır. Yaptığı kontrollerden ve veri iletiminden önce bağlantı kurması gerektiğinden TCP segmentlerinin yapısı UDP paketlerine göre daha karışıktır. Şekil 2.5’de TCP Segment Formatı görülmüyor.



Şekil 2.5: TCP Segment Yapısı

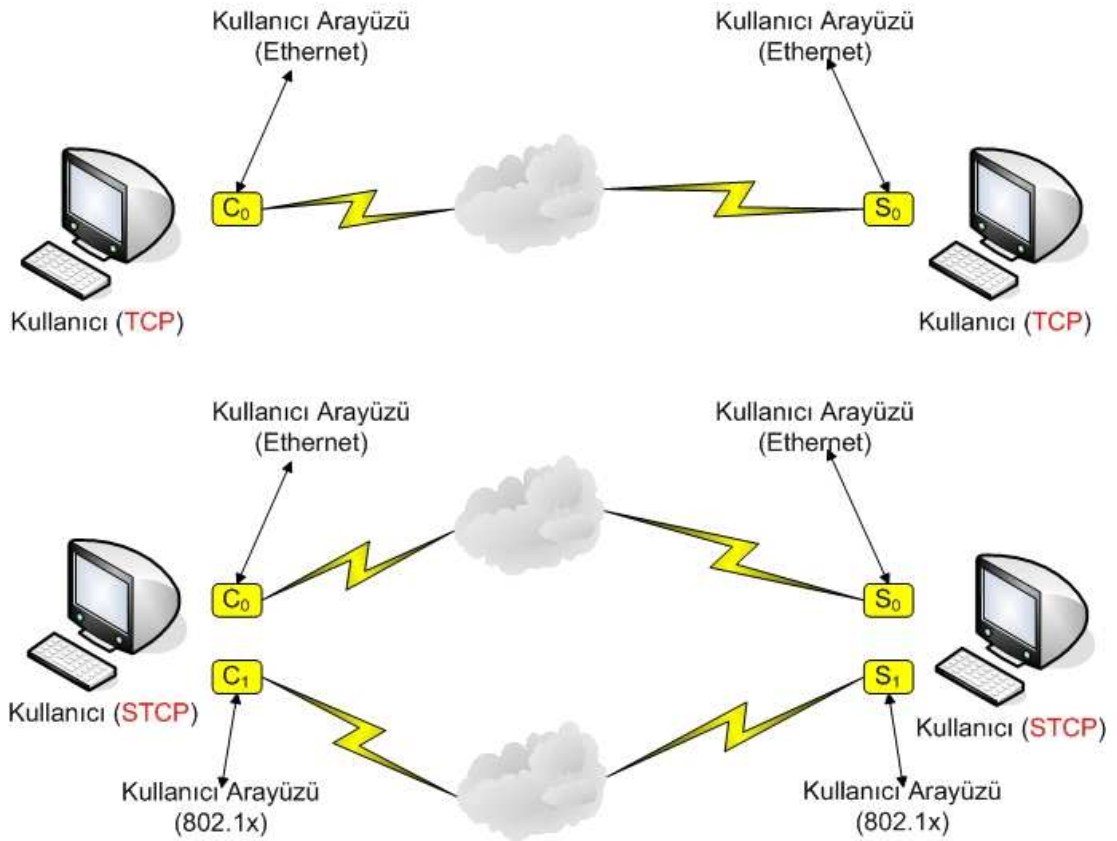
TCP’de veriler segmentler halinde iletilir. Kontrol toplamı bilgisi segmentlerin iletimi sırasında kayıp olup olmadığını belirlemek içindir. Eğer veri iletimi sırasında bir kayıp olmadıysa hedef makineye kaynak tarafında onay (Positive Acknowledgement) gönderilerek verinin iletilmiş onaylanır. Belli bir süre içinde onay gelmezse segment tekrar gönderilir.

TCP bağlantı-temelli (connection-oriented) bir protokol olduğu için ilk olarak gönderilen segmentler bağlantının oluşturulması amacıyla gönderilir. El sıkışma

(handshake) adı verilen yöntem ile bağlantı kurulduktan sonra veri transferi başlar. Segment başlığında kullanılan sıra numarası ile veriler sıraya konular. Pencereleme (Windowing) denilen yöntem ile kontrol sistemi her bir segment için değil, segment toplamı şeklinde yapılır. 4000 adı altında alınan bir onay numarası 4000'e kadar gönderilen segmentlerin alındığını gösterir.

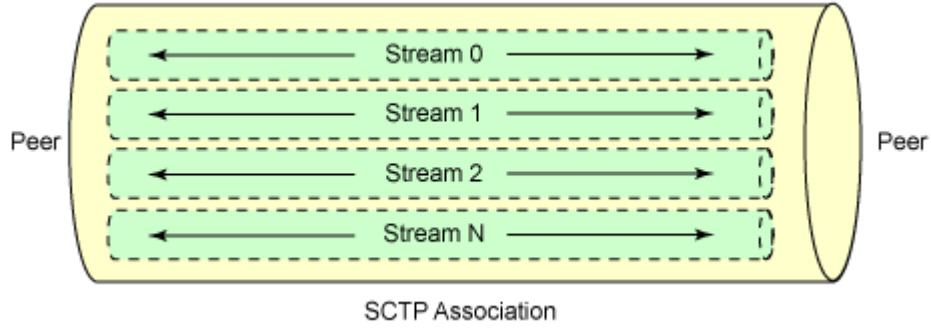
UDP protokolünde olduğu gibi TCP protokolünde de verinin hangi uygulamaya geçireceğini portlara bakarak karar verir.

SCTP protokolü ise yapısı ile TCP protokolüne benzemektedir. Güvenilir, sıralı veri gönderim ilkesi ile çalışır. SCTP protokolünü diğer taşıma katmanı protokollerinden ayıran en önemli özelliği ise multi-homing yapısıdır. Multi-homing kısaca aynı bilgisayarın sahip olduğu farklı iki ağ ara yüzü ile iletişimin eş zamanlı sağlanabiliyor olmasıdır. Şekil 2.6 bu yapıyı göstermektedir[9].



Şekil 2.6: SCTP Multi-homing yapısı

SCTP' nin göze çarpan bir diğer özelliği ise çoklu akış (multi-streaming) yapısıdır. TCP kullanılan bir yapıda kontrol ve veri bilgisi aynı bağlantıyı kullanırken, SCTP de çoklu akış özelliği sayesinde, kontrol ve veri ayrı veri demeti (stream) olarak gönderilerek kontrol bilgisinin veriden sonra gelmesinden kaynaklanabilecek problemlerin üstesinden gelinmiş olunur. Şekil 2.7'de çoklu akış yapısı gösterilmektedir[10],[11].



Şekil 2.7: Çoklu Akış Yapısı

2.1.1.4. Uygulama katmanı (Application layer)

Ağ üzerinden iş yapacak uygulamaların bulunduğu katmandır.

Bu katmanda HTTP (HyperText Transfer Protocol), SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol), SNMP (Simple Network Management Protokol) ve Telnet gibi protokoller çalışır.

HTTP: İnternet'te web sayfaların içeriğini görüntülemek için kullanılır.

FTP: İnternet'te hızlı dosya transferi yapılması için geliştirilmiştir.

SNMP: Belirli bir şebeke üzerindeki cihazların birbirleriyle haberleşmelerini ve ortak bir dil kullanmalarını sağlamak için geliştirilmiştir.

SMTP: İnternet üzerinden e-mail göndermeye yarayan bir protokoldür.

Telnet: İnternet ağı üzerindeki çok kullanıcı bir makineye uzaktaki başka bir makineden bağlanmak için geliştirilen bir protokoldür.

2.2. Kablosuz İletişim ve WiMAX

WiMAX yüksek bant genişliğinde iletişime izin veren ve çalışma yapısı ile GSM altyapısına benzeyen yeni kablosuz ağ teknolojisidir. Bu bölümde WiMAX teknolojisi detaylı olarak incelenecektir[12].

2.2.1 Kablosuz ağ teknolojileri

Kablosuz ağlarda veri iletimi için kullanılan bir kaç teknoloji bulunmaktadır. Bunların en önemlileri elektromanyetik dalgaları kullanılan RF ve çıplak gözle görülebilen ışığın altındaki frekansları kullanan kızılötesi teknolojisidir. RF ve kızılötesi teknolojileri WLAN sistemlerinde kullanılmakta olup, her birinin kendine özgü avantajları ve dezavantajları bulunmaktadır. Kullanıcıların kendi ihtiyaçlarına göre doğru teknolojiyi seçmeleri sistem verimliliğini ve memnuniyeti artırmaktadır. Günümüzde artan çoklu ortam uygulamaları sonucunda oluşan yüksek veri hızı talebi nedeniyle teknolojiler arasındaki rekabette veri hızı en önemli ölçüt olarak görülmektedir. Kapsama alanı veya erişim mesafesi ile girişime (interference) karşı duyarlılık da diğer önemli ölçütler olarak sıralanabilir. Uygulamada yüksek veri hızları ve fiziksel engelleri geçebilme özellikleri nedeniyle RF teknolojisi yaygın olarak kullanılmaktadır[13]. Şekil 2.8' de şu anda yaygın olarak kullanılan kablosuz ağ protokolleri gösterilmektedir.

KABLOSUZ TEKNOLOJİ	KAPSAMA ALANI	STANDART	HIZ
	EV YADA SINIRLI ALANLAR	BLUETOOTH	<1 Mbps
	KAMUSAL ALAN, KAMPÜS	IEEE 802.11	11-54 Mbps
	METROPOL, BÖLGE	IEEE 802.16	11-100 Mbps
	KITALARARASI, DÜNYA	GSM,GPRS	10-384 Kbps

Şekil 2.8: Kablosuz Ağ Teknolojileri

WLAN sistemleri havada yayılan elektromanyetik dalgalarla bir noktadan başka bir noktaya fiziksel bağlantı olmaksızın bilgi iletişimini sağlar. Tipik bir kablosuz yerel ağ yapısında, erişim noktası (AP = Access Point) olarak isimlendirilen hem alıcı hem

verici konumundaki cihaz kablolu ağa bağlanır ve kablolu ağ omurgası ile kablosuz cihazlar arasında veri alışverişi işlemi gerçekleştirir. Bir AP kullanılan ortama bağlı olarak dahili uygulamalarda 25-100 metre, harici kullanımda ise 200 metreye kadar yarıçaplı bir alanı kapsayabilir. WLAN sistemlerinde kullanılan yüksek frekanslı RF sinyali (2.4 GHz ve 5 GHz) temel özelliği nedeniyle katı cisimlere nüfuz edebilir ve geçebilir. Bu özellik görüş hattının sağlanamadığı bina içi kullanımlarda büyük bir avantaj yaratır. Ancak katı cisimler kullanılan maddeye (tahta, çelik, beton gibi) bağlı olarak sinyal zayıflamasına neden olurlar. Bu da sonuçta erişim mesafesini kısaltır. Lisansız kullanımlar için çıkış gücü düzenlemeler ile sınırlandırılmış (genellikle 100mW) olduğundan mesafe artırımı için güç yükseltilmesi söz konusu değildir. Bu nedenle iyi bir kapsama alanı için fiziksel ortam iyi etüt edilmeli ve AP montaj yerleri iyi seçilmelidir[14]. AP veya kullanılıyorsa AP'ye bağlı harici anten, genelde yüksek bir noktaya monte edilir. Bu sadece kapsama alanını genişletmek için gereklidir. Eğer yeterli kapsama alanı sağlanıyor ise AP'ler istenilen her noktaya konulabilir. Kullanıcılar ise kablosuz erişim özelliğine sahip cihazlar ile ağa bağlanabilirler. Bu özelliği bulunmayan bilgisayarlar için hariçten takılan kablosuz ağ adaptörleriyle, dizüstü bilgisayarda PCMCIA kartlarla, masaüstü bilgisayarlarda ise ISA/PCI kartlarla kablosuz erişim gerçekleştirilir.

Bu kısma kadar aktarılan tüm bilgiler tüm kablosuz ağ standartları için ortaktır. WiMAX teknolojisini diğer kablosuz ağ teknolojilerden ayıran özellikler diğer bölümde incelenecektir.

2.2.2 WiMAX teknolojisi

WiMAX IEEE 802.16 standardı ile tanımlanmış ve geliştirme aşaması “WiMAX Forum Network Working Group (NWG)” tarafından dünya üzerinden birçok katılımcı ile gerçekleştirilmektedir. Mobil WiMAX sağladığı geniş kapsama alanı ve yüksek bant genişliği özellikleri ile ağ yapısında önemli ölçüde esneklik kazandırmaktadır. Mobile WiMAX'ın göze çapan bazı özellikleri aşağıda belirtilmiştir.

Yüksek Veri Hızı : MIMO anten yapısında gözlenen gelişmeler ile ileri düzey modülasyon tekniği ve kodlama şemaları mobil WiMAX teknolojisinin UL (Uplink)

hızını sektör başına 28 Mbps, DL (Downlink) hızını sektör başına 63 Mbps seviyesine çıkarmaktadır[15].

Servis Kalitesi: Alt kanallara ayırma (Sub-Channelization) ve MAP temelli işaret şemaları uzay, frekans ve zaman kaynaklarını aynı haberleşme kanalı üzerinde en uygun seviyede işlenmesine olanak sağlamaktadır.

Ölçeklenirlik: Artan global ekonomi değerleri göz önüne alındığında kablosuz haberleşme için ayrılan altyapı hala ihtiyaçları karşılayamaz haldedir ve sınırlama altındadır. Bu sebeple herhangi bir sınır engeliyle karşılaşmamak amacıyla WiMAX teknolojisi 1.25 – 20 Mhz arasında farklı kanallar üzerinden iletim yapacak şekilde geliştirilmektedir.

Güvenlik: Wi-Fi teknolojisinde şu anda kullanılan ve güvenilirliği ciddi şekilde sorgulanan WEP-WPA şifreleme algoritmalarının göstermiş olduğu zayıflığın aksine Mobil WiMAX EAP temelli kimlik denetimi (authentication), AES-CCM temelli kodlama ve CMAC-HMAC temelli mesaj koruma şemaları ile yüksek düzeyde güvenlik önermektedir.

Hareket Kabiliyeti: Mobil WiMAX 50 milisaniye gibi çok düşük bir zaman aralığında el değiştirme işlemini gerçekleştirebilmektedir. Bu özellik VoIP gibi gerçek zamanlı uygulamalarda yüksek performans ile veri transferinin gerçekleştirilmesini sağlamaktadır. Daha açık bir ifade ile, VoIP iletişimini gerçekleştiren biri 120 km/h hızla bir WiMAX erişim noktasından diğerine geçerken iletişimde kopma yaşamayacaktır.

WiMAX, Wi-Fi ve Ethernet ile aynı LLC katmanını kullanmaktadır. Bu sebeple WiMAX LAN ve WAN gibi mevcut altyapılara entegre edilebilir. Bu uyumluluk göz önüne alındığında WiMAX son kullanıcı için tasarlanmış bir sistem değil, tamamen altyapı için hazırlanmış bir teknoloji gibi görünmektedir.

WiMAX için tasarlanan MAC katmanı, değişik fiziksel katmanlarla uyumlu halde geliştirildiği için üreticilere geniş bir yelpaze sunmaktadır. Böylelikle WiMAX kablosuz geniş-bant internetin yanında cep telefonları, avuç-ıçığı bilgisayarlar gibi araçlardan da sinyal alabilmektedir. Bu gibi nedenler GSM şirketlerinin de WiMAX 'e ilgi duymasına sebep olmaktadır. WiMAX teknolojisinin öngördüğü

gerçekleştirebilirse GSM firmaları alt yapıları için WIMAX 'ı kullanabilir, cep telefonlarında geniş bant gerektiren uygulamalar kullanıcıların hizmetine sunulabilir. Değişik fiziksel katmanlarla çalışabilen bir MAC katmanı WIMAX 'a birbirinden ilginç uygulama alanları getirebilir.

2.2.2.1 WiMAX fiziksel katmanı ve teknolojiyi belirleyen değişkenler

WIMAX için tasarlanan MAC katmanı Wi-Fi 'den farklı olarak çalışmaktadır. Wi-Fi 'de tüm kullanıcılar bir erişim noktasına aynı anda iletişim isteği gönderirler. Bu tip bir gönderide istekler rastgele bir biçimde sıralanır. Bu durumda erişim noktasına uzak olan istemci hep sonlarda sıraya girer. Belki kendisine sıra gelene kadar bağlantının ölü olduğuna kanaat getirip, bağlantının tekrar sonlandırılmasına sebep olur. WIMAX 'da ise bağlantı noktasına erişmek isteyen istemciler sıraya alınır. Böylelikle her istemci sadece bir kere yarışa girer. İstemcinin uzaklığına göre belli bir zaman aralığında erişim verilir. Bu tip bir sıralama algoritması ile bant genişliği daha da verimli kullanılır.

OFDM tekniği kullanılarak mevcut bant genişliği çoklu frekans alt taşıyıcılarına bölünür. Bu bölünme işlemi Ters Fourier Dönüşümü kullanılarak gerçekleştirilir. (Inverse Fast Fourier Transform IFFT) . S-OFDMA tekniği ile farklı bant genişliklerinde, alt taşıyıcılar arasındaki frekans aralığı sabit tutularak ölçeklenebilir bir iletim sağlanmış olur. Tablo 2.1'de OFDMA, Tablo 2.2'de WiMAX iletimi parametreleri verilmiştir[16].

Tablo 2.1: OFDMA Parametreleri

DEĞİŞKEN	DEĞER			
Kanal Bant Genişliği(MHz)	1,25	5	10	20
Örnekleme Frekansı (Fp)	1,40	5,6	11,2	22,4
FFT Genişliği	128	512	1024	2048
Alt Kanal Sayısı	2	8	16	32
Alt Taşıyıcı Frekans Boşluğu	10,94 KHz			
İşaret Zamanı (Tb=1/f)	91,4 microsaniye			
Güvenlik Aralığı (Tg = Tb/8)	11,4 microsaniye			
OFDMA İşaret Zamanı (Ts=Tb+Tg)	102,8 microsaniye			

Tablo 2.2: WiMAX Parametreleri

DEĞİŞKEN	DEĞER
Çalışma Frekansı	2500MHz
Dupleks	TDD
Kanal Bant Genişliği	10MHz
Verici-Verici Mesafesi	2,8 km
Minimum Mobil-Verici Mesafesi	32 m
Verici Yüksekliği	1,5 m
Verici Anten Kazancı	15dBi
Mobil Anten Kazancı	(-) 1dBi
Verici Maksimum Gücü	43dBm
Mobil Terminal Maksimum Gücü	23dBm
Alıcı(Rx) Verici(Tx) Anten Sayısı	Tx: 2, Rx:2
Verici Gürültü	4dB
Alıcı Gürültü	7dB

Tablo 2.3’de ise hız, mesafe ve frekansa bağlı olarak kablosuz ağ teknolojileri sıralanmıştır[17].

Tablo 2.3: Frekans ve Hız Değişkenleri Temelinde Kablosuz Ağ Teknolojileri

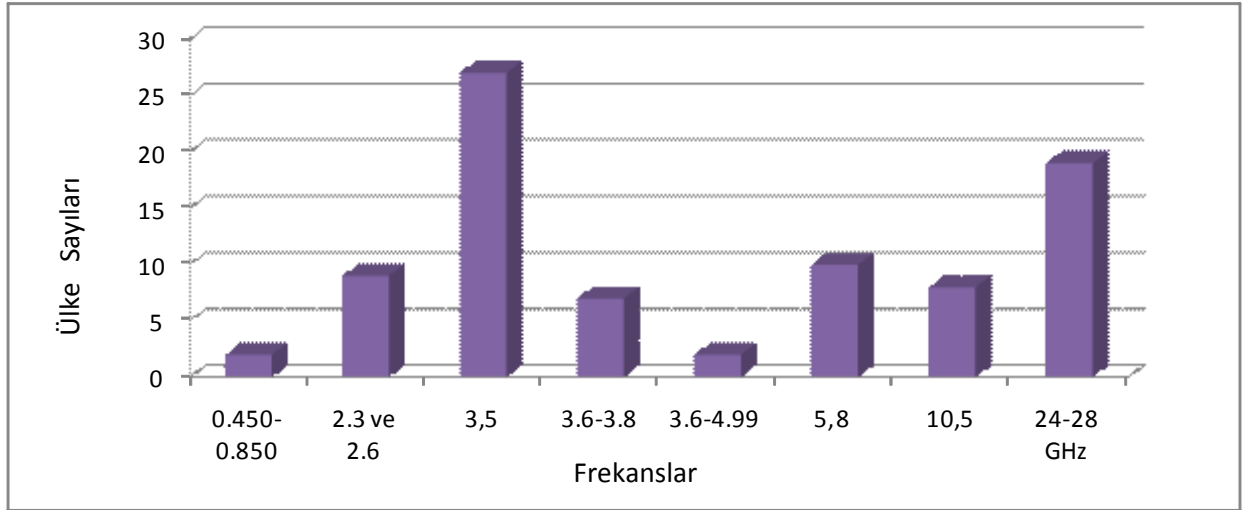
Teknoloji	Standart	Kullanımı	Hız (Mbps)	Mesafe	Frekans
UWB	802.15.3a	WPAN	110–480	10 m	7.5 GHz
Wi-Fi	802.11a	WLAN	Maks. 54	100 m	5 GHz
Wi-Fi	802.11b	WLAN	Maks. 11	100 m	2.4 GHz
Wi-Fi	802.11g	WLAN	Maks. 54	100 m	2.4 GHz
WiMAX	802.16d	WMAN	Maks. 75 (20 MHz Bant Genişliğinde)	6.5–10 km	<11 GHz
WiMAX	802.16e	Mobil WMAN	Maks. 30 (10 MHz Bant Genişliğinde)	1.5–5 km	2–6 GHz
WCDMA/ UMTS	3G	WWAN	Maks. 2 (HSDPA ile Maks. 10)	1.5–8 km	1800, 1900, 2100 MHz
CDMA2000 1xEV-DO	3G	WWAN	Maks. 2.4 (tipik 300–600 Kbps)	1.5–8 km	400, 800, 900, 1700, 1800, 1900, 2100 MHz
EDGE	2.5 G	WWAN	348 Kbps	1.5–8km	900, 1800, 1900 MHz

Tablo 2.4’de ise frekans ve WiMAX Forum tarafından belirlenmiş sertifikasyon profilleri gözlenmektedir[18].

Tablo 2.4: WiMAX Sertifikasyon Profilleri

Frekans (MHz)	Çiftleme	Kanallar (MHz)
2300-2400	TDD	5, 8.75, 10
2305-2320 2345-2360	TDD	5, 10
2496-2690	TDD	5, 10
3300-3400	TDD	5, 7, 10
3400-3800	TDD	5, 7, 10
Frekans (MHz)	Çiftleme	Kanallar (MHz)
3400-3600	TDD	3.5
3400-3600	FDD	3.5
3400-3600	TDD	7
3400-3600	FDD	7
5725-5850	TDD	10

Tablo 2.5: Kullanılan WiMAX Frekans Aralığı ve Ülke Sayıları



Tablo 2.5’de Dünya üzerinde WiMAX için kullanılan frekans aralıkları ve ülke sayıları gösterilmektedir. Ülkemizde ise 2,5 GHz, 3,5 GHz ve 5,8 GHz frekansları açısından durum şu şekilde özetlenebilir:

Ülkemizde 2500–2690 MHz bandı IMT–2000 kullanımına ayrılmış ve henüz kullanılmamaktadır.

5.8 GHz (5725–5875 MHz) frekans bandı ise askeri sistemlere tahsisli görülmektedir.

3.5 GHz bandının tamamı (3410–3600 MHz) ise mülga Telsiz Genel Müdürlüğü tarafından hem kırsal hem de kentsel alanda kullanılmak üzere 29.07.1998 tarihinde Türk Telekomünikasyon A.Ş.'ye (TTAŞ) tahsis edilmiş durumdadır. Ancak 2006 yılı sonuna kadar bandın boşaltılmasına ilişkin hukuki süreç halen devam etmektedir.

3. ÇOKLU-ORTAM AKTARIMI UYGULAMASI

Bu bölümde .NET platformu altında geliştirilen çoklu ortam aktarım uygulaması anlatılacaktır.

3.1. Uygulamaya Genel Bakış ve Bilgiler

Uygulamanın geliştirilme aşamasında kullanılan teknoloji ve süreci detaylandıran bilgiler bu bölüm altında incelenmiştir.

3.1.1 Soket programlama ve .NET mimarisi

TCP soket'lerinde bağlantı türleri soketin özelliğine göre değişir. TCP iletişimde iki ana kavram vardır. Biri bağlantılı, diğeri bağlantısız iletişimdir. Bağlantılı iletişimde soket' ten mesaj bekleyen (listen komutu ile soketi dinleyen) programa bağlantı, “connect” komutu ile yapılır. Bağlantı sağlandıktan sonra mesaj gönderme ve alma işlemi oku (read) ve yaz (write) komutları ile gerçekleşir. Bağlantılı iletişimde kullanılan soket türü SOCK_STREAM olmalıdır. TCP/IP üzerinden iletişimde, hattın yoğunluğu ve kalitesine bağımlı olarak parçalanmış TCP paketleri bu soket türü kullanıldığında alıcı soket altında sıralanır ve birleştirilerek TCP katmanına ulaştırılır. Bu durumda uygulama programı, bir read komutu ile tüm mesajı bir seferde okuyabilir. Kısa mesajlar SOCK_DGRAM soket üzerinden iletilir. Bu tür mesajların parçalanması söz konusu olmadığından yukarıdaki sorun yaşanmaz. Bu tür mesajlarda mesaj boyu 256 byte ya da daha kısa olmalıdır. Ve mesajlar hep aynı uzunlukta olmalıdır. Bağlantısız iletişim ortamında kullanılır. Yani soket dinleyen program “accept” komutu kullanmaz, sokete bilgi gönderen de “connect” komutunu kullanmaz. Burada yalnız listen komutu vardır. Programlar karşılıklı "read" ve "write" komutu ile haberleşir. İletişim açısından çok güvenli bir soket haberleşmesi değildir. Daha güvenli ve bağlantılı iletişimde (connected) DATA_ DGRAM mesajları, SOCK_ SEQPACKET tipi soket' ler üzerinde iletilmelidir. Bir bilgisayar içindeki programlar arası iletişim SOCK_RAW tipi soket

ile kurulabilir. Bu tür soket'lerde Data Gram tipi mesajlar iletilir. Host cihazındaki bir TCP üst-katman kullanıcısı bir port numarası ile tanımlanır. Port değeri IP internet adresi ile birleşerek bir soket oluşturur. Bu değer internet boyunca tek olmalıdır. Bir soket çifti her bir uç-nokta bağlantısını tek olarak tanımlar. Örneğin,

Gönderici Soket = Kaynak IP Adresi + Kaynak Port Numarası

Alıcı soket = Hedef IP Adresi + Hedef Port Numarası

Bir bilgisayarda birden çok soket bulunabilir. Örneğin aynı anda hem telnet soketi hem de ftp soketi açık olabilmektedir. Soketleri birbirinden ayırmak ve istemci sunucu ikilisini birbiri ile buluşturmak için her soket programın Port numarası vardır. Örneğin ftp'nin port numarası 21'dir. Bir ftp istemci, ftp sunucunun 21. portta çalıştığını bildiğinden direk onunla temasa geçer. Telnet 23.'ü portta çalıştığından, telnet sunucu ile ftp sunucu karışmaz. 1–1024 arasındaki portlar standarttır ve yalnız root tarafından kullanılabilir. Normal kullanıcılar soket programları için 1023'ten büyük bir port numarası seçmeleri gerekir.

3.1.2 C# programlama dili ve Multi-Threading

Bilgisayar çalışırken aynı anda birden fazla program çalışabilir çalıştırabilir. Örnek olarak bir yandan programlama gerçekleştirirken bir yandan müzik dinleyebilir yada internet üzerinde gezinilebilir. Bunların dışında kullanılmasa da arka planda bir çok program çalışabilir. Bütün programların eş zamanlı çalışması bilgisayar için bir engel değildir. Bu Windows'un Multi-Tasking (Çok görevli) özelliğidir. Bu noktadan çıkışla .NET yapısı altındaki Multi-Threading özelliği açıklanabilir.

Örnek olarak, kelime işleme uygulamalarda çok sayfası olan belgeler açılmak istediğinde, bu belgenin açılma işlemi zaman almaktadır. İşte bu belgelerin açılma esnasında açılan ilk sayfada işlem yapamazsınız yani uygulama o anda kilitlidir. Oysaki uygulamada ilk sayfa açıktır. Bu tek iş kanallı (Single Thread) durumdadır. Ancak bazı kelime işlemcilerde ilk sayfa açıldığı anda işlem yapmanıza izin verir. Bu tür uygulamalara ise MultiTherading (Çok iş kanallılık) denir.

C# ile yapılan uygulamalarda birden fazla işlem bir arada ve aynı anda yapılabilmektedir. İşlemci aynı anda sadece bir işi yapabilir. Daha açık bir dille ifade

etmek gerekirse, çalışan 3 tane uygulama olduğunu varsayalım. İşlemci kullanarak aynı anda çalışıyormuş gibi görünse de durum çok farklıdır. Uygulamalardan biri %15 diğeri %20 gibi bir işlemci kapasitesine ihtiyaç duyabilir ve multi-threading özelliği bu uygulamaların eş zamanlı çalışabilmesine izin vermektedir. Uygulama her zaman işlemciyi kullanmaz, işlemci boş bırakılan zamanlarda RAM yada sabit disk gibi nesnelere kullanabilir. İşlem tamamlanıncaya kadar işlemciyi çoğu kez boş bırakabilir. Boş kalma anında işlemciyi bir başka uygulama kullanabilir. Kodlama açısından bir örnek verirsek; bir oyun yazılımında kullanıcının hareket ettirdiği bir adam ve bilgisayarın yönettiği, adamı yakalamaya çalışan böcekler olduğunu varsayalım. Böyle bir oyunu yapısal bir programlama dili ile yazarken ya adam ve böceklerin hızları aynı olacak ya da böcekler adamın hareket etme döngüsü içinde duruma bağlı olarak bir hareket edip bir etmeyecekler ve hızları adamın hızının yarısı olacak. Her şeyden önce böyle bir kodlama korkunç karışık ve verimsiz olur. Oysa çok kanallı programlama ile her bir robot (adam ve böcekler) ayrı programlanır ve farklı kanallardan çalıştırılırsa hem kodlama kolay, hem de program daha etkili ve hızlı olur.

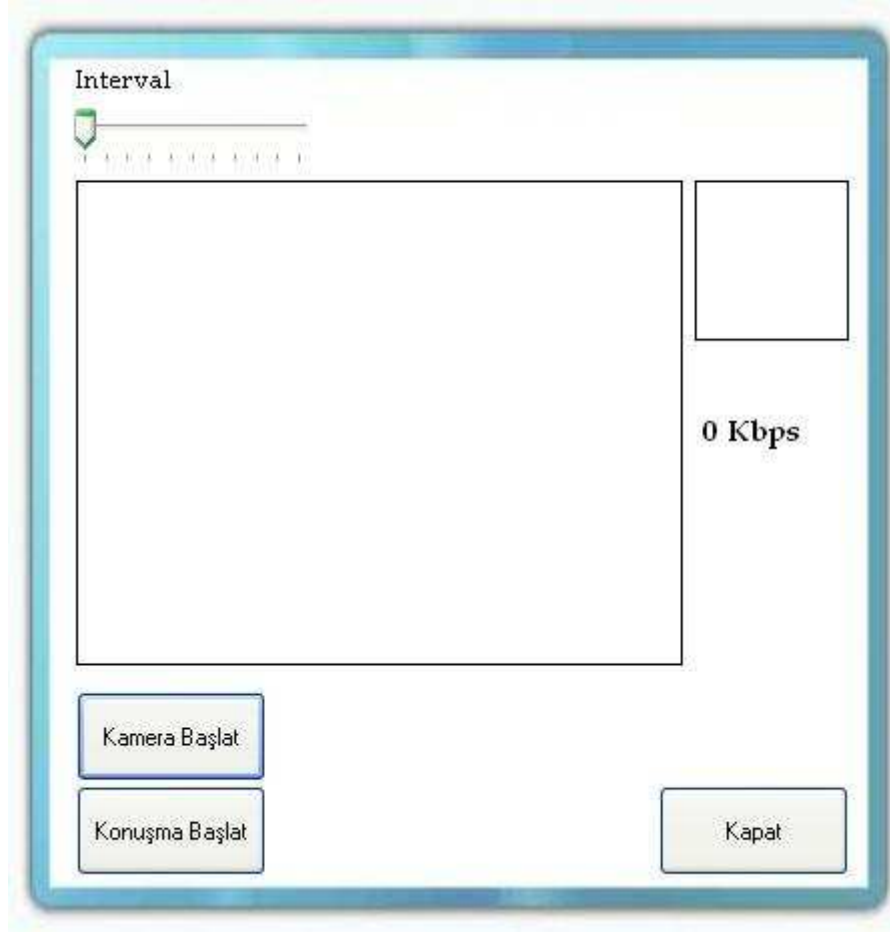
3.2 Uygulama Aşamaları ve Analizi

3.2.1 Giriş

Bu uygulama kablosuz ağda video konferans için tasarlanmıştır. Ancak tekil bir IP numarasına sahip modemi olan kullanıcıların internet üzerinden de iletişime geçmelerini mümkün kılmaktadır. Bunun haricinde “Router” veya “Switch” üzerinden internet ortamına açılan kullanıcıların, program üzerinde port çözümlemesi yapılmadığı için iletişime geçmeleri mümkün değildir. Bu kısımda program yapısı ve arayüzü detaylı olarak incelenmiştir.

3.2.2 Program arayüzü ve değişkenleri

Aşağıda Şekil 3.1’ de genel program arayüzü gösterilmektedir. Arayüz eşzamanlı olarak konuşma ve görüntü aktarımını başlatan butonlardan, görüntü pencerelerinden ve görüntü çerçeve sayısını belirleyen Interval değişkeninden oluşmaktadır.



Şekil 3.1: Program Arayüzü

Interval: Trackbar nesnesi üzerinde maksimum 25 üzerine set edilmiş değer istenildiği zaman düşürülebilir veya arttırılabilir. Dikkat edilmesi gereken nokta buradaki değişken direk olarak pencere sayısını değil, pencereler arasındaki gecikme süresini set etmektedir.

Konuşma Başlat: Ses alma ve gönderme işini yöneten thread' i başlatır.

Kamera Başlat: Görüntü gönderim ve alma görevini yürüten thread' i başlatır.

Kapat: Programdan çıkış ve tüm thread'lerin sonlandırılması görevini yürütür.

Görüntü Kutusu (Büyük) : İletişime geçilen bilgisayar tarafından gönderilen görüntü bilgisi bu alandan gözlenir.

Görüntü Kutusu (Küçük) : Kullanılan bilgisayar üzerinden alınan görüntü verisi bu alandan gözlenir.

3.2.3 Uygulamada kullanılan mesaj formatları

Uygulamada tek tip mesaj formatı 2 farklı şekilde kullanılmıştır. İlk mesaj görüntü, İkinci mesaj ise ses formatıdır.

USER:msj(ses),istemci listesi(Mesaj Yönü:client1IP1-client2IP1)

USER:msj(görüntü),istemci listesi(Mesaj Yönü:client1IP1-client2IP1)

USER:msj(ses),istemci listesi(Mesaj Yönü:client1IP2-client2IP1)

USER:msj(görüntü),istemci listesi(Mesaj Yönü:client1IP2-client2IP1)

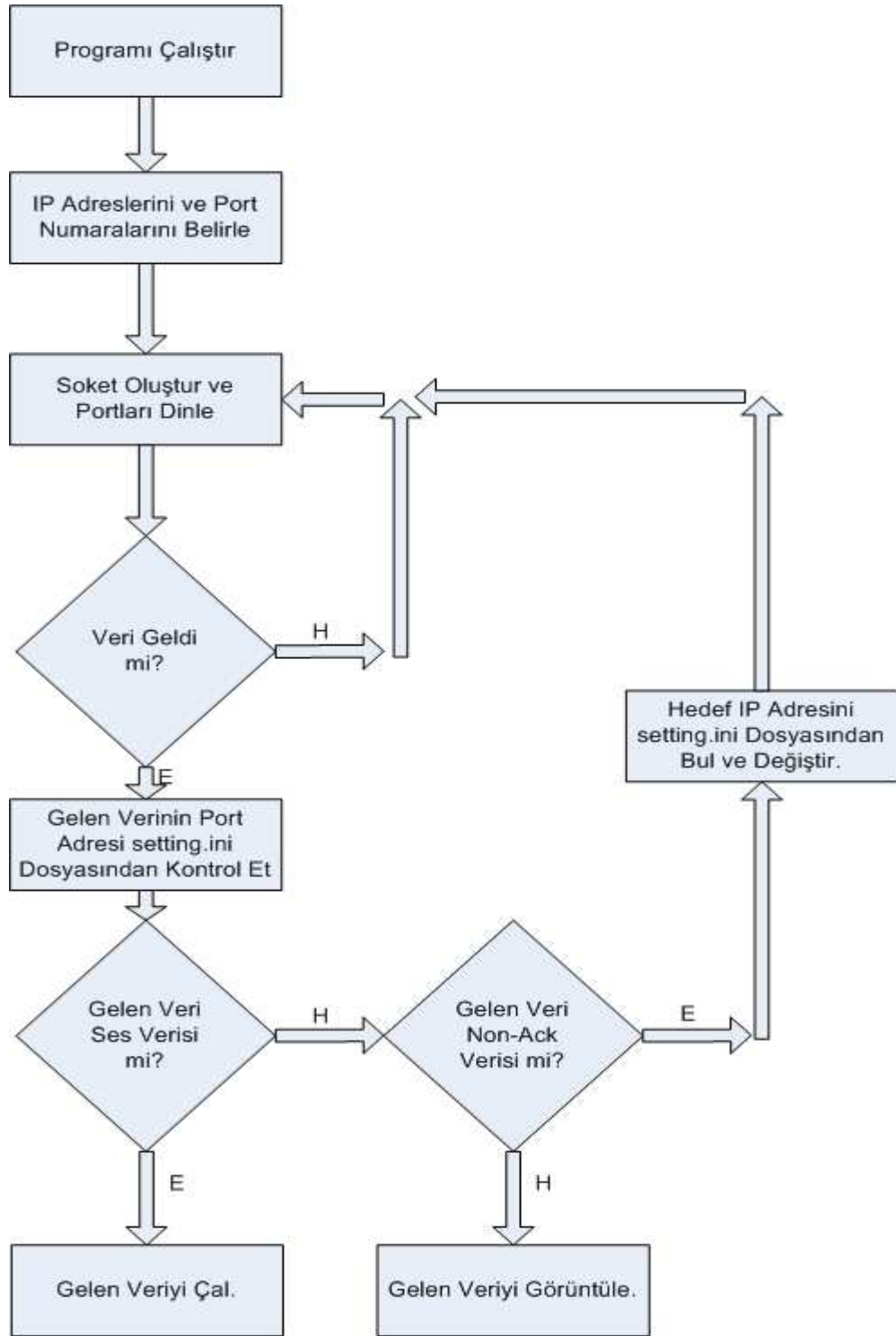
USER:msj(ses),istemci listesi(Mesaj Yönü:client1IP1-client2IP2)

USER:msj(görüntü),istemci listesi(Mesaj Yönü:client1IP1-client2IP2)

SCTP' nin multihoming işlevi, bu mesaj yapısı ile, etkin olarak kullanılan IP adreslerinden program kontrollü değişimle gerçekleştirilmektedir.

3.3 Uygulama Akış Diyagramı ve Kod Parçacıkları

3.3.1 Akış diyagramı



Şekil 3.2: Uygulama Akış Diyagramı

Şekil 3.2' de uygulama akış diyagramı sunulmuştur. Kod parçacıkları üzerinde detaylı açıklamaya geçmeden önce programın genel kapsamı, bu diyagram kullanılarak açıklanabilir.

Arayüz üzerinde kamera ve konuşma başlat butonuna basıldığında setting.ini dosyasına yazılmış olan hedef IP adresleri ve hedef-kaynak port numaraları program üzerinde gerekli değişkenlere atanır. Dinleme kısmını yöneten thread gelen mesajın içeriğine bakarak bunun hangi tipe ait olduğunu belirleyerek, ses yada görüntü çalma thread' ine atar. Eğer iletişim kesildiğine veri alınırsa, farklı bir deyişle ACK verisi alınmazsa, süreç hedef IP adresinin değiştirilmesi ile yeniden başlatılır.

3.3.2 Kod parçacıkları

Bu bölümde programı oluşturan önemli thread ve fonksiyon parçacıkları gösterilmektedir. Daha öncede bahsedildiği gibi yazılım farklı threadlerin birleşiminden oluşmaktadır. İlk olarak ses alma görevini yürüten thread aşağıdaki kod parçası ile oluşturulmaktadır.

```
t = new Thread(new ThreadStart(Voice_In));  
t.Priority = ThreadPriority.Highest;
```

Aşağıdaki kod parçası ise görüntü işlevini yürüten threaddir.

```
thrVideoThread = new Thread(new ThreadStart(StartCapture));  
thrVideoThread.Priority = ThreadPriority.Highest;
```

Aşağıdaki kod parçası ise settings.ini dosyası okuma görevini yürütmektedir.

```
private void ReadInitFile()  
{  
    if (!File.Exists("settings.ini"))  
    { MessageBox.Show("settings.ini bulunamadı");
```

```

        Application.Exit();
    }
    try
    {
        TextReader tr = new StreamReader("settings.ini");

        destinationIP1 = tr.ReadLine();

        VOutPort = tr.ReadLine();

        VInpPort = tr.ReadLine();

        TOutPort = tr.ReadLine();

        TInpPort = tr.ReadLine();

        destinationIP2 = tr.ReadLine();

        tr.Close();

        destinationIP = destinationIP1;

        r = new Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);
    }

```

Bu Kod kısmında ise çerçeveler arasındaki süreyi belirleyen trackbar kontrol ediliyor.

```

private void trackBar1_ValueChanged(object sender, EventArgs e)
{
    int val = 100;

    if (trackBar1.Value == 0)
    {
        return;
    }

    else
    {
        val = trackBar1.Value * 100;    }    tCapture.Interval = val; }

```

Eğer ilk bağlanan IP adresinde kopma yaşanırsa tanımlı diğer IP adresinden iletişimi sürdürmesini sağlayan genel fonksiyon aşağıda verilmiştir.

```
{
    if (firstPathInUse)
    {
        destinationIP = destinationIP2;
    }
    else
    {
        destinationIP = destinationIP1;
    }
    firstPathInUse = !firstPathInUse;
    r = new Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);
} }
```

Aşağıda gözlenen kod parçası ise belirlenen port adresine gelen veri miktarını gözlemleyerek bunu Kbps cinsinden ekrana yazdırmaktadır.

```
private void tBandwithMeter_Tick(object sender, EventArgs e)
{
    lblBWVal.Text = Convert.ToString((bytesRead / 1000) * 8) + " Kbps";
    bytesRead = 0;
}
}
```

Aşağıdaki kod parçası ise ses örnekleme ve veriyi iletme hazır hale getirmekle görevli kısımdır. Burada dikkat edilmesi gereken nokta örnekleme frekansının 11KHz ve kuantalamanın 8 bit ile gerçekleştirilmiş olmasıdır.

```
{ WaveFormat fmt = new WaveFormat(11000, 8, 2);
    m_Player = new WaveOutPlayer(-1, fmt, 8192, 3, new BufferFillEventHandler(Filler));
    m_Recorder = new WaveInRecorder(-1, fmt, 8192, 3, new
BufferDoneEventHandler(Voice_Out)); }
```



řekil 3.3: Uygulama Kullanımında

4. SONUÇ VE ÖNERİLER

Bu tez çalışmasında yeni gelişmekte olan ve 4. nesil haberleşme sistemlerine altyapı niteliği taşıyan WiMAX türü teknolojilerin kullanıldığı bir ortamda, çoklu ortam aktarımını gerçekleyen program .NET platformunda tasarlanmıştır.

Haberleşme sağlayacak olan iki bilgisayar birbirlerine ait IP (Internet Protocol) adreslerini ve veri aktarımını sağlayacak port adreslerini belirledikten sonra veri transferini başlatırlar. Program sunucu yapısı olmadan sadece bilgisayar – bilgisayar (Peer-to-Peer) yapısı üzerine kurulmuştur. Şu anda internet ortamında yaygın olarak kullanılan anlık mesajlaşma programlarına alternatif olarak kullanılabilir olan uygulamanın asıl önemli özelliği ise ikişer farklı IP adresleri üzerinden kurulacak olan iletişimde, herhangi bir hattın kopması durumunda iletişimin sürekliliğinin sağlanabilmesidir. Bu özellik, ağ yapısında kullanılan protokolleri sınıflayan OSI modelinde, taşıma katmanında (transport layer) yer alan TCP (Transmission Control Protocol) yerine, yine benzer bir protokol olan SCTP (Stream Control Transmission Protocol) multihoming özelliği kullanılarak gerçekleştirilmiştir.

Bu tezde üzerinde kısaca durulan ve temel özelliklerine vurgu yapılan WiMAX, 50 km.' ye varan alanda, 70Mbps hızlara varan erişim sağlayarak özellikle kablo ve DSL altyapısının ulaşamadığı alanlara yüksek hızlı internet servislerinin ulaştırılmasını sağlayacaktır. Yüksek erişim alanı sayesinde bir kaç baz istasyonu ile şehirleri kapsama alanına almak mümkün olacak, çok daha uzak mesafelere kablosuz ve hızlı internet erişimi vermek mümkün olacaktır.

Uygulamanın eksik olan port çözümüleme, kodlama ve filtreleme sistemi için kod parçacıklarının yada program bütünlerinin eklenmesi ile, uygulama daha verimli ve kullanılabilir hale getirilebilir.

KAYNAKLAR

- [1] Wireless Broadband - <http://wibro.or.kr/> (**Ziyaret Tarihi: 29 Mayıs 2007**).
- [2] Sundstrom, K., Rueda, A., and McLeod, R.D., "Internet Telephony compression algorithms", WESCANEX'97, winnepeg, MB, pp.13-18, (1998).
- [3] Öztürk, E., "WLAN Kablosuz Yerel Alan Ağları (Wireless Local Area Networks) Teknolojisinin İncelenmesi, Mevcut Düzenlemelerin Değerlendirilmesi ve Ülkemize Yönelik Düzenleme Önerisi", Uzmanlık Tezi, *Telekomünikasyon Kurumu*, (2006).
- [4] Özdemir, M., "Wireless LAN Technology & Security Update", *Cisco Systems Inc.*, s.3, (2003).
- [5] Richardson, M., and Ryan, P., "WiMAX: Opportunity or Hype?", *Proceedings of the Fourth Annual ITERA Conference*, ITERA (2006).
- [6] Sriram, K., and Whitt, W., "Characterizing superposition arrival processes in packet multiplexers for voice and data", *IEEE J.Select Areas Commun.*, vol.SAC-4, pp. 833-846, (1986).
- [7] "Real Time Control Protocol attribute in Session Description Protocol", *IETF Recommendation RFC 3605*, (2003).
- [8] *IEEE 802.16 protocols* - http://en.wikipedia.org/wiki/IEEE_802.16 (**Ziyaret tarihi: 27 Mayıs 2008**).
- [9] http://en.wikipedia.org/wiki/Stream_Control_Transmission_Protocol (**Ziyaret Tarihi: 25 Mayıs 2008**).
- [10] R. Stewart *et al.*, *RFC 2960: Stream Control Transmission Protocol*, IETF, October 2000. <http://www.ietf.org/rfc/rfc2960.txt?number=2960>, (**Ziyaret Tarihi: 25 Mayıs 2008**).
- [11] R. Stewart and Q. Xie, "Stream Control Transmission Protocol (SCTP): A Reference Guide", *Addison Wesley*, (2002).
- [12] Fitzpatrick, J., Murphy, S. and Murphy J., "An Approach to Transport Layer Handover of VoIP over WLAN" *Prentice Hall*, (2000).
- [13] "Design Principles for Voice over WLAN"- *Cisco White Paper*
- [14] "ITU, International Telecommunication Union, Radio Regulations, CHAPTER II – Frequencies", *ARTICLE 5, Frequency allocations*, p.72, (2001).

- [15] Hella, M. and Ismail, M. "RF CMOS Power Amplifiers: Theory, Design and Implementation", *Springer*, (2001).
- [16] "IEEE 802.16e WiMAX OFDMA Signal Measurements and Troubleshooting" - *Agilent Technologies* - Application Note 1578
- [17] Minoli, D., "Telecommunications Technology Handbook, Chapter 8, Wireless Technologies: WPAN, WLAN, and WWAN", Second Edition, *Artech House, Boston London*, s.245-335 (2003).
- [18] Nurflus P., Envara Com, "Silicon Solutions for 5 GHz Wireless Networking, Wireless Networking In The 5 GHz Band" - *The Real Benefits A Technology White Paper*
- [19] Juan, H., and Hsiang-Chun "Scalable Video Streaming over Mobile WiMAX" *IEEE International Symposium on Circuits and Systems*, 2007. ISCAS 2007, pp.3463-3466, (27-30 May 2007)
- [20] Ramadin, D., K., "Overview of Wireless Broadband Technologies", *Intel Corporation*
- [21] Prasad R., Muñoz L., "WLANs and WPANs towards 4G Wireless, Fundamentals of WLAN", *Artech House Boston, London*, s. 133,134 (2003).
- [22] Leland, W. , Taqqu, M., Willinger, W., and Wilson, D., "On the self-similar nature of ethernet traffic", *IEEE/ACM Trans .Netw.,vol.2,no.1,pp* (1-15, Feb, 1994).
- [23] Black, U., "Data Networks", *Concepts, Theory and Practice. Englewood Cliffs, Prentice Hall*, New Jersey, (1989).
- [24] http://ist-trike.org/public/STRIKE_WP7_RWTH_ComNets_IN_Int_007.pdf (Ziyaret tarihi: 27 Mayıs 2008).
- [25] Wang, S., Xuan, D., Bettati, R., and Zhao, W., "Differentiated Services with Statistical Real -Time Guarantees in Static-Priority Scheduling Networks" , *Proc.IEEE Real -Time Systems Symp.*, Dec. , (2001).
- [26] Ohrtman, F., "WiMAX Handbook", *McGraw-Hill*, (2005).
- [27] YunSeong, E. and KwangDu, L., "A 2.4GHz/5.2GHz CMOS power amplifier for dual-band applications", *2004 IEEE MTT-S International, Microwave Symposium Digest*, Volume 3, (June, 2004)
- [28] Jeffries, R., "Extreme Programming Adventures in C#", *Microsoft Publications* (2004)

EK-1

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.IO;

using System.Threading;

using System.Net.Sockets;

using Voice;

using System.Net;

namespace EcoVCON
{
    public delegate void GetPicSizeCallBack();

    public partial class Form1 : Form
    {
```

```
public Form1()
{
    InitializeComponent();

    CheckForIllegalCrossThreadCalls = false;

    t = new Thread(new ThreadStart(Voice_In));
    t.Priority = ThreadPriority.Highest;
}

private void btExit_Click(object sender, EventArgs e)
{
    this.Close();
}

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    try
    {
        thrVideoThread.Abort();
    }
}
```

```
catch { }

try
{
    mytcpl.Stop();
}

catch { }

try
{
    ns.Flush();
}

catch { } try
{
    ns.Close();
}

catch { }

try
{
    t.Abort();
} catch { }

try{
    r.Close(); } catch { }
```

```

try{ Stop(); } catch { }

try{ myclient.Close(); } catch { }

try{ ms.Close(); } catch { }

try{ myns.Close(); } catch { }

try { mysw.Close(); }

catch { }

}

private void trackBar1_ValueChanged(object sender, EventArgs e)
{
    int val = 100;

    if (trackBar1.Value == 0)
    {
        return;
    }

    else
    {
        val = trackBar1.Value * 100;
    }
}

```

```
tCapture.Interval = val;
}

private Thread thrVideoThread;

private bool panelLeftClicked;

private int mlX;

private int mlY;

private string destinationIP;

private string VOutPort;

private string VInPort;

private string TOutPort;

private string TInPort;

private string destinationIP2;

private string destinationIP1;

private bool firstPathInUse = true;

private TcpClient myclient;

private MemoryStream ms;
```



```

private NetworkStream myns;

private BinaryWriter mysw;

private TcpListener mytcpl;

private Socket mysocket;

private NetworkStream ns;

private long bytesRead;

private WaveOutPlayer m_Player;

private WaveInRecorder m_Recorder;

private FifoStream m_Fifo = new FifoStream();

private Socket r;

private Thread t;

private bool connected = false;

private byte[] m_PlayBuffer;

private byte[] m_RecBuffer;

private void panel1_MouseDown(object sender, MouseEventArgs e)
{
    panelLeftClicked = true;

    m1X = e.X;

    m1Y = e.Y;
}

```

```

private void panel1_MouseUp(object sender, MouseEventArgs e)
{
    panelLeftClicked = false;
}

private void panel1_MouseMove(object sender, MouseEventArgs e)
{
    if (panelLeftClicked)
    {
        this.Location = new Point(this.Location.X + (e.X - mlX), this.Location.Y +
(e.Y - mlY));
    }
}

private void Form1_Load(object sender, EventArgs e)
{
    ReadInitFile();

    thrVideoThread = new Thread(new ThreadStart(StartCapture));
    thrVideoThread.Priority = ThreadPriority.Highest;
    thrVideoThread.Name = "VideoThread";
    thrVideoThread.Start();
}

```

```
        bytesRead = 0;

        tBandwithMeter.Start();
    }

private void ReadInitFile()
{
    if (!File.Exists("settings.ini"))
    {
        MessageBox.Show("settings.ini bulunamadı");
        Application.Exit();
    }

    try
    {
        TextReader tr = new StreamReader("settings.ini");

        destinationIP1 = tr.ReadLine();

        VOutPort = tr.ReadLine();

        VInpPort = tr.ReadLine();

        TOutPort = tr.ReadLine();

        TInpPort = tr.ReadLine();
    }
}
```

```

destinationIP2 = tr.ReadLine();

tr.Close();

destinationIP = destinationIP1;
}
catch
{

}
}

private void StartCapture()
{
try
{
mytcppl = new TcpListener(int.Parse(VInpPort));
mytcppl.Start();
mysocket = mytcppl.AcceptSocket();
ns = new NetworkStream(mysocket);

//System.Diagnostics.Debug.Assert(!this.InvokeRequired,
"InvokeRequired");

```

```
pictureBox2.Image = Image.FromStream(ns);
```

```
this.Invoke(new GetPicSizeCallBack(GetPicSize));
```

```
mytcp.Stop();
```

```
if (mysocket.Connected == true)
```

```
{
```

```
    while (true)
```

```
    {
```

```
        StartCapture();
```

```
    }
```

```
}
```

```
    myns.Flush();
```

```
}
```

```
catch (Exception)
```

```
{
```

```

        if (firstPathInUse)
        {
            destinationIP = destinationIP2;
        }
        else
        {
            destinationIP = destinationIP1;
        }

        firstPathInUse = !firstPathInUse;

        r = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
ProtocolType.Udp);
    }
}

private void GetPicSize()
{
    MemoryStream m = new MemoryStream();

    pictureBox2.Image.Save(m, System.Drawing.Imaging.ImageFormat.Jpeg);

    bytesRead += m.Length;
}

```

```

private void tCapture_Tick(object sender, EventArgs e)
{
    StartSendingVideo(destinationIP, Int16.Parse(VOutPort));
}

```

```

private void StartSendingVideo(string remote_IP, int port_number)
{
    try
    {
        ms = new MemoryStream();

        pictureBox1.Image.Save(ms,
System.Drawing.Imaging.ImageFormat.Jpeg);

        byte[] arrImage = ms.GetBuffer();

        myclient = new TcpClient (remote_IP,port_number);

        myns = myclient.GetStream ();

        mysw = new BinaryWriter (myns);

        mysw.Write(arrImage);

        ms.Flush();

        mysw.Flush();

        myns.Flush();

        ms.Close();

        mysw.Close ();

        myns.Close ();
    }
}

```

```

        myclient.Close ();
    }
    catch (Exception ex)
    {
        //tCapture.Enabled = false;

        if (firstPathInUse)
        {
            destinationIP = destinationIP2;
        }
        else
        {
            destinationIP = destinationIP1;
        }

        firstPathInUse = !firstPathInUse;

        r = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
        ProtocolType.Udp);
    }
}

private void button1_Click(object sender, EventArgs e)
{

```



```

        this.webCamCapture1.TimeToCapture_milliseconds = 1;

        this.webCamCapture1.Start(0);

        tCapture.Enabled = true;
    }

    private void webCamCapture1_Load(object sender, EventArgs e)
    {

    }

    void webCamCapture1_ImageCaptured(object source,
    WebCam_Capture.WebcamEventArgs e)
    {
        this.pictureBox1.Image = e.WebCamImage;
    }

    private void tBandwithMeter_Tick(object sender, EventArgs e)
    {
        lblBWVal.Text = Convert.ToString((bytesRead / 1000) * 8) + " Kbps";

        bytesRead = 0;
    }

    private void button2_Click(object sender, EventArgs e)

```

```

{
    if (connected == false)
        {
            t.Start();
            connected = true;
        }

    Start();
}

private void Start()
{
    Stop();

    try
    {
        WaveFormat fmt = new WaveFormat(11000, 8, 2);

        m_Player = new WaveOutPlayer(-1, fmt, 8192, 3, new
BufferFillEventHandler(Filler));

        m_Recorder = new WaveInRecorder(-1, fmt, 8192, 3, new
BufferDoneEventHandler(Voice_Out));

    }

    catch

    {

```

```

    if (firstPathInUse)
    {
        destinationIP = destinationIP2;
    }
    else
    {
        destinationIP = destinationIP1;
    }

    firstPathInUse = !firstPathInUse;

    r = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
ProtocolType.Udp);

    //Stop();

    //throw;

}

}

private void Stop()
{
    if (m_Player != null)
    try
    {
        m_Player.Dispose();
    }
}

```

```

    }

    finally

    {

        m_Player = null;

    }

    if (m_Recorder != null)

        try

        {

            m_Recorder.Dispose();

        }

        finally

        {

            m_Recorder = null;

        }

        m_Fifo.Flush(); // clear all pending data
    }

private void Voice_Out(IntPtr data, int size)

{

    try

    {

        //for Recorder

```

```

if (m_RecBuffer == null || m_RecBuffer.Length < size)

    m_RecBuffer = new byte[size];

System.Runtime.InteropServices.Marshal.Copy(data, m_RecBuffer, 0,
size);

//Microphone ==> data ==> m_RecBuffer ==> m_Fifo

r.SendTo(m_RecBuffer, new IPEndPoint(IPAddress.Parse(destinationIP),
int.Parse(TOutPort)));

}

catch

{

    if (firstPathInUse)

    {

        destinationIP = destinationIP2;

    }

    else

    {

        destinationIP = destinationIP1;

    }

    firstPathInUse = !firstPathInUse;

    r = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
ProtocolType.Udp);

}

```

```
}
```

```
private void Voice_In()
```

```
{
```

```
    try
```

```
    {
```

```
        byte[] br;
```

```
        r.Bind(new IPEndPoint(IPAddress.Any, int.Parse(TInpPort)));
```

```
        while (true)
```

```
        {
```

```
            br = new byte[8192];
```

```
            r.Receive(br);
```

```
            m_Fifo.Write(br, 0, br.Length);
```

```
        }
```

```
    }
```

```
    catch
```

```
    {
```

```
        if (firstPathInUse)
```

```
        {
```

```
            destinationIP = destinationIP2;
```

```
        }
```

```

else
{
    destinationIP = destinationIP1;
}

firstPathInUse = !firstPathInUse;

r = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
ProtocolType.Udp);
}
}

private void Filler(IntPtr data, int size)
{
    if (m_PlayBuffer == null || m_PlayBuffer.Length < size)
        m_PlayBuffer = new byte[size];

    if (m_Fifo.Length >= size)
        m_Fifo.Read(m_PlayBuffer, 0, size);

    else

        for (int i = 0; i < m_PlayBuffer.Length; i++)
            m_PlayBuffer[i] = 0;

    bytesRead += size;

    System.Runtime.InteropServices.Marshal.Copy(m_PlayBuffer, 0, data, size);
}

```

```
        // m_Fifo ==> m_PlayBuffer==> data ==> Speakers
    }

    private void tVoice_Tick(object sender, EventArgs e)
    {
    }

}

}
```


ÖZGEÇMİŞ

03.03.1982 tarihinde Ankara'da doğdu. İlk,orta ve lise öğrenimini Kocaeli' de tamamladı. 2000 yılında girdiği Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliği bölümünden 2004 yılında mezun oldu. 2005 yılı süresince askerlik eğitimini tamamlayarak 2006 yılında İstanbul V.O.S.S. Varinak A.Ş. bünyesinde teknik servis mühendisi olarak göreve başladı. 2007 yılından beri aynı şirketin Bilgi İşlemleri Teknik Destek bölümünde görevini sürdürmektedir. 2005 yılından beri Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı' nda yüksek lisans eğitimine devam etmektedir.