

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ÜÇÜNCÜ NESİL MOBİL UMTS AĞLARINDA
NOKTADAN NOKTAYA HİZMET KALİTESİ VE
PERFORMANS ÖLÇÜMÜ**

Bilgisayar Müh. Çiğdem AK

**FBE Bilgisayar Mühendisliği Anabilim Dalında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Prof. Dr. A. Coşkun SÖNMEZ

İSTANBUL, 2010

İÇİNDEKİLER

İÇİNDEKİLER	ii
SİMGE LİSTESİ	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ.....	vi
ÇİZELGE LİSTESİ	vii
ÖNSÖZ.....	viii
ÖZET	ix
ABSTRACT	x
1.GİRİŞ.....	1
2.UMTS AĞININ ÖZELLİKLERİ	3
2.1 UMTS Şebeke Yapısı	5
2.2 Baz İstasyon Alıcı-Vericisi (BTS-Base Transceiver Station)	6
2.3 Radyo Şebeke Denetleyicisi (RNC-Radio Network Controller).....	7
2.4 UTRAN (UMTS Terrestrial Radio Access Network)	7
2.5 NODE B	9
3.ÖNERİLEN ÖLÇÜM METODOLOJİSİNİN DAYANAKLARI.....	10
3.1 UMTS Ağında Protokoller	10
3.2 Performans Ölçümü İçin Platform Tercihleri.....	11
3.2.1 Ölçüm Yazılımından Beklenenler	11
3.2.2 Tercih Edilen Donanım Platformu	11
3.2.3 Tercih Edilen Yazılım Platformu	12
3.2.4 Donanım Platformunun Yapısı ve Yazılıma Sağlanan Arayüz	13
4.HİZMET KALİTESİ (QoS)	16
4.1 Gecikme Bütçesi (Delay Budget)	17
4.2 Gecikme (Delay) ve Geri Dönüş Süresi (Round Trip Time).....	18
4.3 Bant Genişliği (Bandwidth) ve Akış Hacmi (Throughput)	18
4.4 Jitter	20
4.5 Sırasız Teslim (Out of Order Delivery)	21
4.6 Paket Kaybı (Packet Loss).....	22
5.ÖNERİLEN ÖLÇÜM METODOLOJİSİ.....	24
5.1 Donanım Platformu Önerimizin Dayanakları.....	24
5.2 Geliştirilen Ölçüm Yazılımının Donanımla Etkileşimi	25
5.3 Geliştirilen Yazılımın Yapısı.....	26
6.KAYNAK KODUNUN MİMARİSİ.....	27
6.1 Yansıtma Sunucusunun Kod Yapısı	27
6.1.1 TCP Dinleyici İş Parçacığı Sınıfı - tcpListenerThread.....	28
6.1.2 TCP İş Parçacığı Sınıfı - tcpThread.....	29
6.1.3 UDP İş Parçacığı Sınıfı - udpThread.....	29
6.2 Ölçüm İstemcisinin Kod Yapısı	30
6.2.1 ICMP Echo Taleplerinin Üretilmesi ve Gözlemlenmesi	30

6.2.2	TCP Akışı Yaratımı	33
6.2.3	UDP Akışı Yaratımı	36
6.2.4	TCP, UDP ve ICMP ile Yapılabilen Ölçüm Türlerinin Karşılaştırması	38
6.2.5	Paket Koklama (Packet Sniffer) ile Ekstra Veri Elde Edilmesi	38
7.	VERİLERİN İNCELENMESİ	40
7.1	Veri eksenleri ve matematiksel model.....	40
7.2	Ölçüm Sonuçları	41
7.2.1	A Operatörü	41
7.2.2	B Operatörü	44
7.2.3	C Operatörü	45
7.2.4	Üç Operatörün Karşılaştırılması	46
7.2.5	Konumsal Matematik Model	48
7.2.6	3G-İnternet Ağ Geçidi	52
7.2.7	Kullanıcı Ekipmanının (UE) Modem Olarak Performansı.....	53
7.2.8	UMTS Spesifik Baz İstasyonları	54
8.	SONUÇ VE ÖNERİLER	56
	KAYNAKLAR.....	57
	İNTERNET KAYNAKLARI	58
	EKLER	59
Ek 1a	Program giriş noktası main() metodu	60
Ek 1b	tcpListenerThread Sınıfı	60
Ek 1c	tcpThread Sınıfı	60
Ek 1d	udpThread Sınıfı	61
Ek 1e	Sniffing tuşuna tıklanması durumunda çalışan kod bloğu	61
Ek 1f	Paket koklayıcı iş parçacığına ait ana fonksiyon.....	62
Ek 1g	Yakalanan paketlerin DNS çözümünü yapan ve kuyruğa ekleyen kod bloğu	63
	ÖZGEÇMİŞ.....	64

SİMGE LİSTESİ

$A(x,y)$	Modele göre (x,y) konumu için öngörülen değeri veren fonksiyon
$g_{1...n}$	İndise göre n adet ölçüm sonucu
K	Uzaklık fonksiyonunun karesinin tersi olarak tanımlı katsayı fonksiyonu
$U(x,y,i,j)$	Pisagor teoremine göre kartezyen uzayda uzaklık fonksiyonu
λ	Ölçüm sonucu g 'lerin ortalaması

KISALTMA LİSTESİ

3G	Third Generation
3GPP	The 3rd Generation Partnership Project
ABR	Average Bit Rate
API	Application Programming Interface
ATM	Asynchronous Transfer Mode
CBR	Constant Bit Rate
CDMA	Code division multiple access
CLR	Common Language Runtime
DLL	Dynamic Loading Library
EDGE	Enhanced Data Rates for GSM Evolution
FDD	Frequency Division Duplex
FEC	Forward Error Correction
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
IP	Internet Protocol
JRE	Java Runtime Environment
MIDP	Mobile Information Device Profile
MSIL	Microsoft Intermediate Language
NIC	Network Interface Card
NODE B	B D�ğ�m� (UMTS Baz İstasyonu)
POTS	Plain Old Telephony System
PPP	Point to Point Protocol
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RNC	Radio Network Controller
RSVP	Reservation Protocol
RTP	Real-time Transport Protocol
SAR	Specific Absorption Rate
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TDD	Time Division Duplex
TTL	Time to Live
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
VBR:	Variable Bit Rate
VOIP	Voice Over IP
W-CDMA	Wideband Code Division Multiple Access

ŞEKİL LİSTESİ

	Sayfa
Şekil 2.1 GSM sistemlerine ait transmisyon hızları	4
Şekil 2.2 UMTS erişim şebeke mimarisi	6
Şekil 2.3 UMTS Şebekesinde ATM ağ altyapısı	8
Şekil 3.1 TCP/IP'nin ATM paketleri ile kapsüllenmesi	10
Şekil 4.1 Hizmet kalitesi öncelik seviyeleri	17
Şekil 4.2 Ses İletiminde Gecikme Kaynakları	18
Şekil 4.3 Trafik şekillendirme	19
Şekil 4.4 Jitter	21
Şekil 4.5 İleri Hata Düzeltme (Forward Error Correction – FEC)	22
Şekil 6.1 Sunucu yazılımının sınıf diyagramı	27
Şekil 6.2 IP ve içindeki ICMP paketinin alanları	30
Şekil 6.3 ICMP paketlerinin sınıf diyagramı	32
Şekil 6.4 ICMP Echo talepleriyle A operatörü için gecikme (delay) ölçümü	33
Şekil 6.5 TCP State Diagram	34
Şekil 6.6 TCP Paket Başlığı	35
Şekil 6.7 Kaynak kodunda TCP paketine ait sınıf diyagramı	36
Şekil 6.8 IP ile beraber UDP paket başlığı	37
Şekil 6.9 UDP paketi sınıf diyagramı	37
Şekil 7.1 A Operatörü için gecikme ve paket kaybı	41
Şekil 7.2 A operatörü için kullanıcı ekipmanı üzerinden bant genişliği ve gecikme ölçümü ..	42
Şekil 7.3 A operatöründe veri aktarımı sırasında ses görüşmesinin gecikmeye etkisi	43
Şekil 7.4 A operatörü konumsal ölçüm sonuçları (kbit/sn)	43
Şekil 7.5 B operatöründe gecikme ve veri aktarımı sırasında ses görüşmesinin etkisi	44
Şekil 7.6 B operatörü için kullanıcı ekipmanı üzerinden bant genişliği ve gecikme ölçümü ..	45
Şekil 7.7 B operatörü konumsal ölçüm sonuçları (kbit/sn)	45
Şekil 7.8 C operatörü konumsal ölçüm sonuçları (kbit/sn)	45
Şekil 7.9 Operatörlerin bant genişliği performansının zamansal karşılaştırması	46
Şekil 7.10 Operatörlerin bant genişliği performansının mekansal karşılaştırması	47
Şekil 7.11 A operatöründe beş noktaya göre hesaplanmış ekstrapolasyon ızgarası	50
Şekil 7.12 Ölçümlerin yapıldığı bölge	50
Şekil 7.13 A operatörü için gecikme ızgarası ve haritanın süperpozisyonu	51
Şekil 7.14 A operatörünün Internet bulutu tarafından gözüken ağ geçidi	52
Şekil 7.15 Kullanıcı ekipmanının işletim sistemine bildirdiği kapasite	53
Şekil 7.16 Bilgisayarın internete çıkışında UMTS kullandığının teyidi	54

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 2.1 2. ve 3. nesil teknolojilerin karşılaştırmalı çizelgesi	5
Çizelge 4.1 ATM hizmet kategorilerinin tipik kullanım alanları	20
Çizelge 4.2 ATM hizmet kategorilerinin karakteristikleri	20
Çizelge 6.1 Protokollerin QoS parametrelerine göre ölçüm yeteneği	38

ÖNSÖZ

Yakın dönemde dünya çapında mobil telekom operatörleri üçüncü nesil paket veri iletişimi hizmetleri ve altyapılarına büyük yatırımlar yaparak bu hizmetleri halka sunmaya başlamışlardır. Diğer yandan devrim niteliğinde mobil veri erişimi vadeden bu teknolojilerin teorik performansları ile saha uygulamalarındaki performansları büyük farklılıklar göstermektedir. Bu farklılıklar özellikle ülkeler ve operatörler arasında gözlemlense de, aynı operatörün kapsama alanı içinde de zaman ve mekana bağlı olarak hizmet kalitesi ve performans dalgalanmaları bulunmaktadır.

Bu çalışmanın amacı, bu dalgalanmaları kabul görmüş hizmet kalitesi değişkenleri cinsinden, bahsi geçen zaman, mekan ve operatör eksenlerinde inceleyebilmek için bilimsel bir ölçüm metodolojisi ve aracı önermektir. Ayrıca bir diğer amaç da, önerilen bu sistem ile elde ettiğimiz sınırlı miktardaki veri yardımıyla, yeterli sayıda ölçümün yapılması halinde verilen rasgele bir konum ve zaman için oluşacak fiziki durumu öngörebilecek matematiksel bir benzetim modelinin içerik olarak doğrulanmasıdır.

Bu tez çalışmasında danışmanım olan, yüksek lisans eğitimim süresince ve tez çalışmam sırasında bana göstermiş olduğu büyük anlayış ve her türlü yardımlarından dolayı derinden saygı ve sevgi duyduğum değerli hocam Prof. Dr. A. Coşkun SÖNMEZ'e ve ölçümler sırasındaki yardımlarından dolayı mütefekkir arkadaşım Hakan AYRAL'a tüm kalbimle teşekkür ederim.

ÖZET

ÜÇÜNCÜ NESİL MOBİL UMTS AĞLARINDA NOKTADAN NOKTAYA HİZMET KALİTESİ VE PERFORMANS ÖLÇÜMÜ

ÇİĞDEM AK

Bilgisayar Mühendisliği, Yüksek Lisans Tezi

UMTS gibi üçüncü nesil (3G) paket anahtarlama mobil ağların her geçen gün daha fazla operatör tarafından kullanılmaya başlaması beraberinde önceki 2.5G (GPRS) ve 2.75G (Edge) ağların spesifikasyonlarında özel olarak yer almayan hizmet kalitesi (QoS) ve performans taahhütü unsurlarını karasal geniş bant hizmet sağlayıcılarından hücre bazlı mobil telefon operatörlerine taşımıştır. Her ne kadar 3G spesifikasyonlarını hazırlayan çalışma grubu 3GPP bu konuda kesin sınırlar belirlemiş olsa da, saha uygulamasında çevresel şartlar ve çekirdek ağın kurulumundaki tercihler sebebiyle beklenen performans elde edilemeyebiliyor. Bu sebepler ışığında, sahada iki nokta arasında hizmet kalitesi ve ağ performansının ölçümünün sadece operatör değil bağımsız üçüncü taraflarca da yapılabilmesi önem arz etmektedir.

Bu çalışmada önerilen sistem, karasal ağ hizmetlerinin bu tür ölçümlerinde hali hazırda kullanılmakta olan kabul görmüş yazılımsal çözümlerin ve bu amaçla kendi geliştirdiğimiz yazılımın, uygun donanımsal modifikasyonla mobil noktadan mobil noktaya (point to point) yada mobil noktadan internet bulutuna (point to net) UMTS ağı üzerinde erişimde QoS kriterlerini oluşturan parametreleri ölçmesi esasına dayanmaktadır.

Aynı operatör ağı içinde kalmak ile operatör ağından internet bulutuna çıkmak arasındaki ölçüm farklarına dayanarak, operatörlerin internet çıkışı sağlayan gateway'lerinin QoS parametreleri değerlendirilmiş ve elde edilen deneysel sonuçlar yardımıyla farklı 3G hizmeti veren operatörlerin performansları ve hizmet kaliteleri karşılaştırılmıştır.

Anahtar kelimeler: 3G, GSM, UMTS, Mobil, Network, Hizmet Kalitesi, QoS, Gecikme, Bant Genişliği, Paket Kaybı, Değişken Gecikme, Performans Değerlendirme.

ABSTRACT

POINT TO POINT QUALITY OF SERVICE AND PERFORMANCE EVALUATION FOR THIRD GENERATION MOBILE UMTS NETWORKS

ÇİĞDEM AK

Computer Engineering, M.S. Thesis

As the third generation packet switching networks like UMTS are more and more adopted by many carrier providers the quality of service (QoS) which has not been part of the specifications of previous 2.5G (GPRS) and 2.75G (EDGE) has also become the concern of mobile carriers just like terrestrial broad band service providers. Although 3GPP workgroup which has prepared the specifications for 3G has defined strict limits on this subject, on the field implementation environmental conditions and managerial decisions about the setup of core network may lead to sub-optimal performance. In this context it becomes very important for independent 3rd parties to be able to perform performance measurements between two points without the help of the carrier provider.

The system that is proposed in this thesis, allows to use widely adopted open source measurements software and a specific software that we have developed with UMTS performance measurement in mind to be used on a hardware modified mobile computer for point to point connections over UMTS and for point to internet connections to be able to measure the QoS criteria parameters.

So, QoS parameters have been evaluated by getting to the internet through the gateway of carrier provider based on the measurement differences between mobile point to point communication in 3G network and mobile point to internet communication. Also the performance and service quality of each carrier provider has been compared in terms of QoS with the obtained experimental measurement results over 3G network.

Keywords: 3G, GSM, UMTS, Mobile, Network, Quality of Service, QoS, Delay, Bandwidth, Packet Loss, Jitter, Performance Evaluation.

1. GİRİŞ

Gezgin haberleşme sistemleri, 1980’lerde başlayan 1G ve 1990’lardan itibaren kullanılan 2G teknolojilerinden, ülkemizde yeni kullanılmaya başlanan 3G teknolojilerine uzanan ve hatta 4G teknolojilerinin tasarımının yapıldığı bir gelişim sürecine sahiptir (Büyükbaş, 2005).

1G olarak bilinen ilk hücresel ağ nesli sadece ses taşıyabilen analog sistemlerden oluşuyordu. 1990’larda aralarında GSM, TDMA ve CDMA’nın olduğu ikinci nesil (2G) ağlar piyasaya sürüldü. 2G ağlar, 1G ağlarının analog işletimlerini dijital işletimlerle değiştirdi. Böylece sesin yanında verilerinde kablosuz olarak iletilmesi sağlandı. Taşıyıcılar da (carriers) dünyanın hız ihtiyacını karşılamak için 2G ağlarını geliştirmeye devam ettiler ki bu da 2G ağlarına göre çok daha büyük bir bant genişliğine sahip, geçici bir hücresel ağ neslini doğurdu. Bu ağlar 2.5G ağlarıydı. Bir sonraki aşamada ise UMTS gibi 3G ağlar gezgin haberleşme açısından en parlak dönem olarak hayatımızda yer buldu (Büyükbaş, 2005) .

1G ve 2G teknolojilerinden farklı olarak 3G mobil tabanlı şebekeler kullanıcılara çoklu ortam hizmetleri sunulması için tasarlanmıştır ve sesin yanında görüntü, veri ve yüksek hızda internet bağlantısı sağlanabilmektedir. Bu nedenle performans değerlendirmesi açısından bütün QoS parametlerinin tek bir alandan elde edilerek karşılaştırılmasını sağlayabilecek en iyi sistem 3G-UMTS ağlarıdır.

UMTS (The Universal Mobile Telecommunications System) yani Evrensel Mobil Haberleşme Sistemi 3. kuşak (3G) mobil networklerden biridir. 3G sistemler ise 3GPP yani 3G Partnership Projects tarafından standartlaştırılmış ve temeli gerçek zamanlı ses akışı sırasında yüksek hızla veri iletimine dayanan yapılardır [2].

UMTS network altyapısı çeşitli kablosuz erişim teknolojileri üzerinden (GSM, Wideband Code Division Multiple Access (WCDMA) ve Code Division Multiple Access (CDMA2000)) ses (voice) ve veri (data) bütünleştirmesi yapan çeşitli uygulamalar için network servisi sağlayan yapıdır. Ayrıca UMTS dolaşım ve servis kapasiteleri ile kapsamlı bir mobil çoklu ortam servis setine dünya çapında erişim sağlar. UMTS abone başına 2Mbit/s hızlara kadar mobil veri servisleri ve yüksek konuşma kalitesi sunar. Genişletilmiş UMTS networklerinde ise özellik ekleme ve modifikasyon destekleride vardır çünkü burdaki amaç daha efektif paket iletimi sağlamaktır (Eckert vd., 2006).

UMTS ile birlikte önceki 2.5G (GPRS) ve 2.75G (Edge) ağların spesifikasyonlarında özel olarak yer almayan hizmet kalitesi (QoS) ve performans taahhüdü unsurları karasal geniş bant hizmet sağlayıcılarından hücre bazlı mobil telefon operatörlerine taşınmıştır. Her ne kadar 3G spesifikasyonlarını hazırlayan çalışma grubu 3GPP bu konuda kesin sınırlar belirlemiş olsa da, saha uygulamasında çevresel şartlar ve çekirdek ağın kurulumundaki tercihler sebebiyle beklenen performanslar elde edilemeyebiliyor.

Bu tez çalışmasında önerilen sistem kendi geliştirdiğimiz yazılım ve uygun donanımsal modifikasyonla mobil noktadan mobil noktaya (point to point) yada mobil noktadan internet bulutuna (point to net) UMTS ağı üzerinden erişimde QoS parametrelerin ölçülmesi ve kendi içinde değerlendirilmesi esasına dayanmaktadır. Buna göre ağı içinde kalmak ile operatör ağından internet bulutuna çıkmak arasındaki ölçüm farklarına dayanarak, operatörlerin internet çıkışını sağlayan gateway'lerinin QoS parametreleri değerlendirilmiş ve elde edilen deneysel sonuçlar yardımıyla farklı 3G hizmeti veren operatörlerin performansları ve hizmet kaliteleri karşılaştırılmıştır.

2. UMTS AĞININ ÖZELLİKLERİ

UMTS (Evrensel Gezgin Haberleşme Sistemi), IMT-2000'nin standartlarına uygun üçüncü nesil haberleşme sistemidir (Sehier ve Urie, 2001). Yüksek hızlı veri iletimine ve gerçek küresel gezinmeye olanak tanıyan bir şebeke sağlamayı hedeflemektedir. Yeni UMTS şebekesinin mevcut GSM işleticilerinin kullandıkları şebeke alt yapısı üzerine kurulması işleticiler tarafından tercih edilmektedir.

UMTS'in sağlayacağı özellikler;

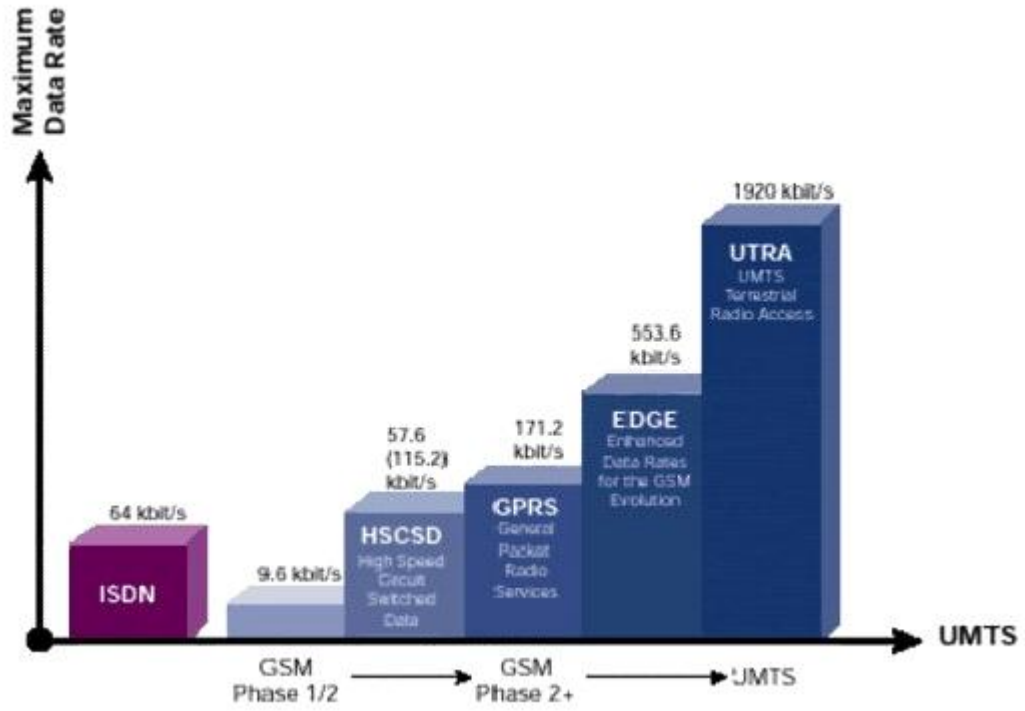
- Ses kalitesinde artış
- Bireye özel uygulamalar ve güvenlik artışı
- Kapsama alanının genişlemesi ve hücre sayısında azalma
- Sistem planlama, kurulum ve işletme masraflarında azalma
- Diğer elektronik cihazlarla olan elektromanyetik etkide azalma
- Aktarmadan kaynaklanan çağrı kayıplarında azalma
- Kablosuz güvenli veri transferi

UMTS bir çoğul işlev, çoğul hizmet, çoğul uygulama sayısal sistemi olup, akıllı şebeke (IN-Intelligent Network) işlevsel yapısına sahiptir. Dünya çapında bir gezginlik ve 2 Mbps hızlarına kadar genişbantlı çoğul ortamli hizmetler sunar (Holma ve Toskala, 2000).

UMTS (Rel'99); GPRS ve CAMEL ile genişletilmiş ve desteklenmiş GSM Phase 2+ ağ çekirdeklerini de kapsamaktadır.

GSM mobil istasyonları (Mobile Stations, MSs) şebekeye GSM hava arayüzü yoluyla bağlanacaklar, UMTS/GSM dual mod kullanıcı ekipmanı (UE) çok yüksek veri oranlarında (2Mbps), şebekeye UMTS hava arayüzü (UMTS air interfase) Uu' yu kullanarak bağlanabilirler [1].

Şekil 2.1'de çeşitli fazlardaki GSM sistemleri ve bunlara ait transmiyon hızları görülmektedir.



Şekil 2.1 GSM sistemlerine ait transmisyon hızları [1]

UMTS' in ilk versiyonunda tanımlanmış olan PLMN (Public Land Mobile Network) şu üç ana kategoriden oluşan şebeke bileşenlerini içermektedir:

- GSM Phase1/2 ağ çekirdek bileşenleri: MSC, VLR, HLR, AC ve EIR.
- GSM Phase 2+ genişletmeleri: GPRS (SGSN ve GGSN) ve CAMEL.
- UMTS' in spesifik genişletmeleri ve değişiklikleri, özellikle UTRAN.

Aşağıdaki çizelge 2.1'de 2. ve 3. nesil teknolojilerin bandgenişlikleri ve sahip oldukları özellikleri yer almaktadır.

Çizelge 2.1 2. ve 3. nesil teknolojilerin karşılaştırmalı çizelgesi [1]

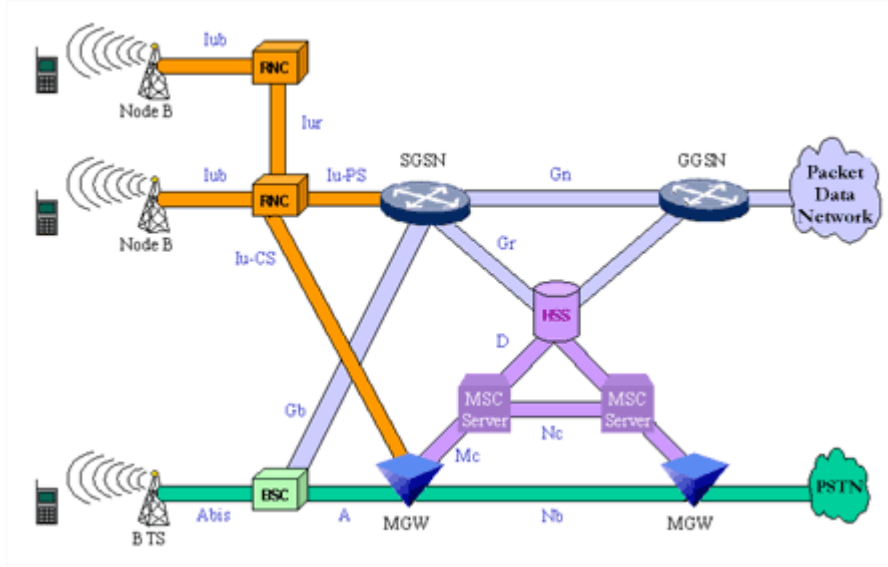
		Teknoloji	Bandgenişliği (Kbit/s)	Özellikler
1. Nesil Mobil	AMPS/ NMT	Gelişmiş Mobil Telefon Sistemi- (Advanced Mobile Phone System) Nordic Mobil Telefon- (Nordic Mobile Telephony)	9.6	- Analog ses hizmeti - Veri kapasitesi yok
2. / 2.5. Nesil Mobil	GSM	Küresel Mobil İletişim Sistemleri (Global System for Mobile Communication)	9.6 → 14.4	- Sayısal ses hizmeti - Gelişmiş mesaj gönderme hizmeti - Evransel dolaşım - Devre anahtarlama veri
	HSCSD	Yüksek Hızda Devre Anahtarlama Veri (High-Speed Circuit Switched Data)	9.6 → 57.6	- Gelişmiş GSM - Daha hızlı veri hızı
	GPRS	Genel Paket Telsiz Hizmeti (General Packet Radio Service)	9.6 → 115	- Gelişmiş GSM - Her zaman bağlantı imkanı - Paket anahtarlama veri
	EDGE	GSM Evrimi için Geliştirilmiş Veri Hızı (Enhanced Data Rate for GSM Evolution)	64 → 384	- Gelişmiş GSM - GPRS'den daha hızlı
3.Nesil Mobil	IMT-2000/ UMTS	Uluslararası Mobil İletişim 2000/ International Mobile/Evransel Mobil İletişim Sistemi (Telecommunications 2000 / Universal Mobile Telecommunications System)	64 → 2,048	- Her zaman bağlantı imkanı - Küresel dolaşım - IP-olanağı

Kaynak: Forrester Research

2.1 UMTS Şebeke Yapısı

UTRAN (UMTS Terrestrial Radio Access Network), şekil 2.2’de de görüldüğü gibi çekirdek şebekeye Iu arayüzü ile bağlı bulunan RNS (Radio Network Subsystem)’ lerden oluşmuştur.

RNS’ ler kendi aralarında doğrudan, Iur arayüzü ile bağlanabilirler. Herbir RNS, Radyo Şebeke Denetleyicisi (RNC-Radio Network Controller) ile bir ya da daha fazla sayıda B Nodu’ ndan oluşur (Richardson, 2005). İkinci unsur (B Nodu), kendilerine tahsis edilmiş hücreler içindeki radyo kaynaklarının kullanımını kontrol etmekle sorumludur. B Nodu, BTS olarak anılan radyo istasyonlarının bir veya daha fazlasını içerebilir. Bunlar, alan denetleyicisi ile kontrol edilirler. Radio Network Controller (Radyo Şebeke Denetleyicisi) farklı B Node’ ları arasındaki makro diversiteli tümleşik fonksiyonlar ile, yerel devir-teslim kontrolünü gerçekleştirmektedir (Holma ve Toskala, 2000).



Şekil 2.2 UMTS erişim şebeke mimarisi [1]

2.2 Baz İstasyon Alıcı-Vericisi (BTS-Base Transceiver Station)

Üçüncü kuşak sistemlerdeki BTS' ler, tümleşik hizmetlere erişebilmek için esnek bir mimariye sahip olmak zorundadırlar. Esneklik gereksinimlerinin karşılanabilmesi için, tasarımcılar, donanım mimarisini işleve bağlı olarak yapılandırdılar. Bu, birinci ve ikinci kuşak sistemlerde kullanılmış olan kanal tabanlı mimari yerine, W-CDMA' deki gibi tümleşik bir mimari kullanılmasını ifade etmektedir. Bu çözüm, geleceğin hizmetlerinin duyacağı talep çeşitliliği ile baş edebilmek için daha büyük bir esneklik kabiliyeti sağlar.

İletim ve alış işlemleri açısından pasif birimler olan antenler, doğru akım (DC) güç kaynaklarına ihtiyaç duymazlar. Herbir sektörde iki özdeş anten kullanılır. Bunlardan biri alış ve iletim için, diğeri ise yalnızca alış için kullanılır. Herbir sektör için iki anten kullanmak yerine, çift polariteli bir antende kullanılabilir. Böylece, sektör başına düşen anten sayısı azaltılabilir. Baz istasyon mimarisi, taşıyıcı kapasitesini farklı tip hizmetlerin farklı kullanıcılarına serbest bir biçimde tahsis edebilir. Söz konusu hizmetler şu şekilde sıralanabilir (Karim ve Sarraf, 2002):

- Ses hizmeti.
- 384 Kbps' de sınırsız sayısal bilgi taşıma hizmeti.
- 472 Kbps' lik ham veri iletim hizmeti.

Toplam kullanıcı veri bandgenişliğini sınırlandıran faktörler; mobil istasyonun veri kapasitesi, radyo hücre boyutu ve radyo arayüzünden kaynaklanan etkilerdir.

2.3 Radyo Şebeke Denetleyicisi (RNC-Radio Network Controller)

RNC, GSM sistemindeki BSC' ye karşılık düşmektedir ve genel bir ATM altyapısı üzerinde kurulur (Karim ve Sarraf, 2002). İşlemci ve diğer cihazların tümü ATM anahtarına bağlanır. Bu sayede istenildiğinde sistem kapasitesinin çok kolay biçimde arttırılabilmesi mümkündür.

RNC' nin işlevleri:

- Radyo kaynak ve makrodiversite tümleşik işlevleri (Yumuşak devir-teslim tümleşik fonksiyonu olarak anılmaktadır.).
- ITU-T tavsiyelerine uygun olarak konuşma kodlama, kod çözme.
- Iub iletim arayüzü tabanlı, 1,5 ve 2 Mbps ATM hatlarının ATM adaptasyon katmanı 2 (AAL2) hattı ile sonlandırılması.
- Iu arayüzü (155 Mbps' lik bir ATM hattı üzerinde taşınır.).
- Zamanlama ve senkronizasyon işlevleri.

RNC' nin mimarisi, kendisini gelişmiş W-CDMA radyo şebeke işlevleri için ideal kılan bir esnekliğe sahiptir. Örneğin, yeni ya da artan talepleri karşılamak için ilave olarak basılmış devre kartlarının, RNC katında bulunan alt rafların boş konumda bulunan yuvalarına yerleştirilmesiyle kapasite kolayca arttırılabilecektir. RNC, gerçek zamanlı olarak ölçüm verileri toplayabilmektedir.

2.4 UTRAN (UMTS Terrestrial Radio Access Network)

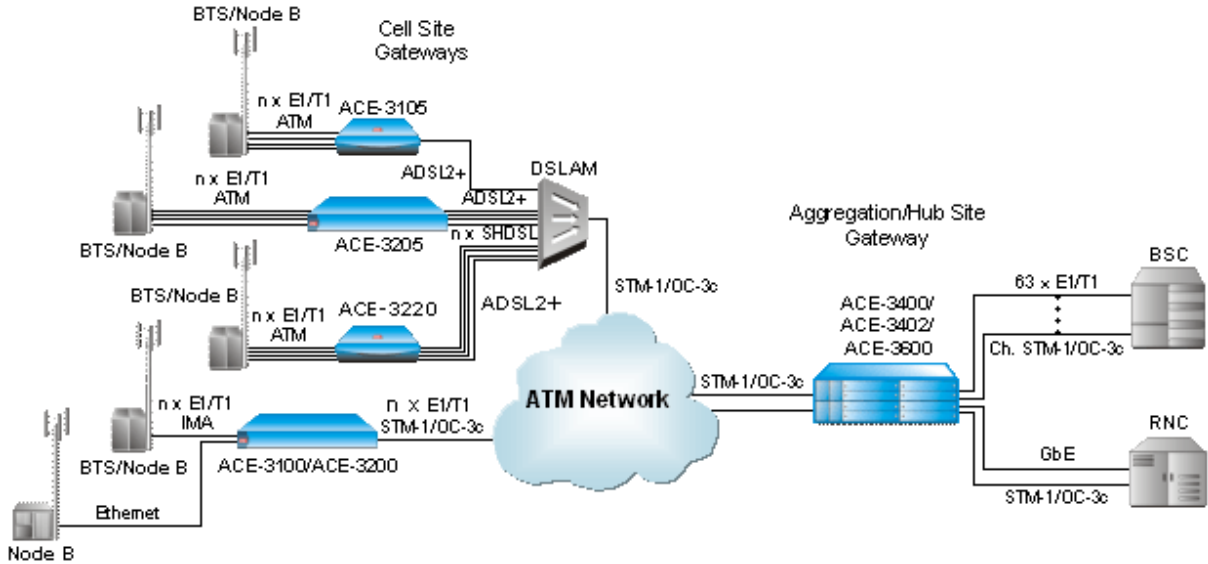
Mevcut şebekelerin bir genişletmesi olarak görülebilecek bir UMTS standardıdır. UTRAN ile birlikte iki yeni şebeke bileşeni getirilmiştir; RNC ve Node B. UTRAN, birbirinden bağımsız radyo şebeke sistemleri halinde alt bölümlere ayrılır. Herbir RNS (Radio Network System), bir RNC (Radio Network Controller) tarafından kontrol edilmektedir. RNC, herbiri bir ya da daha fazla sayıda hücre sunabilen bir Node B bileşenleri grubuyla bağlantılıdır (Karim ve Sarraf, 2002).

MSC, SGSN ve HLR gibi mevcut şebeke bileşenleri, UMTS' in ihtiyaç duyduğu gereksinimleri karşılamak üzere geliştirilebilir, fakat RNC, Node B ve taşınabilir el cihazları tamamen yeniden tasarlanmak zorundadır. RNC, BSC' nin yerini alacaktır ve Node B hemen hemen BTS' nin işlevselliğini yerine getirecektir. GSM ve GPRS şebekeleri genişletilerek ve yeni hizmetler; A, Gb ve Abis gibi mevcut arayüzler ile Iu, Node B ile RNC arasındaki Iub ve

iki RNC arasındaki UTRAN arayüzünü içeren bir şebekeye baştan başa entegre edileceklerdir. UMTS dört yeni açık arayüz tanımlamaktadır:

- Uu: UE ile Node B arasında (UTRA, UMTS W-CDMA arayüzü).
- Iu: RNC ile GSM Phase 2+ arasında şebeke çekirdek arayüzü (MSC/VLR veya SGSN).
 - ◆ Iu-CS: Devre bağlaşmalı veriler için.
 - ◆ Iu-PS: Paket bağlaşmalı veriler için.
- Iub: RNC ile NodeB arayüzü.
- Iur: RNC ile RNC arayüzü.

RNC, UTRAN' la birlikte otonom bir radyo kaynak yönetimi (Radio Resource Management-RRM) sunmaktadır. RNC, GSM-BSC' dekiyle aynı işlevi yerine getirir. RNC ile, RNS bileşenlerinin (RNC ve Node B' leri) merkezi kontrolü sağlanmaktadır. RNC; Iu, Iur ve Iub arabirimleri arasındaki protokol değişimini yürütmektedir. Ayrıca, OSS' ye erişimden ve tüm RNS' nin merkezi işletim ve bakımından sorumludur. Arayüzlerin ATM tabanlı olmasından dolayı RNC anahtarları, ATM hücreleriyle arayüzler arasında bulunmaktadırlar. Kullanıcının, Iu-CS ve Iu-PS arayüzlerinden gelen devre bağlaşmalı ve paket bağlaşmalı verileri; Iur, Iub ve Uu arayüzleri vasıtasıyla, UE' ye doğru ya da UE' den gelen çoğulortam iletişimini sağlamak için, bir arada çoğullanır. Aşağıdaki şekil 2.3'de UMTS şebekesindeki ATM altyapısı gösterilmiştir.



Şekil 2.3 UMTS Şebekesinde ATM ağ altyapısı [6]

RNC, GSM BSS' de eşdeğeri olmayan Iur arayüzünü kullanarak RRM' nin tamamının otonom olarak idaresini sağlar ve şebeke çekirdeğini bu ağır yükten kurtarır. UE' ye olan

RNC bağlantısı, tıkanıklık ve devir-teslim gibi hizmet kontrol fonksiyonları, tamamıyla tek bir RNC ile idare edilmektedir.

Eğer bir RNC, RNC' ler arası yumuşak devir-teslim yoluyla aktif bir bağlantıya katılmışsa bu RNC, DRNC (Drift RNC) olarak adlandırılır. DRNC yalnızca kod kaynaklarının tahsisinden sorumludur. SRNS (Serving Radio Network Subsystem) fonksiyonlarının eski DRNC' ye aktarılması mümkündür. CRNC (Controlling RNC) terimi, UTRAN erişim noktalarındaki lojik kaynakların kontrolünü yapan RNC' yi tanımlamak için kullanılır.

2.5 NODE B

Node B, hücrelerle radyo iletişimini temin eden fiziksel birimdir. Bölümlemeye bağlı olarak, Node B vasıtasıyla bir veya daha fazla sayıda hücre sunulabilir. Tek bir Node B, FDD ve TDD modlarının her ikisini de destekleyebilir ve işletme maliyetlerini düşürmek için bir GSM-BTS ile birlikte konumlandırılabilir. Node B; W-CDMA Uu radyo arabirimi ile UE' ye, ATM tabanlı arabirim olan Iub ile de RNC' ye bağlıdır. Node B, ATM' nin sonlandığı noktadır.

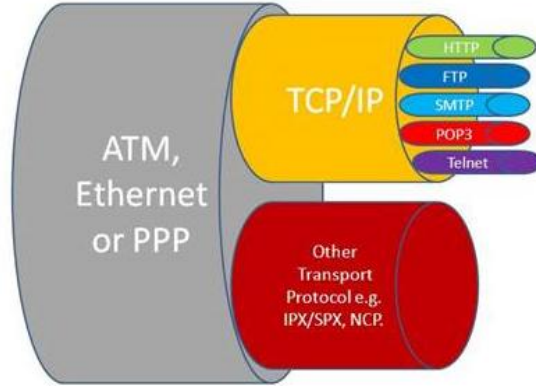
Node B' nin başlıca görevi, Uu radyo arayüzü kanalıyla gelen verileri dönüştürmektir. Node B; iletim hatası düzeltimi (FEC-Forward Error Correction), oran adaptasyonu (rate adaptation), W-CDMA saçılma/sıkıştırma işlemleri ile QPSK modülasyonunu da içine alır. Bağlantı kalitesini ve gücünü ölçer ve çerçeve hata oranını (FER-Frame Error Rate) hesaplar. RNC' ye iletilen verileri devir-teslim raporu halinde birleştirilmiş bir halde sunar. Node B, aynı zamanda FDD' nin daha yumuşak devir-teslim gerçekleştirmesinden de sorumludur. Böylece Iub arayüzü ilave iletişim kapasitesi gereksiniminden kurtarılmıştır.

Node B, kendi güç kullanım hattının (DL) güç kontrolüne de katılmaktadır. İletim hattı temelli dahili güç kontrol döngüleri yardımıyla iletişim güç kontrol (TPC-Transmission Power Control) komutlarını ayarlar. Dahili güç kontrolü için ön tanım değerleri, harici güç kontrol döngüsü yoluyla RNC' den sağlanır.

3. ÖNERİLEN ÖLÇÜM METEDOLOJİSİNİN DAYANAKLARI

3.1 UMTS Ağında Protokoller

UMTS ağına paket seviyesinde erişmek için operatörler şebeke tarafından desteklenen pahalı özel donanımlar kullanmaktadırlar. Bu cihazlar çekirdek ağ (core network) üzerinden geçen ATM paketleri seviyesinde yakalama ve izleme yapabilmektedirler [6]. Öte yandan daha önce de bahsedildiği gibi UMTS ağlarında ATM paketleri üzerinden taşınan veri iletişimi nerdeyse tamamen IP paketlerinden oluşmaktadır; dolayısıyla IP paketleri şekil 3.1’de de görüldüğü gibi ATM paketleri içine kapsüllenmiştir (encapsulation) (Khafizov ve Yavuz, 2000; Inamura et al, 2002).



Şekil 3.1 TCP/IP’nin ATM paketleri ile kapsüllenmesi

Her ne kadar çekirdek ağın yüksek hızda anahtarlama yapabilmesi ve ATM’nin daha güçlü QoS mekanizmaları olması 3G için en alt seviyede ATM’nin seçilmesine sebep olmuş olsa da, 3GPP’nin 3.5G ve 4G için planı bu ayrımı kaldırmak ve ağ bölgelerinin tümünde çekirdek ağdan, kullanıcı terminaline kadar ALL-IP olarak nitelendirdikleri tamamen IP protokolü ile iletişim kurmak şeklindedir (Khafizov ve Yavuz, 2000; Inamura et al, 2002). Özellikle cep telefonu başta olmak üzere UMTS ağını kullanacak kullanıcı ekipmanlarının (UE) sayısının çokluğu göz önünde bulundurulduğunda ALL-IP bir UMTS ağının bütün IPv4 adres uzayını tek başına dolduracağı aşıkardır, bu nedenle 3.5G ve 4G ağlarda hem gelişmiş QoS, hem de geniş adres uzayı sebebiyle IPv6 kullanılması öngörülmektedir. Bu sayede RNC seviyesinde talep edilen hizmet kalitesine göre işaretlenen (tagging) IP başlıkları çekirdek yönlendiricilerde (Core Routers) farklı önceliklerdeki kuyruklara alınarak işlenecektir.

Mevcut 3G ağlarında QoS sorunu ATM seviyesine bırakılmış olduğu için IP başlıklarında QoS işaretleme bulunmamaktadır ve kullanıcı terminalleri gündelik olarak kullandığımız TCP/IP yığınının çok benzer bir yığın kullanmaktadır. ATM paketlemesi baz istasyonu ve radyo ağ denetleyicisi (Node-B, RNC) seviyesinde tamamen kullanıcı ekipmanı (UE) cihaza şeffaf bir şekilde gerçekleştirilmektedir (Wee Lum ve OnChing, 2005).

3.2 Performans Ölçümü İçin Platform Tercihleri

3.2.1 Ölçüm Yazılımından Beklenenler

Bu çalışmada ölçüm yazılımından beklentimiz UMTS ağı üzerinde bir IP bağlantısı gerçekleştirmesi, farklı özelliklerde yankı (echo request), download ve upload taleplerinde bulunması, ve bu taleplerin sonuçlarını QoS parametreleri cinsinden kaydetmesinden oluşmaktadır. Bu talepler farklı hacimlerde, hızlarda ve zamana yayılmış olacağı için yazılımı uygun bir yazılım/donanım platformu için geliştirmek çok önemlidir.

3.2.2 Tercih Edilen Donanım Platformu

Operatörün kendi geliştirdiği özel ölçüm donanımlarına erişimi olmayan üçüncü taraflar için performans ölçümü amacıyla UMTS ağını girmek için en uygun başlangıç noktası kullanıcı ekipmanı (UE) olarak gözüktüğü de bu cihazların işlem gücü, zamansal çözünürlüğü, ve en önemlisi programlanabilirliği (dolayısıyla kullanılan protokole erişim) sınırlı olduğu için ölçümün temel platformu olmaya elverişli değildir.

Çalışmanın başlangıcında ölçüm yazılımını Symbian'ın Series 60 işletim sistemine sahip cep telefonlarında çalıştırmak üzere C ile kodlamak, yada Java dilinin cep telefonları için olan taşınabilir sürümü J2ME ile ölçüm yazılımını geliştirmek göz önünde bulundurulmuş, fakat böyle bir tercihin her donanımsal, hem de yazılımsal (UE üzerindeki işletim sistemi ve programlama arayüzleri) olarak bir darboğaz yaratacak olması sebebiyle vazgeçilmiştir.

Bu sebepten ötürü mobil ölçüm için bir taşınabilir bilgisayarın UMTS destekli bir UE (örneğin cep telefonu yada USB üzerinden bağlanan bir UMTS modem) üzerinden operatör ağına bağlanması ve UE'yi sadece bir modem olarak kullanması tercih edilmiştir.

3.2.3 Tercih Edilen Yazılım Platformu

Ölçüm aracı olarak kullanılacak yazılım olarak UMTS dışındaki normal kablolu ve kablosuz (WLAN) ağlarda hizmet kalitesini ölçen yazılımların kullanılması göz önünde bulundurulmuş ama bu yazılımların takip edilmesi istenen özel performans parametrelerini sunmaması sebebiyle, bu hedefe yönelik paket kurgulama, inceleme ve koklama* tekniklerine dayanan özel bir yazılımın geliştirilmesine karar verilmiştir.

TCP/IP yığını ve soket mimarisine verdiği içsel destek ve sağladığı hızlı geliştirme ortamı (RAD environment) sebebiyle günümüzde en çok tercih edilen programlama platformlarından biri olan .NET üzerinde çalışan nesneye yönelik C# programlama dili tercih edilmiştir. Visual Studio 2008 geliştirme ortamında yazılan C# kaynak kodu Microsoft C# derleyicisi olan csc.exe ile .Net 3.5 MSIL (MS Intermediate Language) ara diline derlenmiştir.

.Net platformu tıpkı Java programlama dili gibi kaynak kodunun öncelikle bir ara temsile derlenmesi ve daha sonra bu derlenmiş temsili üzerinde çalıştığı bilgisayarın özelliklerine göre x86 ASM makine diline çeviren bir JIT (Just-in-time) derleyiciden oluşmaktadır. .Net platformuna dahil programlama dillerinde yazılan kod Java'da olduğu gibi managed code olarak adlandırılır. Managed code'un özelliği yürütülen kodun doğrudan işlemci üzerinde çalışmadığı, bunun yerine bir sanal makine (virtual machine) uygulaması içinde çalışmasıdır. Örneğin Java dilindeki .java uzantılı kaynak kodu Java Byte Code denen bir ara dile derlendikten sonra .jar uzantılı bir Java arşivi oluşur ve bu arşiv Java Runtime Environment (JRE) isimli sanal makine tarafından çalıştırılır. .Net platformunda ise birden çok programlama dili vardır; bunların her biri kendine ait bir derleyici ile MSIL ortak diline derlendikten sonra CLR (Common Language Runtime) isimli sanal makine üzerinde çalışır.

Doğrudan işlemci üzerinde çalışan C ve C++ gibi dillerde yazılan kodlara native code denir. Managed code'un native code'a göre avantajları küçük bir performans maliyeti karşılığında sanal makinenin otomatik hafıza boşaltımı (automatic garbage collection), sıkı değişken türü zorlaması (strongly typed variables, type checking), hafıza sınırlarının sürekli takibi ile kodun istenmeyen hafıza konumlarına yazmasının engellenmesi (range, bound, overflow checking), uygulama alanları arasında izolasyon sağlama ve nesneye yönelik güvenlik erişim (Access Control List - ACL) kontrolüdür.

* Packet Forging, Packet Inspection, Packet Sniffing

Microsoft'un açık standartlara dayandırdığı .Net platformunun (derleyici, MSIL ve CLR) temel programlama dilleri C# ve Visual Basic .Net'tir. Öte yandan bu platform açık standartlara dayandığı için üçüncü taraflarca açık kaynak kodlu onlarca programlama dili için MSIL'e derleyen derleyiciler üretilmiştir (örneğin Python için IronPython, F#, Eiffel, Delphi.Net). Benzer şekilde platformun temelini oluşturan CLR için de açık kaynak kodlu alternatifler üretilmiştir; bunlardan en tanınanları MONO projesi ve GNU Portable.Net'tir ve komut (instruction) seviyesinde MSIL ile uyumludurlar.

Genel kanının aksine managed code'a derlenmiş bir program bir yorumlayıcı (interpreter) üzerinde çalışmaz. Common Language Runtime tıpkı Java çalışma zamanı ortamı JRE (Java Runtime Environment) gibi bir tam-zamanında (Just-in-time) derleyicidir; bu sayede MSIL kodu her makinada o makinanın ve işletim sisteminin özellikleri ile o anda makinanın durumuna özel olarak yeniden x86 ASM makine koduna derlenerek optimal makine kodu elde edilir.. JIT derlemeden sonra oluşan makina kodu işlemci açısından native koddan farksızdır. Buna ek olarak JIT derleyici programın çalışması sırasında kodun hangi kısımlarının sıklıkla çalıştığını belirleyerek hot spot olarak adlandırılan bu kısımlar için arka planda yeniden derleme yaparak özel optimizasyonlar (işlemci register'larının hot spotlara tahsisi, döngülerin açılması – loop unwinding) ile işlem zamanı (run-time) iyileştirmelerde de bulunur. Genellikle JIT derlemenin bu avantajları sebep olduğu performans kaybını fazlasıyla telafi etmekte ve oluşan kod çoğunlukla native kod olarak yazılıp derlenmiş programlar ile eşdeğer yada daha iyi performans oluşturmaktadır.

3.2.4 Donanım Platformunun Yapısı ve Yazılıma Sağlanan Arayüz

Ölçüm için kullanılan donanım platformunun yapısı daha önce de belirtildiği gibi UMTS destekli hücresel modem olarak davranan bir UE (Bluetooth üzerinden modem olarak tanıtılmış 3G destekli cep telefonu yada USB 3G modem) ile buna bağlı bir diz üstü bilgisayardan oluşmaktadır.

İşletim sisteminden bağımsız olarak UE bilgisayara modem olarak hizmet verdiğinde öncelikle UE ile bilgisayar arasında bir seri bağlantı kurulmaktadır. Her ne kadar geçmişte seri bağlantılar RS232 standardına göre 9 iğneli DE-9 kapılarına bağlı 9 bakır telden oluşan seri kablolar ile sağlanmakta ise de, günümüzde bu kapılar ortadan kalkmış ve RS232 standardı seri iletişim infrared (IrDA), yada bluetooth gibi başka erişim kanalları üzerinden sanal bir COM portu yaratılarak emüle edilmektedir. USB bağlantılı UMTS modem'lerde cihaz USB

bağlantısı üzerinden işletim sistemine kendisini sanal bir COM portu ve modem gibi tanıttığı için ayrıca bir iletişim bağlantısına gerek kalmamaktadır. USB bağlantısının* yarattığı sanal COM portu ilgili USB cihazın işletim sistemi sürücüsü tarafından yaratılarak işletim sistemi çekirdeğine eklenir. Her ne kadar fiziksel yapı bilgisayar, USB kablo ve USB cihaz (telefon yada modem) şeklinde olsa da mantıksal seviyede** işletim sistemi sanki gerçekten fiziksel bir RS232 bağlantısı varmış gibi davranır.

İşletim sistemi veri iletişimine ihtiyaç duyduğunda, sanal COM portu üzerinden UMTS modem'e erişmekte ve geçmişte POTS (Plain Old Telephony System) telefon hatları üzerinden kullanılan çevirmeli bağlantı modemlerinininkine benzer AT komutları vasıtasıyla modem'e emirler gönderir. AT komutları bilgisayar ile modem arasındaki komut iletişimini sağlamaktadır ve asıl taşınacak veri için bir protokol öngörmez.

İşletim sistemi üzerinde çalışan yazılım AT komutları ile modem'e bir bağlantı kurması emrini gönderdiğinde OSI katman yapısına uygun olarak aşağıdan yukarıya doğru ağ katmanlarının bağlantıları sırayla kurulur. COM portu emülasyonu ve AT komutları kullanılan tek eski teknolojiler (legacy) değildir. Modem'in bağlantıyı kurması ile OSI yapısının 1. katmanı olan fiziksel katman (physical layer) bağlantısının sağlanmasına müteakip 2. katman olan veri bağlantısı katmanında (data link layer) başka bir legacy protokol olan PPP (point to point protokol) kullanılır. Kablolu ethernet ağ yapısında MAC'e denk düşen PPP, 1994 yılında RFC 1661 ile tanımlanmış ve her tür protokolü kapsülleyebilen noktadan noktaya bir tünelleme protokolüdür. İronik bir biçimde günümüzün en güncel ATM ve Ethernet tabanlı ağlarına bağlanırken de bu eski protokolün türevleri olan PPPoA (point to point protocol over ATM) ve PPPoE (point to point protocol over Ethernet) kullanılmaktadır.

Bütün bu bahsedilen sanal COM portu, AT komutları ile bağlantı ve kontrol, bu COM portu üzerinden PPP kullanımı ve bu PPP üzerinden TCP/IP yığınının çalıştırılması işletim sistemi ve cihazın sürücüler tarafından sağlanır. Bu yapı kullanıcı aksini istemediği sürece tamamen transparandır ve kullanıcı yazılımı (benzer şekilde bu metinde önerilen yazılım) sadece işletim sisteminin TCP/IP yığına ait programlama arayüzü API ile muhataptır.

* USB kablosu ile bağlanmış cep telefonu, USB 3G Modem ve hatta USB üzerinden bağlanmıyor olsa da Bluetooth üzerinden cep telefonu modem'ine yapılan bağlantılarda oluşan sanal COM portları

** İşletim sistemi çekirdeğinde (kernel) tanımlı physical/logical device ayrımı açısından

Yazılımsal olarak, bu amaçla geliştirilmiş bir program işletim sistemini bypass ederek layer 2 seviyesinde UMTS modem'e giden PPP iletişimine doğrudan müdahil olabilir (Mun Choon ve Ramachandran, 2002). Bu, yaptığımız donanım tercihleri sınırlarında UMTS ağını inceleyebileceğimiz en alt seviyedir.* Öte yandan bu seviyede ölçüm yapmak için hem PPP protokolünü, hem de TCP/IP yığınının yazılımın içine sıfırdan kodlamak gerekecektir ki bu zahmetli uygulama işletim sistemine ait PPP ve TCP/IP sürücülerine nazaran ekstra ölçüm verisi sağlamayacaktır.

* Bundan daha aşağıda kalan ATM seviyesine erişebilmek için tamamen farklı özel donanımlara ihtiyaç vardır.

4. HİZMET KALİTESİ (QoS)

Sınırlı ağ kaynaklarının taleplere göre tahsisi özel mekanizmalar ile düzenlenmediği durumlarda hizmet kalitesinin (QoS) düşmesi kaçınılmazdır. Öte yandan hizmet kalitesi özellikle gerçek zamanlı ağ uygulamalarında çok önemlidir; çünkü hedefe ulaşan (yada ulaşmayan) veri ile ilgili bir sorunun giderilmesi için kaynaktan yeniden iletim talep etmek zaman kısıtı nedeniyle mümkün değildir. Şekil 4.1’de de görüldüğü gibi QoS için belirli öncelik seviyeleri mevcuttur.

Hizmet kalitesini belirleyen en önemli iki parametre gecikme (delay) ve bant genişliği’dir, bunlara ek olarak paket kayıp oranı (packet loss) ve jitter da iletişim kalitesini büyük ölçüde etkilemektedir.

Bazı ağ protokolleri kendi bünyelerinde hizmet kalitesi güvencesini temin etmek için hizmetlerin ayrımı ve ağ kaynaklarının tahsisi ile ilgili mekanizmalar barındırmaktadır; aşağıda bunlardan en bilinenler sıralanmıştır (Wee Lum ve OnChing, 2005).

- The Type of Service (TOS) field in the IP header (yerini Diffserv’e bırakmıştır)
- IP Differentiated services (DiffServ)
- IP Integrated services (IntServ)
- Resource reSerVation Protocol (RSVP) *
- Multiprotocol Label Switching (MPLS, 8 QoS sınıfı vardır)
- RSVP-TE
- Frame relay
- X.25
- Bazı ADSL modemler
- Asynchronous Transfer Mode (ATM) **
- IEEE 802.1p
- IEEE 802.1Q
- IEEE 802.11e
- HomePNA Home networking over coax and phone wires

* Sıklıkla RTP ile birlikte kullanılır.

** UMTS şebekelerinde Core Network ATM kullanır; ATM’nin telefon ve iletişim hizmeti sağlayan operatörlerce çok tercih edilmesinin birincil sebebi protokol içinde QoS mekanizmalarını barındırmasıdır. Tamamen IP tabanlı UMTS şebekelerinin (4G) kurulması, IP protokolüne de ATM gibi bütünsel QoS mekanizmalarının yerleştirilmesini beklemektedir. IPv6, sınırlı adres uzayı ve QoS gibi IPv4’ün cevap veremediği sorunları çözmek için tasarlanmıştır ve birçok paket anahtarlama ağıda ATM’nin yerini alması beklenmektedir.

QoS Priority Levels

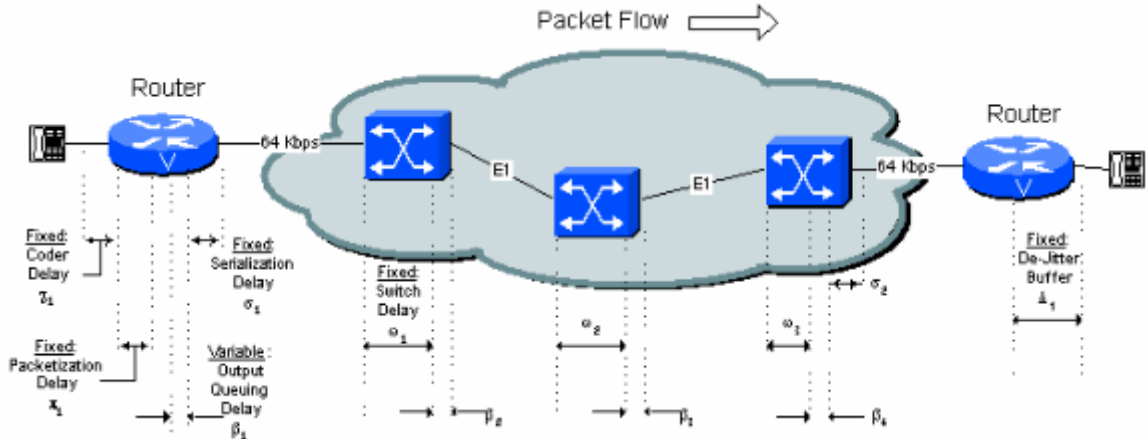
Priority Level	Traffic Type
0 (lowest)	Best Effort
1	Background
2	Standard (Spare)
3	Excellent Load (Business Critical)
4	Controlled Load (Streaming Multimedia)
5	Voice and Video (Interactive Media and Voice) [Less than 100ms latency and jitter]
6	Layer 3 Network Control Reserved Traffic [Less than 10ms latency and jitter]
7 (highest)	Layer 2 Network Control Reserved Traffic [Lowest latency and jitter]

Şekil 4.1 Hizmet kalitesi öncelik seviyeleri (Sari vd., 2008)

4.1 Gecikme Bütçesi (Delay Budget)

Bir iletişimin kullanılabilir olarak nitelendirilebilmesi için sinyalin kaynağında örneklenmesi ile hedefte yeniden oluşturulması arasında örnekleme, işleme ve iletim süreçleri toplamda belirli bir limitin altında zaman tutmalıdır (Zheng vd., 2003). Özellikle gerçek zamanlı uygulamalarda bu limit sabit ve kısadır; örneğin IP üzerinden ses taşıma (VoIP) için genel kabul görmüş toplam gecikme üst sınırı 200ms'dir, bunun üstündeki gecikmeler konuşmakta olan tarafların iletişimini güçleştirir. Bu bağlamda VoIP uygulamaları için fiziksel mesafeden bağımsız olarak ses verisinin örneklenmesi ile başlayan 200ms'lik bir gecikme bütçesinden bahsedilebilir; işleme (processing) ve iletim'in bütün kademeleri için bu bütçe uygun bir şekilde dağıtılarak bütçe aşılmaksızın sinyalin hedefte yeniden oluşturulması gerekmektedir; ses için >400ms gecikmeler kabul edilemez.*

* Ses verisi taşınırken gecikmenin 25ms'yi geçmesi durumunda iki tarafta da echo engelleyici filitreler kullanılması gerekmektedir, aksi takdirde hedefte gecikmeli olarak yeniden oluşturulan ses yeni bir kaynak olarak algılanır ve yükseltilerek asıl kaynağa geri iletilerek bir geri besleme döngüsü (feedback loop) oluşur.



Şekil 4.2 Ses İletiminde Gecikme Kaynakları (Zheng vd., 2003)

4.2 Gecikme (Delay) ve Geri Dönüş Süresi (Round Trip Time)

Gecikme, veri paketlerinin kaynaktan hedefe varana kadar kablo üzerinde (on the wire) geçirdikleri süredir. Daha önce bahsedildiği gibi gecikmeyi oluşturan birden çok bileşen vardır. Şekil 4.2’de de gösterildiği gibi bu bileşenlerin en önemlileri örnekleme gecikmesi, kodlama gecikmesi, paketleştirme gecikmesi, değişken çıkış kuyruğu gecikmesi, hatta çıkış gecikmesi, hat üzerindeki her bir anahtarlama ve yönlendirme cihazı için anahtarlama gecikmesi, varış kuyruğu gecikmesi ve jitter önleme gecikmesidir.

İki nokta arasındaki iletim için gecikme simetrik olmayabileceği* için bir verinin hedefe ulaştıktan sonra bir cevapla kaynağa geri dönmesi için geçen toplam süreye** de geri dönüş süresi (RTT, Round Trip Time) denir. Hizmet kalitesi (QoS) özellikle gerçek zamanlı uygulamalarda önem kazandığı için gecikme QoS’i etkileyen en önemli iki faktörden biridir.

4.3 Bant Genişliği (Bandwidth) ve Akış Hacmi (Throughput)

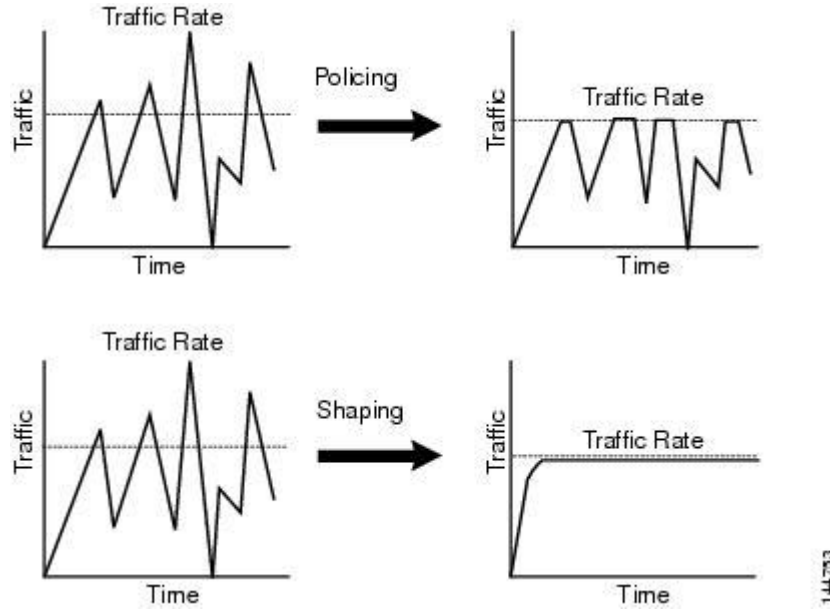
QoS’i etkileyen en önemli diğer faktör bant genişliği ve akış hacmidir. Toplam çıkış bant genişliğinin her hangi bir öncelik ayırt etmeden bütün trafik taleplerine eşit olarak dağıtıldığı QoS kullanılmayan senaryoda öncelikli trafik sıklıkla diğer talepler tarafından satüre edilmiş bağlantı’dan (link) ihtiyaç duyduğu bant genişliğini temin edemez. Hizmet kalitesini ister ATM, ister TCP seviyesinde aktif olarak kullanan bir ağ her akışı (flow) işaretler (tagging) ve her akışı bir QoS sınıfı ile ilişkilendirir. Her akışın kurulması (call setup) aşamasında ihtiyaç duyulan bant genişliği, taahhüt edilmesi istenen bant genişliği (CBR) ve gerektiğinde mevcutsa kullanılacak taahhüt edilmeyen ekstra bant genişliği talebi (ABR) olarak bildirilir. Kaynak ve hedef arasındaki hat üzerindeki bütün anahtarlama ve yönlendirme cihazları

* Özellikle downlink ve uplink hızlarının asimetrik olduğu ADSL ve benzeri bağlantı türlerinde

** Hedefte cevabı oluşturmak için yapılan işlemlerden doğan gecikme hariç.

ellerinde bu talebi karşılayacak kaynak var ise, talebi onaylar ve ilgili bant genişliğini diğer akışlardan ayırarak bu akış için rezerve eder. Hattaki bütün ağ cihazlarının taahhüt onayını müteakiben bu akış, belirtilen talebe özel olarak tanımlanan bir QoS sınıfı ile ilişkilendirilerek kurulmuş olur. Bu akış sonlanana kadar* hat üzerindeki cihazlar taahhüt ettikleri kaynaklar kullanılmıyor olsa bile başka akışlara kullandırmaz, öte yandan ekstra bant genişliği talebi (ABR) bütün akışlar için ortak bir havuzdan temin edilir ve zaten en başta taahhüt edilmemiştir. Bu iki QoS bant genişliği talep/taahhüt türüne ek olarak alt ve üst sınırı belirlenmiş değişken bant genişliği (VBR) de talep edilebilir. Çizelge 4.1 ve 4.2 de ATM hizmet kategorileri için karşılaştırmalı bilgiler özetlenmiştir.

QoS açısından bant genişliğini yönetmenin iki yolu vardır. Bunlardan birincisi policing olarak adlandırılmış olan bir akışın kendisine tahsis edilen bant genişliğini aşması halinde ilgili paketlerin hemen drop edilmesine dayanır; diğeri ise ağ cihazı üzerinde tahsis edilen bant genişliğinin üstünde kalan kısma ait paketlerin ayrı bir kuyruğa alınarak akış dalgalanmalarının tahsis edilmiş bant genişliğinin optimal kullanılmasını engellememesi olan akış şekillendirmedir (traffic shaping). Aşağıdaki şekil 4.3 policing ve traffic shaping'i özetlemektedir.



Şekil 4.3 Trafik şekillendirme (Sari vd., 2008)

* Call Teardown

Çizelge 4.1 ATM hizmet kategorilerinin tipik kullanım alanları

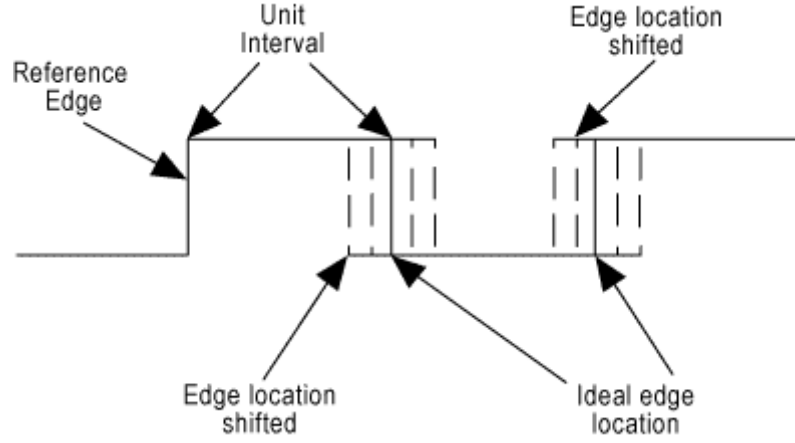
ATM Hizmet Kategorisi	Tipik Kullanım
Constant Bit Rate (CBR)	Gerçek zamanlı, QoS garantisi
Real-Time Variable Bit Rate (rt-VBR)	Gerçek zamanlı, istatistiksel mux
Non-Real-Time Variable Bit Rate (nrt-VBR)	Sadece istatistiksel mux
Available Bit Rate (ABR)	Resource exploitations, feedback control
Unspecified Bit Rate (UBR)	Best effort, garantisiz

Çizelge 4.2 ATM hizmet kategorilerinin karakteristikleri

Gecikme - Trafik veri miktarı	Kritik	Kritik olmayan
Öngörülebilir – sabit	CBR	-
Öngörülebilir - değişken	rtVBR	nrtVBR
Öngörülemez - ayarlanabilir	-	ABR
Öngörülemez - ayarlanamaz	-	UBR

4.4 Jitter

Jitter'ı gecikmede düzensizlik olarak tanımlayabiliriz. Elektronik iletişimde periyodik bir sinyalin zamansal değişkenliğine jitter denir ve genellikle bir zaman referansına göre ölçülür. Ağ iletişimde ise bir paketin ağ içindeki iki nokta arasında iletimi için gereken sürenin değişkenliğine jitter denir. Aslında standartlara göre kullanılması gereken doğru terim *paket gecikme değişkenliği*dir (Packet Delay Variation - PDV). PDV ağ performansının hizmet kalitesi (QoS) açısından ölçümünde önemli bir faktördür. Sabit gecikmeye sahip ideal bir ağda gecikme değişkenliği yani jitter yoktur. Jitter dağılımı rastlantısal yada öngörülebilir (deterministic) olabilir. Rastlantısal jitter durumunda paket gecikmesi değişkenliği genellikle gauss dağılımını izler.



Şekil 4.4 Jitter (Sari vd., 2008)

Ses yada görüntü aktarımı gibi gerçek zamanlı ağ uygulamalarında yüksek jitter hizmet kalitesini önemli ölçüde düşürür. Periyodik olarak örneklenmiş ses yada görüntü verisi paketlerinin sıralı ama farklı zaman aralıklarıyla hedefe ulaşması, verinin örnekleme zamanlamasına göre değil de ulaşma zamanına göre oynatılmasında görüntü yada sesin aslına sadık olmayan şekil 4.4'deki görüldüğü gibi değişken bir hızda yada kesikli olarak oluşmasına sebep olur.

Jitter'ı düzeltmek gecikme bütçesinden zaman harcayarak mümkündür. Özel bir de-jitter kuyruğuna alınan paketler içerdikleri zaman damgası (time stamp) bilgisine uygun olarak bekletilerek önceki ve sonraki pakete göre doğru zamansal konumda oynatılır; bu bekletme süresi gecikme bütçesinden harcanan zamandır.

4.5 Sırasız Teslim (Out of Order Delivery)

Paketlerin hedefe kaynaktan çıktıkları sıradan farklı bir sıra ile ulaşması durumunda, sırasız ulaşan paketlerin kendinden önce yola çıkmış paketler ulaşana kadar özel bir kuyrukta bekletilmesi gerekir. Her bir TCP paketi bir sıra numarası^{*} taşıdığı için bu numaralara bakarak sırasız gelen paketlerin kabul edilebilir bir zaman aralığında ulaşması halinde yeniden sıralanarak oynatılması mümkündür. Öte yandan daha önce ulaşmış olması gereken bir paketin ağda kaybolmamış olduğunun yada çok daha fazla gecikmeyeceğinin bir garantisi olmadığı için, gecikme bütçesinde paket yeniden sıralama için tahsis edilen zaman kadar beklendikten sonra halen ulaşmamış paket kaybedilmiş^{**} olarak kabul edilir ve daha sonra ulaşsa bile kuyruğa alınmaz.

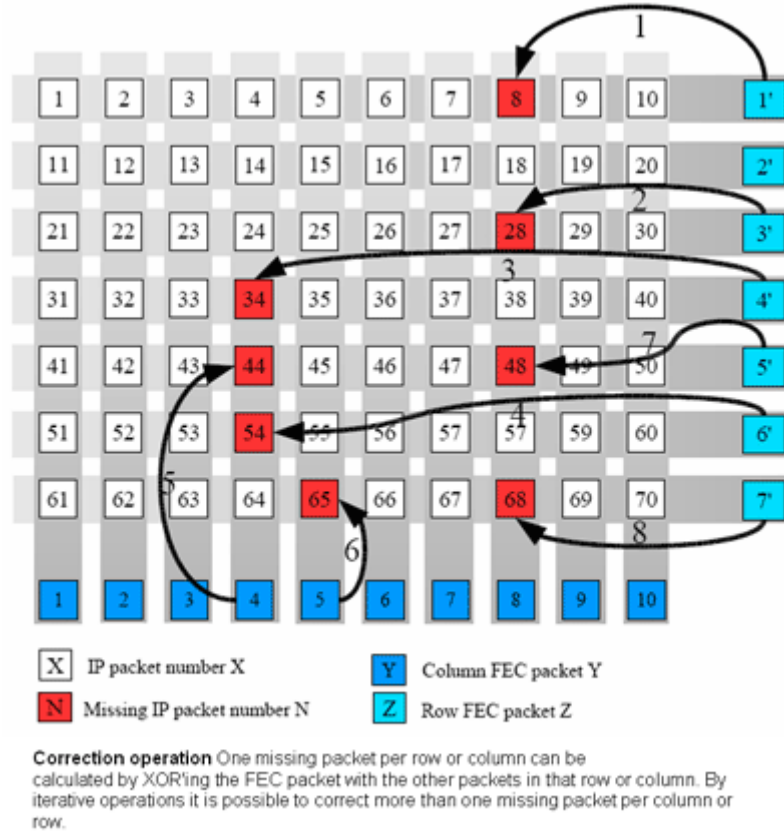
^{*} Sequence number

^{**} Packet Loss

4.6 Paket Kaybı (Packet Loss)

Bir veri paketinin kaynaktan hedefe ulaşamamasına paket kaybı denir. Gerçek zamanlı uygulamalarda kaybedilen paketleri yeniden talep edecek zaman yoktur bu nedenle paket kaybı hizmet kalitesini etkileyen önemli faktörlerdendir. Paket kaybı, bir paketin yolda tamamen dolu olduğu için bir kuyruğa girememesi, paketin üzerinde belirtilen yaşam süresi'ni (Time To Live – TTL) hedefe varamadan doldurması, yada iletim sırasında veri içeriği zarar gördüğü için doğrulama bilgisi olan checksum'da uyumsuzluk algılanması sebeplerinden biri nedeniyle ağ cihazları tarafından drop edilmesi ile gerçekleşir.

Paket kayıpları önlenemez olduğu için etkilerini azaltmak amacıyla ileri yönelik hata düzeltme (Forward Error Correction – FEC) algoritmaları kullanılarak daha fazla bant genişliği kullanımı karşılığında arada eksik kalan bir paket ulaşan önceki ve sonraki paketlerdeki ekstra bilgilere dayanarak bir seviyeye kadar yeniden oluşturulabilir.



Şekil 4.5 İleri Hata Düzeltme (Forward Error Correction – FEC) [3]

İleri hata düzeltme, veri paketlerinin çeşitli boyutlarda eksenlere dağıtılarak, bu eksenlerin kenarlarına o eksenin her bir dilimi için * bazı türev bilgilerin ** hesaplanması esasına dayanır. Şekil 4.5'deki örnekte 2 boyutlu bir matrise dağıtılmış 70 paketin her bir satır ve sütun için parity'leri hesaplanmıştır. Hatalı 8 paket ilgili satır yada sütundaki diğer paketler ve bu parity ile yeniden oluşturulabilir. Hatalı bir A paketinin bulunduğu satırda başka hatalı paket yoksa o satırdaki paketler ve o satırın parity'si ile A paketi sıfırdan üretilir, A paketinin bulunduğu satırda hatalı bir B paketi varsa bu kez sütun kullanılarak hata düzeltme denenir. A'nın bulunduğu sütunda da hatalı bir C paketi varsa, A paketi geçilir ve diğer hatalı paketlerin onarılmasına devam edilir. Paketleri düzeltilmesi B yada C'nin de tamir edilebilir hale gelmesine imkan tanıyabilir ve bu da A'nın düzeltilmesini mümkün kılar. Şekil 4.5 aynı satır ve sütunlarda birden çok hatalı paket olması halinde tüm verinin hangi sıra ile onarılabileceğini örnek olarak göstermektedir.

* Örneğin verilerin 2 boyutlu bir matrise yayılması durumunda bu iki eksenin kenarlarında oluşan her bir satır yada sütuna ait özet bilgi, yada 3 boyutlu bir matrise yayılması durumunda bu üç eksenin kenarlarında oluşan 3 adet 2 boyutu yüzeyde oluşan özet bilgi.

** En basit örnek olarak parity bilgisi yada bunun daha çok bite blok olarak uygulanması anlamında XOR

5. ÖNERİLEN ÖLÇÜM METODOLOJİSİ

Önerdiğimiz metodoloji, ölçüm aracı olarak .Net platformu üzerinde çalışan C# programlama dili ile geliştirdiğimiz bir yazılım aracılığıyla iki nokta arasında farklı karakteristiklerde ağ iletişimlerinin yaratılması ve bu iletişimlerdeki başarımların parametrelerinin kaydedilmesine ve bu parametrelerdeki değişimlerin hangi faktörlere bağlı olduğunu incelemeye dayanmaktadır.

5.1 Donanım Platformu Önerimizin Dayanakları

Programlanabilirlik açısından en geniş esnekliği sağladığı için mobil noktaların x86 tabanlı taşınabilir bilgisayarlar olması tercih edilmiştir. Bizim için üç sebepten ötürü bu tercih kaçınılmaz olmaktadır. Yaptığımız araştırmalar cep telefonu gibi kullanıcı ekipmanlarının (UE) Windows Mobile gibi modern bir işletim sistemi ile çalışıyor olsa bile Berkley socket arayüzünün ve TCP/IP yığınının tüm yordamlarını barındırmadığını göstermiştir; bu tür sınırlı taşınabilir cihazlar socket arayüzü ve protokol yığınlarının sadece sınırlı bir kısmını barındırmakta ve dolayısıyla bir çok ölçüm için sınırlı bir programlanabilirlik esnekliği sunmaktadır (Sari vd., 2008). İkinci olarak yine aynı sebepten ötürü bu cihazlar akmakta olan trafik ile ilgili ham veriye (paket başlıkları ile birlikte OSI standardının 2. yada en az 3. katmanı seviyesinde) erişime izin vermemekte, ve zamansal ölçümlerde yüksek performanslı zaman sayaçları bulunmamaktadır. Son olarak bu tür sınırlı kullanıcı ekipmanları gerçek zamanlı (real time) olarak aynı anda trafik üretme, gelen trafiği yakalama, paket seviyesinde işlem yapma ve sonuçları kaydetme için gereken işlem gücü ve kayıt ortamı hızına sahip değildirler.

x86 tabanlı bir bilgisayarda ise, işletim sistemi (Windows yada Linux türevi) mikrosaniye yada 100ns seviyesinde çözünürlüğe sahip yüksek performanslı zaman sayaçları, tam bir socket arayüzü ve IP yığını, ham socketler (raw socket) ile 3. ve çekirdek sürücüler (kernel driver) ile 2. OSI katmanı seviyesinde trafik yakalama, yüksek işlem gücü (>2Ghz) ve sabit diske yüksek hızda veri kaydı imkanlarını sunmaktadır.

Bu sınırlamalara rağmen Java 2 Mobile Edition (J2ME) platformu MIDP2 profilinde doğrudan TCP ve UDP paketlerinin taşınmasına izin vermektedir. Bu nedenle J2ME MIDP2 profilini hedef alan yazdığımız mobil Java uygulaması ile de ölçümler gerçekleştirerek sonuçlarımızı teyit ettik.

5.2 Geliştirilen Ölçüm Yazılımının Donanımla Etkileşimi

C# programlama dili, C ve C++ 'nın mirasını devam ettiren Java'ya çok benzeyen nesneye yönelimli, tür zorlayıcı ve yönetimli* bir programlama dilidir. Daha önce .Net platformunu anlatırken de belirttiğimiz gibi, Java'ya çok benzer bir kullanılmayan hafızanın geri toplanması mekanizması** içerir, bu özellik hem syntactic olarak kod uzunluğunu azaltarak daha okunur hale getirmekte, hem de hafıza kaçaklarının (memory leak) önüne geçmektedir.

En alt seviye donanımdan, bizim kodumuza kadar etkileşimde bulunan sistemleri seviyelerine göre sıralayacak olursak. Donanımın üzerinde çalışan işletim sistemi çekirdeğinde (kernel) makineye bağlı ağ kartı (NIC) için bir sürücü yazılımı çalıştırmakta ve bütün programların aynı NIC'i paylaşarak standart bir arayüzle kullanabilmesi için yazılım programlama arayüzleri sunmaktadır (API). Windows işletim sisteminde bu API Berkley Socket arayüzü ile uyumlu olarak geliştirilmiş WinSock API'sidir.

Yakın dönemde bilgisayar mühendisliğinde hem donanım, hem de yazılım geliştirme alanında giderek artan bir trend olan soyutlama ve sanallaştırma*** donanım ve kullanıcı yazılımı arasında arabuluculuk yapan işletim sisteminin sunduğu sürücü ve kullanıcı kütüphanelerine ek olarak artık üçüncü bir soyutlama katmanı olan çalışma zamanı ortamı (runtime environment) motorlarını endüstri standardı haline getirmiştir. Hem tür zorlamasız dinamik diller (örneğin PHP, Perl, JavaScript, Python), hem de sunucu tarafı, web yada masaüstü yazılımlarının geliştirildiği programlama dilleri (Java, C#, VB.Net, IronPython) kendilerine özel bir soyutlama katmanı olarak kendi çalışma zamanı ortam motorlarını zorunlu koşturmaktadır.

Geliştirdiğimiz ölçüm aracı yazılım C# programlama dili ile .Net platformuna ait arayüz kütüphanelerine erişmektedir. Bu kütüphaneler aynı Java'da olduğu gibi işlevlerine göre isim uzayları (namespace) altında gruplanmıştır, ve spesifik gruplar çoğunlukla dinamik olarak yükelenen bir kütüphane (DLL – dynamically loaded library) dosyasına denk gelmektedir. .Net platformuna ait bu DLL'ler, işletim sistemine ait aynı işlemi gerçekleştiren DLL'lerin platform içinden güvenle çağrılabilmesi için birer sarıcıdır (wrapper). İşlevin kendisi işletim sisteminin kütüphanesinde bulunmasına karşın, isim uzayına denk gelen .Net kütüphanesi parametre geçişlerinde tür kontrolü yapar, işletim sistemi fonksiyonunu çağırır, geri dönen

* Object Oriented, Strongly Typed ve Managed

** Garbage Collector

*** Abstraction ve Virtualization

sonucu alır ve eğer işlem başarısızlıkla sonuçlanmışsa yine .Net platformunun spesifikasyonlarına uygun bir şekilde ilgili Exception'ı oluşturur.

Bizim yazılımımız ağırlıklı olarak ağ erişimi için *System.Net* ve *System.Net.Sockets* çok iş parçacıklı çalışma için (multi threading) *System.Threading*, yüksek performanslı ölçümler için *System.Diagnostics*, işletim sisteminin sürücü seviyesinde tuttuğu ağ istatistiklerine erişmek için *System.Management*, isim uzaylarını kullanmaktadır.

5.3 Geliştirilen Yazılımın Yapısı

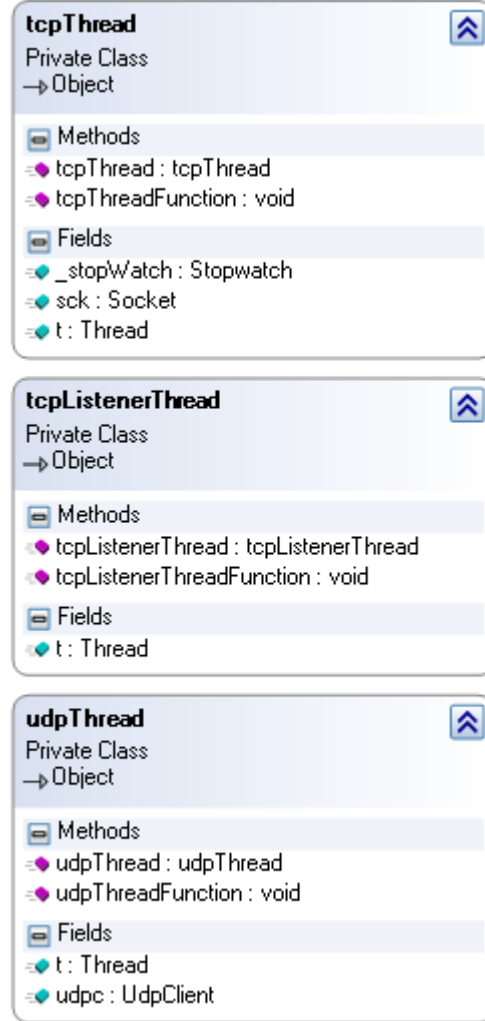
Her tür ağ iletişimi kalitesi ölçümü, en az iki nokta arasında bir veri akışı olduğu varsayımına dayanır ve o anda bu yoksa bile sentetik olarak bir akış yaratarak ağ ortamını (media), anahtarlama/yönlendirme ekipmanlarını ve protokol yığını farklı durumlara göre test eder. Aynı araç ile yapılan ölçümler farklı konumlar arasında, bağlantı türleri arasında veya bizim durumumuzdaki gibi farklı operatörler üstünden yapılarak bunların karşılaştırılmasıyla bir benchmark da elde edilebilir.

Yazılımımız noktadan noktaya sentetik bir veri trafiği yaratabilmek için iki ayrı programdan oluşmaktadır. Bunlardan birincisi yansıtma sunucusu olarak adlandırdığımız ve gelen trafiği aynı akış üzerinden göndericiye hiç değiştirmeden ileten programdır; diğeri ise bu sunucu ile bağlantı kuran ve bu bağlantı üzerinden farklı parametrelere göre veri akışı üreten ve aynı akışın kayıpsız olarak geri gelmesini bekleyen istemci programdır. İstemci program geri dönen akışın hizmet kalitesi (QoS) parametreleri açısından nasıl davrandığını gözlemleyerek, bu iki nokta arasındaki iletişimin çeşitli değişkenlere göre nasıl değiştiğini, yada farklı şartlar altında yapılan ölçümlerinin birbirlerine göre nasıl değişimler gösterdiğini hesaplayabilmemizi sağlar.

6. KAYNAK KODUNUN MİMARİSİ

6.1 Yansıtma Sunucusunun Kod Yapısı

Sunucu yazılımının tek görevi gelen veri akışını aynen geri yansıtmak olduğu için hiç kullanıcı etkileşimi bulunmamaktadır. Bu sebepten ötürü sunucunun metin tabanlı bir konsol uygulaması olması tercih edilmiştir.



Şekil 6.1 Sunucu yazılımının sınıf diyagramı

Sunucu yazılımının içindeki kodlar şekil 6.1’de de görüldüğü üzere *tcpThread*, *tcpListenerThread* ve *udpThread* olmak üzere üç ana sınıf içinde tasnif edilmiştir.,

Program ilk çalıştığında diğer dillerde olduğu gibi statik olarak tanımlı *main()* fonksiyonu başlar ve bir *tcpListenerThread* instance’ı ve bir *udpThread* instance’ı yaratır. Her ne kadar normal şartlarda programın burada sonlanması gerekiyormuş gibi gözükse de bu sınıflar bir çok diğer değişkenle beraber birer iş parçacığını (thread) sarmalayan (wrapper) sınıflar

olduklarından *main()* fonksiyonunun son satırı çalıştığında programın üç aktif iş parçacığı bulunmaktadır.

Her ne kadar kullanıcı arayüzü metin konsolu ile sınırlı olsa da buradaki kayıt (log) mesajlarının görüntülenmesinin ağ iletişimi işlemlerinden ayrı bir önceliğe sahip olmasını temin etmek için hiçbir ağ iletişim kodu programın ana iş parçacığında çalıştırılmamıştır. Bunun yerine her tür bekleme ve iletme işlemi kendine ait bir iş parçacığı yaratan bir nesnenin içinde kodlanmış ve iletişimin yaratılması yada yansıtılması gibi hayati işlemlere ait iş parçacıklarının önceliği işletim sisteminin ön tanımlı normal önceliğinden bir kademe yukarıda tutularak işletim sisteminin işlemci zamanlayıcısının (process scheduler) bu iş parçacıklarına öncelik vermesi sağlanmıştır. Bu çok iş parçacıklı yapı, çok çekirdekli (multi-core), fiziksel çok işlemcili (multi-processor) yada mantıksal çok işlemci (hyper-threading processor) sistemlerde her bir iş parçacığının (ekran güncelleme, yeni bağlantı bekleme, var olan bağlantının aktarımına devam etme) farklı bir çekirdek üzerinde çalışarak birbirlerini beklemek zorunda kalmamalarını temin etmektedir. Bu işlemlerin aynı iş parçacığını paylaşması tercih edilseydi, mikrosaniyeler seviyesinde ölçüm yapmak istendiği zaman ölçüm sonuçlarının ağ iletişim hattı dışında, ölçümü yapan bilgisayarın ve işletim sisteminin kendi gecikme ve dar boğazlarından da etkilenmesine kaçınılmaz olurdu.

6.1.1 TCP Dinleyici İş Parçacığı Sınıfı - *tcpListenerThread*

Sunucuya gelecek TCP akışlarını önceden belirlenmiş bir port üzerinden dinlemek için *tcpListenerThread* sınıfı constructor fonksiyonunda yeni bir iş parçacığı (thread) yaratır ve bünyesindeki *tcpListenerThreadFunction()* fonksiyonunu bu iş parçacığının giriş fonksiyonu olarak atar.

Bu fonksiyon kendi iş parçacığı içinde çalışmaya başladığında .Net kütüphanesinde tanımlı *TcpListener* sınıfından bir instance yaratır. Constructor fonksiyonuna parametre olarak önceden tanımlı 1337 sayısı gönderilerek bu port üzerinde bağlantı bekleyecek bir soket yaratılmış olur. Ardından sonsuz bir döngüye giren fonksiyon her adımda bir bağlantı oluşana kadar bloke olan *AcceptSocket()* metodunu çağırır ve oluşan soket nesnesini yeni bir *tcpThread* nesnesi oluşturmak için ilgili constructor'a parametre olarak sunar.

6.1.2 TCP İş Parçacığı Sınıfı - `tcpThread`

TCP bağlantılarını bekleyen iş parçacığı bir bağlantı talebi ilgili SYN paketi ile geldiği zaman, protokolün 3 yönlü el sıkışma (3-way handshake) standardına uygun olarak SYN+ACK paketi gönderir ve cevaben ACK paketi gelirse kurulan bağlantıyı bir socket nesnesi olarak yaratır ve `tcpThread` sınıfının constructor metoduna parametre olarak gönderir. Bu sınıf kendi içinde yaratıldığı sırada gelen socket nesnesi ile bir iş parçacığı nesnesi barındırır; constructor metodu gelen parametreyi sınıfın socket tipindeki alanına atar, yeni bir iş parçacığını `tcpThreadFunction()` metodu ile ilişkilendirerek yaratarak sınıfın iş parçacığı türündeki alanına atar ve iş parçacığını başlatır.

Her TCP bağlantısı için münferit olarak yaratılan iş parçacığı çalışmaya başladığında barındırdığı yüksek performanslı sayaç nesnesini sıfırlayarak saymaya başlatır. Bizim uygulamamızda aynı anda birden çok TCP akışı ile ölçüm yapmayı öngörmüyor olsak da bu çok iş parçacıklı (multithreaded) yaklaşım daha önce de belirtildiği gibi ölçümün kullanıcı ara yüzü ve bağlantı bekleyen kod kısmından bağımsız çalışmasını temin eder. Bağlantı devam ettiği süre boyunca TCP bağlantısına ait iş parçacığı bir döngü içinde socket kuyruğunda veri olup olmadığına bakar ve kuyrukta veri varsa tamamını `buffer` isimli byte dizisine çeker ve hemen ardından aynı veriyi aynı socket bağlantısı üzerinden geri gönderir. Bu veri yansıtma (echo) işlemi bağlantı kesilene kadar her seferinde ilgili zaman ve veri miktarı kaydedilerek devam eder. Bağlantının kesilmesi durumunda iş parçacığı fonksiyonu sonlanır ve ilgili bütün nesneler referans sayaçları sıfıra ulaştığı için otomatik olarak .Net çalışma zamanı motoru tarafından temizlenerek boş belleğe iade edilir.

6.1.3 UDP İş Parçacığı Sınıfı - `udpThread`

UDP bağlantıları için yansıtma işlemi yapan kod mantık olarak TCP için kullandığımız yaklaşıma çok benzemektedir. TCP bağlantısı için kullandığımız yöntemden farklı olarak UDP bağlantılarında dinleme ve akışlar için ayrı nesneler kullanmaz. Bunun sebebi UDP'nin TCP gibi bağlantıya yönelimli (connection oriented) bir protokol olmamasıdır. Bütün UDP akışlarını karşılamak ve yansıtmak için sadece bir adet `udpThread` nesnesi `main()` fonksiyonunda yaratılır ve çalıştırılır.

Constructor metodunda yeni bir iş parçacığı (thread) ve yeni bir socket yaratan `udpThread` sınıfı bu iş parçacığında bir sonsuz döngüye girerek socketin bağlı olduğu porta gelen bütün UDP datagramlarını geldiği IP adresine geri gönderir. UDP akışlarında bağlantıdan söz

edilemeyeceği için farklı IP'lerden bu porta gelen bütün datagramlar aynı soket tarafından yakalanır ve iş parçacığında yansıtılır.

6.2 Ölçüm İstemcisinin Kod Yapısı

Ölçüm istemcisi programı yansıtma sunucusunun aksine grafik tabanlı bir kullanıcı arayüzüne sahiptir.

6.2.1 ICMP Echo Taleplerinin Üretilmesi ve Gözlemlenmesi

Ölçüm istemcisinin şekil 6.2'de görülen ICMP paketleri ile ölçüm yapma tekniği bir çok ağ yolu izleme programı (route tracer, traceroute - tracert) ile benzerlik göstermektedir. Trace route yazılımları IP üzerinde çalışan ICMP (Internet Control Message Protocol) protokolü ile yapılan ve ping olarak adlandırılan yansıma talepleri (Echo Request) ile çalışır. Bu yansıma taleplerinin karşılanması ve cevaplanması doğrudan TCP/IP yığınının (stack) tanımına dahil olduğu için TCP/IP çalıştıran her tür donanımdan (PC, Router, ADSL Modem, Ağ Depolama Birimi, Ağ Arayüzü Yazıcı, System on a Chip - SoC, vb..) ICMP Echo taleplerine RFC 792'ye uygun olarak cevap vermesi beklenir.

ICMP packet				
	Bit 0 - 7	Bit 8 - 15	Bit 16 - 23	Bit 24 - 31
IP Header (160 bits OR 20 Bytes)	Version/IHL	Type of service	Length	
	Identification		flags and offset	
	Time To Live (TTL)	Protocol	Checksum	
	Source IP address			
	Destination IP address			
	ICMP Payload (64+ bits OR 8+ Bytes)	Type of message	Code	Checksum
Quench				
Data (optional)				

Şekil 6.2 IP ve içindeki ICMP paketinin alanları (Mun ve Ramachandran, 2002)

- **RFC 792 (ICMP), sayfa 15:**

Yankı talebi (echo request) mesajında alınan veri yankı cevabı (echo reply) mesajında geri döndürülmelidir. Tanımlayıcı (identifier) ve sıra numarası (sequence number) yankı göndericisi tarafından cevapları, yankı talepleri ile eşleştirmekte kullanılabilir. Yankı üreten düğüm cevabında aynı değerleri döndürür.

- **RFC 1122 (Requirements for Internet Hosts - Communication Layers), sayfa 42:**

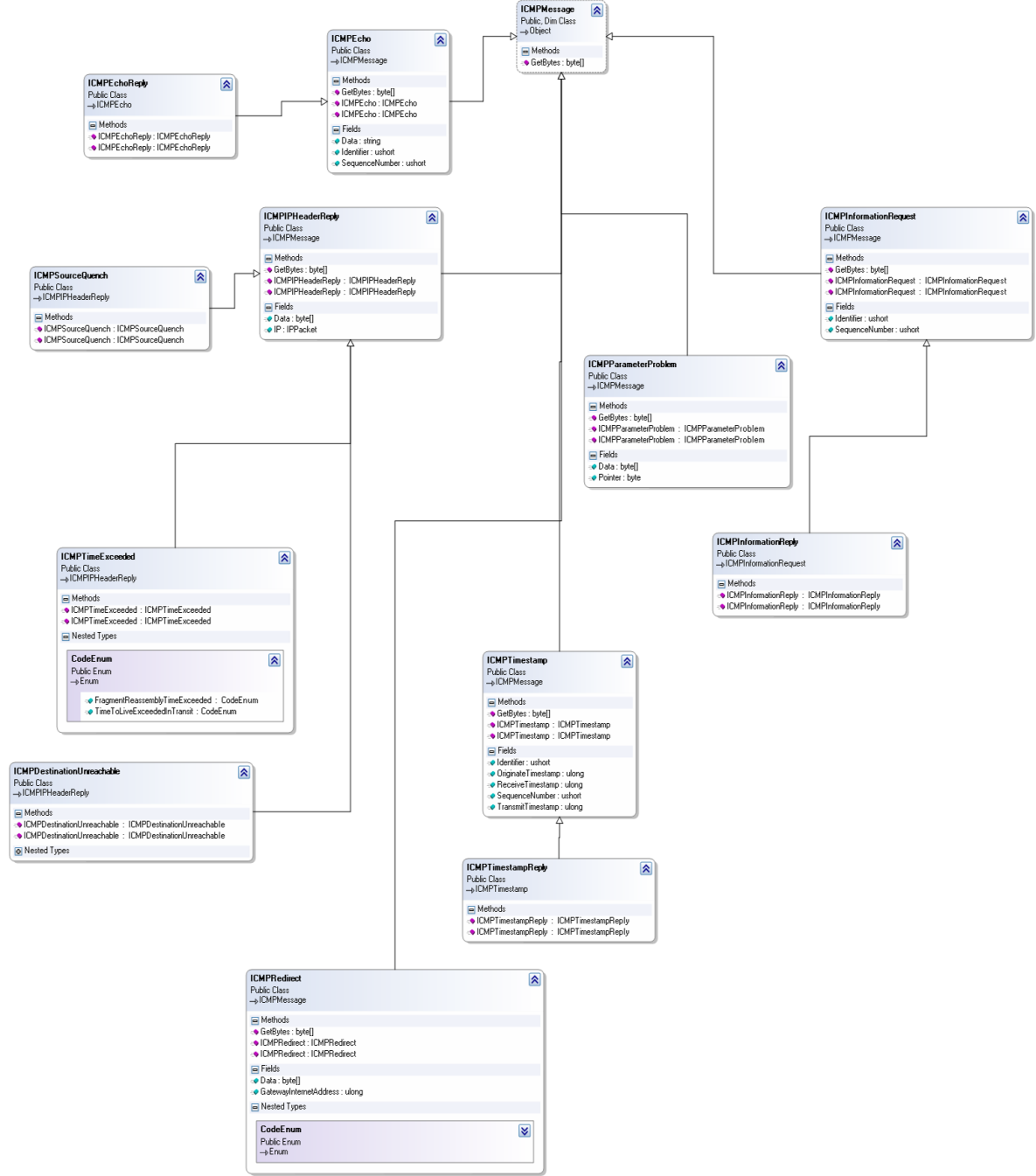
Her host ICMP Echo sunucu fonksiyonunu Echo Request'leri alacak ve ilgili Echo Reply'ları gönderecek şekilde kurmak zorundadır. Bir host tanı aracı olarak uygulama-katmanı seviyesinde Echo Request'ler gönderecek ve ilgili Echo Reply'ı alacak bir arayüz kurmak zorundadır. IP broadcast yada IP multicast adreslerine hitaben gönderilmiş bir ICMP Echo Request işlem yapmaksızın kuyruktan atılabilir.

- **RFC 1812 (Requirements for IP Version 4 Routers), sayfa 58 ve 59:**

Bir yönlendirici ICMP Echo sunucu fonksiyonunu Echo Request'leri alacak ve ilgili Echo Reply'ları gönderecek şekilde kurmak zorundadır. Yönlendiriciler bir ICMP Echo Request datagram'ı almaya, yeniden birleştirmeye (reassemble) ve geri yankılamaya kendine bağlı ağların MTU* boyutuna kadar destek vermeli. Echo sunucu fonksiyonu IP broadcast ve IP multicast adreslerine yönelik ICMP echo request'lerine cevap vermemeyi tercih edebilir. Bir yönlendirici, seçildiğinde her tür ICMP Echo Request'i göz ardı etmesini sağlayacak bir konfigürasyon seçeneği barındırmalıdır. Böyle bir seçenek varsa bunun öntanımlı (default) değeri ICMP cevaplarını vermek yönünde olmalıdır. Bir ICMP Echo cevabındaki IP kaynak adresi ilgili ICMP Echo Request mesajındaki hedef adresi ile aynı olmak zorundadır. Bir ICMP Echo Request mesajındaki verinin tamamı ilgili Echo Reply mesajında dahil edilmelidir.

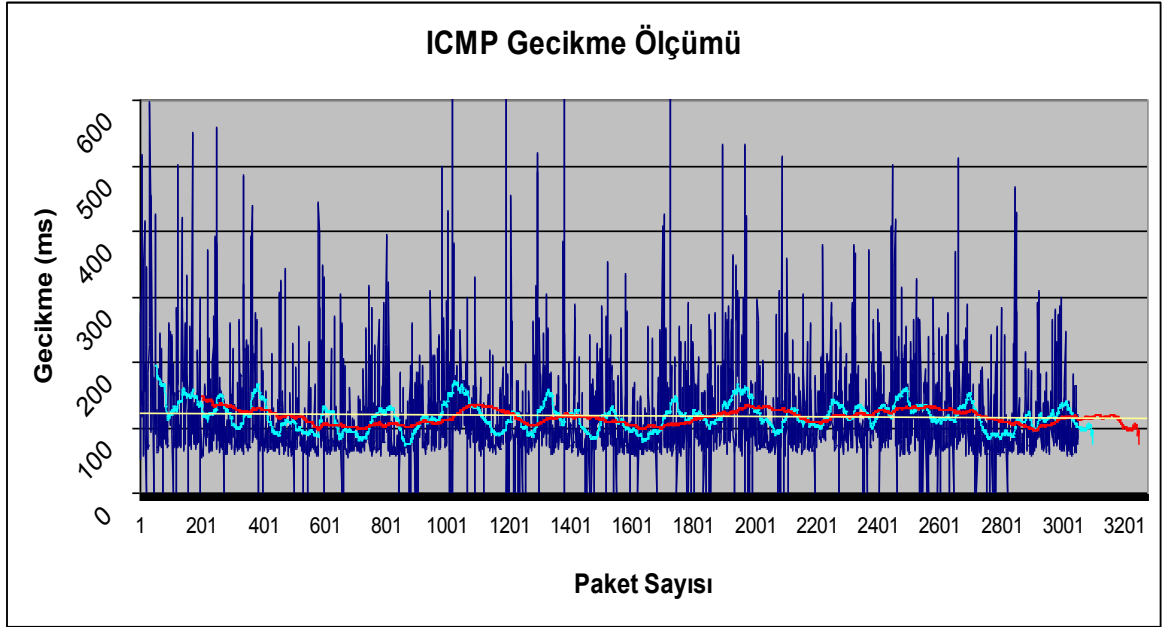
* Maximum Transmition Unit : Bir ağ arabiriminin byte olarak bir paket içinde gönderebileceği maksimum yük boyutu.

Şekil 6.3’de ICMP protokolünü kullanmak için oluşturulan kodlara ait class diagram görülmektedir.



Şekil 6.3 ICMP paketlerinin sınıf diyagramı

Şekil 6.4’de de görüldüğü gibi öncelikle hedef makineye 32 byte’tan başlayan ve sırayla 64KB’ye kadar artan büyüklüklerde her bir büyüklük için birden çok ICMP echo talebi gönderilir; ölçümün gecikme ve paket kaybı parametreleri kaydedilir. Ardından TTL 1’den (Time To Live) başlayarak maksimum TTL 30’a kadar artan hop sayısı için aynı ICMP echo talepleri gönderilerek hat boyunca aradaki bilgisayarlar ve ağ cihazları belirlenir ve aynı ölçüm parametreleri kaydedilir. TTL’nin her bir artışında toplam delay’in ne kadar arttığına bakarak hop’lar arasındaki gecikme ile ilgili bir tahmin hesaplanır. Unutulmaması gereken echo talebinin süre olarak delay yerine RTT’yi (Round Trip Time) vermekte olmasıdır, bu nedenle bütün gecikme değerleri ikiye bölünerek hat üzerinde iletişim hızının yöne göre simetrik olduğu varsayımında bulunulur.



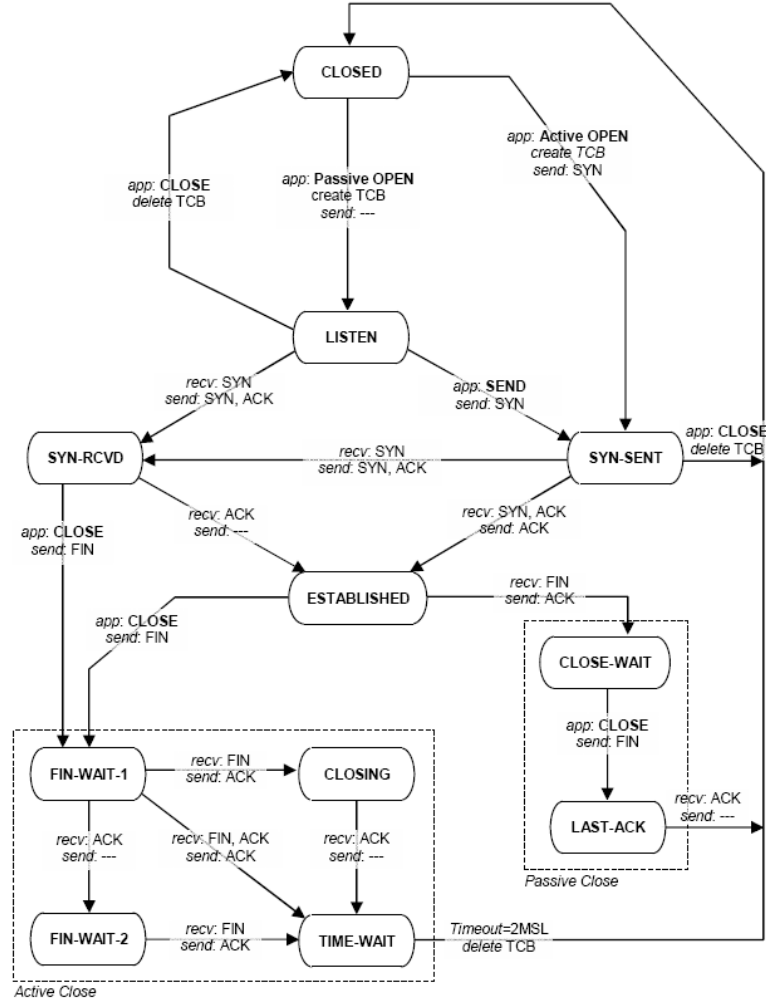
Şekil 6.4 ICMP Echo talepleriyle A operatörü için gecikme (delay) ölçümü

Yaptığımız diğer ölçüm türleri (TCP, UDP) ile birlikte karşılaştırıldığında bant genişliği dışında diğer bütün QoS parametreleri için ICMP’nin en hassas ölçümleri sunduğunu gördük.

6.2.2 TCP Akışı Yaratımı

ICMP paketleri ile ölçüm yapmaktaki temel yetersizlik, ICMP paketlerinin IP katmanının üstünde çalışan ama TCP ve UDP’den farklı bir protokol olmasıdır. Bu nedenle TCP iletişimine özgü sayesinde ölçebileceğimiz bant genişliği ve sırasız iletim (out of order) gibi QoS parametrelerine ICMP ile ulaşamayız. Bu sebepten ötürü önerdiğimiz yazılım bir ölçüm

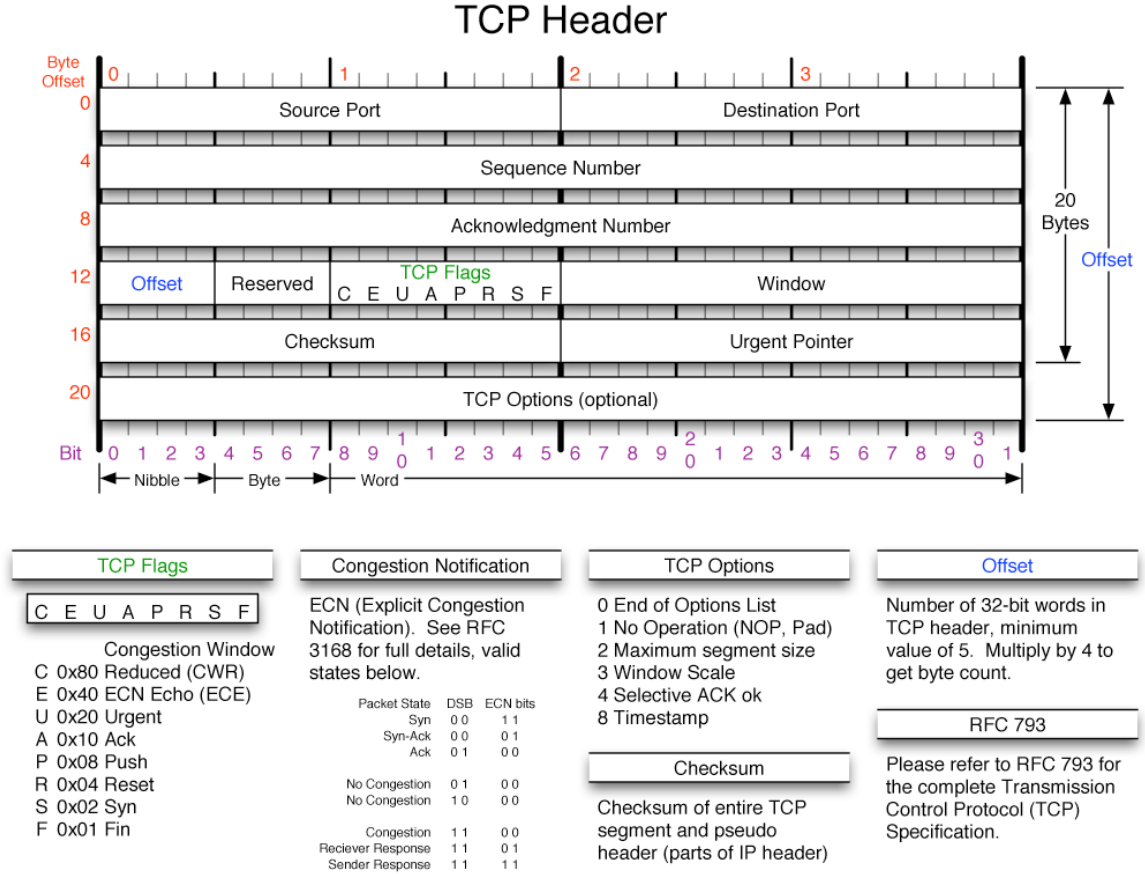
metodu olarak ICMP yansımalarına benzer bir şekil 6.5'deki gibi TCP akış yansımaları yazılımsal olarak sağlamaktadır.



Şekil 6.5 TCP State Diagram (Zagnal ve Khan 2005)

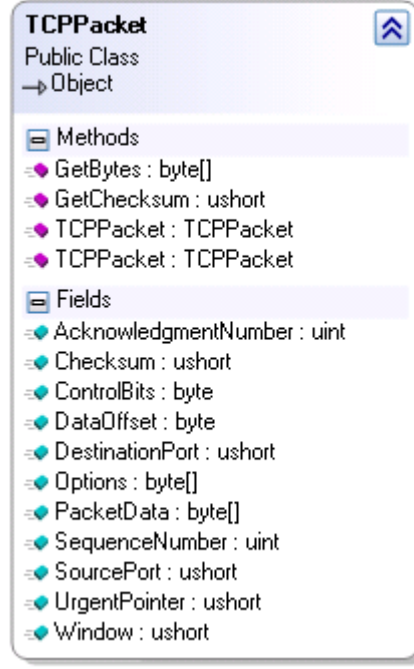
TCP akışlarıyla ICMP'de olduğu gibi farklı TTL'ler için ölçüm yapılamadığından, yazılımın TCP ile yaptığı ölçümler bütün hat için tek bir değer olur ve hop'lar arasındaki QoS parametresi farklılıklarına ulaşamaz.

Şekil 6.6'da bir TCP header yapısı görülmektedir.



Şekil 6.6 TCP Paket Başlığı [5]

Yukarıda da belirtildiği gibi TCP protokolünü UDP ve ICMP'den ayıran en temel özellik bağlantı yönelimli (connection oriented) bir protokol olmasıdır. Bu sebepten ötürü TCP paketleri ile yapılan ölçümler bize UMTS ağlarında paket kayıplarının telafisinin hizmet kalitesi parametreleri açısından maliyetini, ve yine bu hizmet kalitesi parametrelerinin birbirlerine göre değişimini ölçme imkanı tanır. Örneğin paket kayıpları, gecikme ve jitter (hatta sırasız iletim) gibi QoS parametreleri bant genişliğinden büyük ölçüde etkilenmektedir ve efektif veri akış hacmi bant genişliğini satüre etmeye yaklaştıkça bu diğer parametrelerde hızlı bir şekilde gerileme kaydedilir.



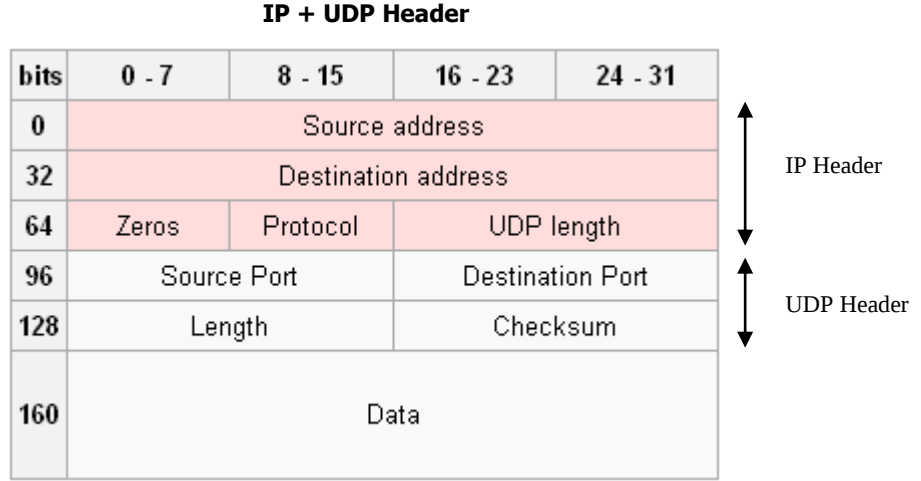
Şekil 6.7 Kaynak kodunda TCP paketine ait sınıf diyagramı

Yazılımımız ilgili RFC’de belirtilen paket yapısı ile örtüşen bir class tanımına dayanmaktadır. Her TCP paket başlığı şekil 6.7’de görülen *TCPPacket* sınıfından yaratılmış bir nesne ile temsil edilmekte ve yansıyan akışın hacmini kullanarak akışın bant genişliği hesaplanmaktadır. Bu sistemle iletilen paketlere ait ACK ve zaman aşımına uğrayan paketlere ait yeniden iletim talebi paketleri gözlenemediği için kullanılan gerçek bant genişliğinin ölçülenden biraz daha yüksek olması beklenmelidir.

6.2.3 UDP Akışı Yaratımı

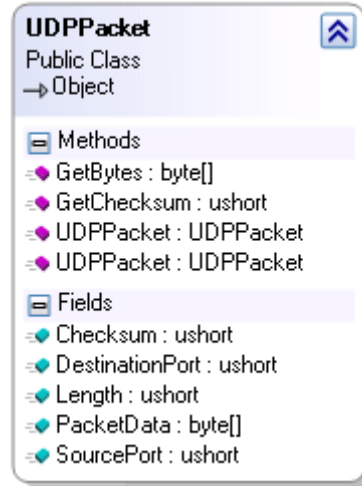
UDP paketleri aracılığıyla yaptığımız ölçümler nitelik olarak ICMP ve TCP’nin arasında kalmaktadır. UDP bir yandan ICMP’de olduğu gibi bağlantıya yönelik olmayan tek soket üzerinden birçok iletişimin paket bazında gerçekleştirilebildiği bir protokol olmasına karşın, bir yandan da tasarımındaki asıl amaç IP katmanını üzerinde düşük gecikme ve yüksek hızla veri taşımaktır.

Şekil 6.8’de bir IP + UDP header yapısı görülmektedir.



Şekil 6.8 IP ile beraber UDP paket başlığı [5]

Nerdeyse tüm yönlendiriciler TCP ve UDP akışlarını farklı türlerdeki kuyruklarda biriktirir ve UDP paketlerine öncelik verir; bu doğasından ötürü UDP daha sade bir paket başlığı formatına sahiptir. Bu protokolün tanımladığı işlevlerin de az olmasından kaynaklanmaktadır; nitekim UDP akış kontrolü, hata kontrolü, eksik paketleri yeniden isteme, başarılı iletimi onaylama ve paketleri yeniden sıralama özelliklerine sahip değildir. RTP, DNS, streaming media, VoIP, IPTV, TFTP ve birçok online oyun bahsedilen özellikleri nedeniyle UDP üzerine kuruludur.



Şekil 6.9 UDP paketi sınıf diyagramı

UDP paketleri ile yapılan ölçümler TCP ile yapılanlar ile aynı esaslara dayanır. Her bir UDP paketi şekil 6.9’da görülen *UDPPacket* sınıfına ait bir nesne ile yaratılır ve yansıtma sunucusuna gönderilir. TCP’den farklı olarak kayıp paketler işletim sistemindeki protokol

yığını tarafından yeniden talep edilmediği için kayıp paketler ve gerçek bant genişliği kullanımı ölçülebilir. Aynı zamanda çoğu yönlendirici UDP paketlerini TCP paketlerine nazaran daha öncelikli olarak işlediğinden UDP paketleri ile ölçülen gecikme de daha belirleyicidir; yine de gecikme ölçümleri için ICMP verileri esas alınmalıdır.

6.2.4 TCP, UDP ve ICMP ile Yapılabilen Ölçüm Türlerinin Karşılaştırması

Aşağıdaki çizelgede 6.1’de de görüldüğü gibi IP üzerinde çalışan farklı protokoller farklı QoS parametrelerini ölçmek için daha uygundur. TCP protokolü akış olarak kullanıldığında kayıp paketlerin yeniden talep edilmesi, sırasız iletimin kuyruklarda yeniden sıralanması özelliklerinden dolayı sadece bantgenişliği ölçümü için uygundur (ki bu durumda bile TCP paket başlıklarının oluşturduğu trafik ayrıca eklenmelidir). ICMP protokolü ise 1 byte ile 64 kbyte arasında paketler kullandığı ve bir akış oluşturmadığı için bant genişliği ölçümlerinde kullanmak için verimli değildir.

Çizelge 6.1 Protokollerin QoS parametrelerine göre ölçüm yeteneği

Protokol	Bantgenişliği	Gecikme	Jitter	Paket Kaybı
TCP	✓	✗	✗	✗
UDP	✓	✓	✓	✓
ICMP	✗	✓	✓	✓

6.2.5 Paket Koklama (Packet Sniffer) ile Ekstra Veri Elde Edilmesi

Bahsi geçen ilk üç ölçüm yönteminde protokolleri işletim sisteminin programlama arayüzleri üzerinden kullanmakta olduğumuz için iletişimle ilgili olarak TCP/IP yığınının bize izin verdiği kadar bilgiye erişmemiz mümkün olmaktadır. Öte yandan paket ağlarının işleyiş durumu ve mekanizmaları ile ilgili en belirleyici bilgiler bu yığının erişmemize izin vermediği paket başlıkları kısmında yer almaktadır.

Paket koklama* OSI katmanının daha alt seviyelerinde veri yakalamamızı temin eden bir tekniktir ve her alt katmanda veri yakalama şansı kazanmamız bir üst katmana ait başlık bilgilerini ham olarak elde edebiliyor olmamız anlamına gelmektedir; kuşkusuz bu da protokol yığınının o anki durumu (hangi TCP state’te olduğumuz, üç yönlü el sıkışmanın nasıl gerçekleştiği, vb..) hakkında bilgi vermektedir.

* Packet Sniffing

Bilindiği üzere OSI katmanlarından ikincisinde yer alan MAC ve Ethernet standartları broadcast mantığıyla çalışmaktadır; bütün paketler ağdaki bütün cihazlara iletilir ve kendisine yönelik olmayan paketleri bırakma işlemi cihaza bırakılır. Elbette bu durum donanım seviyesinde uygun bir destek ve yazılım seviyesinde (protokol yığını) paketlerin drop edilmemesinin sağlanmasıyla ağda gezen bütün paketlerin izlenebilmesine imkan tanır. Hem bu riski azaltmak ama daha öncelikli olarak artan cihaz sayısı ile broadcastlerin tıkanıklık ve çakışma (Congestion Control ve Collision Avoidance) yaratmasını engellemek için bağlantı katmanının (link layer) seviyesinde anahtarlama yapan Layer 2 switch'ler yada Layer 3 yönlendiriciler (router)

Paket koklama için kabul görmüş iki temel yazılımsal yöntem mevcuttur : Bunlardan birincisi soket arayüzünün bir parçası olan ham soketler kullanmaya dayanır. Her ne kadar Berkley soket arayüzü tanımlarında geçse de geçmişte yalnızca Unix ve Linux türevi işletim sistemlerinin yaratılmasına izin verdiği ham soket yaratma ve kullanma, Windows 2000 ve XP'de yönetici seviyesindeki kullanıcılar için mümkün kılınmış ama daha sonra XP için çıkan ikinci hizmet paketi (Service Pack 2) ile geri iptal edilmiştir. Bugün Vista ve XP işletim sistemleri yönetici seviyesindeki kullanıcılara sınırlı ham soket erişimi imkanı tanımaktadır; buradaki sınır yaratılan ve dinlenen paketlerin o makineye ait MAC ve IP adresini içermesi sayesinde başkalarına yönelik paketlerin izlenememesi ve başka bir makineden geliyormuş gibi tahrif edilmiş paket üretilmemesini sağlar.

Ham soketler ile paket koklama sadece o cihaza yönelik iletişimde IP başlığı dahil olmak üzere veriye erişimi sağlar ama OSI'nin ikinci katmanı seviyesinde bilgi almamıza izin vermez.

İşletim sistemlerinin çekirdek seviyesindeki ağ kartı sürücüsüne özel modüller eklenmesi ile en alt seviyede (Link Layer'da başlıklarıyla beraber MAC paketleri için) paket koklama yapmak mümkündür ama bu bizim konumuzun sınırları içinde bir fayda sağlamayacaktır.

Yazılımımız önceki yöntemlerimizde olduğu gibi kullanıcı arayüzünden oluşabilecek bir girişimi engellemek amacıyla paket koklama kodlarını ayrı bir iş parçasığı üzerinde çalıştırmaktadır. Koklama işlemini başlatacak tuşa kullanıcı arayüzündeki form'da tıklanmasını takiben bağımsız bir iş parçasığını *snifferThreadFunction()* ile ilişkilendirerek yaratır ve başlatır.

7. VERİLERİN İNCELENMESİ

7.1 Veri Eksenleri ve Matematiksel Model

Daha önce de bahsedildiği gibi paket anahtarlama ağılarda hizmet kalitesi (QoS) dört temel ana parametre ile ölçülmektedir. Bunlar sırasıyla bantgeniřliđi/bitrate, paket kayıp oranı, gecikme ve gecikme deđiřkenliđi anlamına gelen jitter'dır*. Her ne kadar jitter gecikme üzerinden hesaplanan türev bir veri olsa da kendi bařına hizmet kalitesi ačíısından belirleyici bilgiler tařımaktadır.

UMTS ađlarına özel olarak QoS parametrelerinin operatör bazında deđiřiklik göstermesi kaçınılmazdır; her operatörün kendine ait anahtarlama ve yönlendirme cihazları, çekirdek ađı, internet ađ gečídi** ve internet omurgasına bađlantısı farklı performanslarda alıřmaktadır. Son olarak da UMTS uygulamalarının büyük ölçüde mobil kullanıcılara hitap ettiđi düşünölecek olursa ölçümler aynı operatör içinde dahi zamana ve konuma göre farklılık gösterecektir. Normal GSM řebekesiyle ses görüřmelerinde de göröldüğü gibi haftanın günleri, günün saatleri ve hatta mevsimsellik trafik yükünde deđiřikliklere sebep olmaktadır; mekansal olarak da yoğun kullanılan köprüleri kapsayan hücreler, futbol maçı ve sinema etkinliklerine bađlı olarak mobil kullanıcıların aşırı lokalize olduđu konumlar ve baz istasyonlarına ok uzakta kaldığı için sinyal gücünün düşük olduđu řehirler arası, tünel ve metro içi gibi yerlerde de bütün hizmetlerin kalitesinde farklılıklar oluşmaktadır.***

Herhangi bir operatör ait hizmet kalitesini modelini matematiksel olarak tanımlayabilmek için řu řekilde bir fonksiyon önermekteyiz: zaman, konum eksenlerinin ölçümler için bir uzay oluşturduđu ve burada her noktanın {**bitrate, packet loss ratio, delay, jitter**} řeklinde bir deđerler dörtlüsü olarak temsil edildiđini düşünürsek; belirli bir 3G operatörü için yeterli sayıda dođru seilmiş (zaman,konum) noktalarında yapılacak ölçüm sonuçları, lineer interpolasyon ile sürekli bir hizmet kalitesi fonksiyonuna dönüřtürölebilir.

* Her ne kadar sinyal gücü de hizmet kalitesi ile iliřkili olsa da, hiçbir QoS parametresi sinyal gücünün doğrudan bir türevi olmadıđından, sinyal gücü ve QoS parametreleri arasındaki korelasyon bu tezin konusu dıřında kalmaktadır.

** Core Network ve Internet Gateway

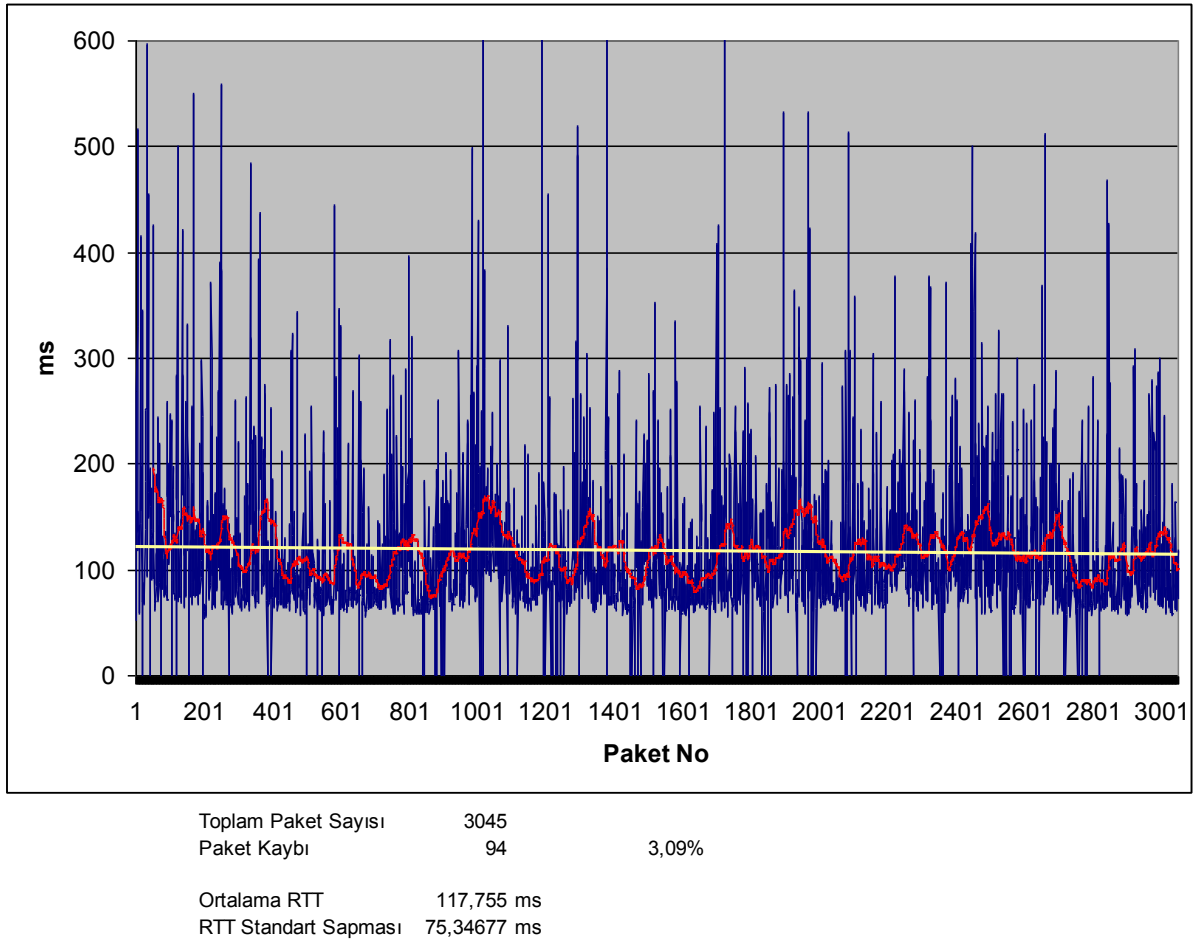
*** Sabah trafiđi ve iř ıkıřı saati, hafta içi ve hafta sonu, yaz tatili dönemleri

7.2 Ölçüm Sonuçları

Hizmet vermekte olan üç GSM operatörüne ait ölçüm sonuçlarını anonimleştirmek adına operatörlerin ticari ünvanları yerine, verileri A, B ve C operatörü olarak gruplandırmanın daha uygun olacağını düşündüğümüzden takip eden kısımlarda bu isimlendirme kullanılmıştır.

7.2.1 A Operatörü

A operatörü yaptığımız ölçümlerde bütün QoS değerleri için en iyi performansı gösteren oldu. Özellikle düşük gecikme ve yüksek bant genişliği ile 3G şebekesinden beklenen değerleri bütün konumlarda sundu. Her ne kadar sinyal seviyesi bu operatörde de 2G seviyesine göre düşük olsa da en geniş 3G kapsama alanını da A operatöründe gözlemledik.



Şekil 7.1 A Operatörü için gecikme ve paket kaybı

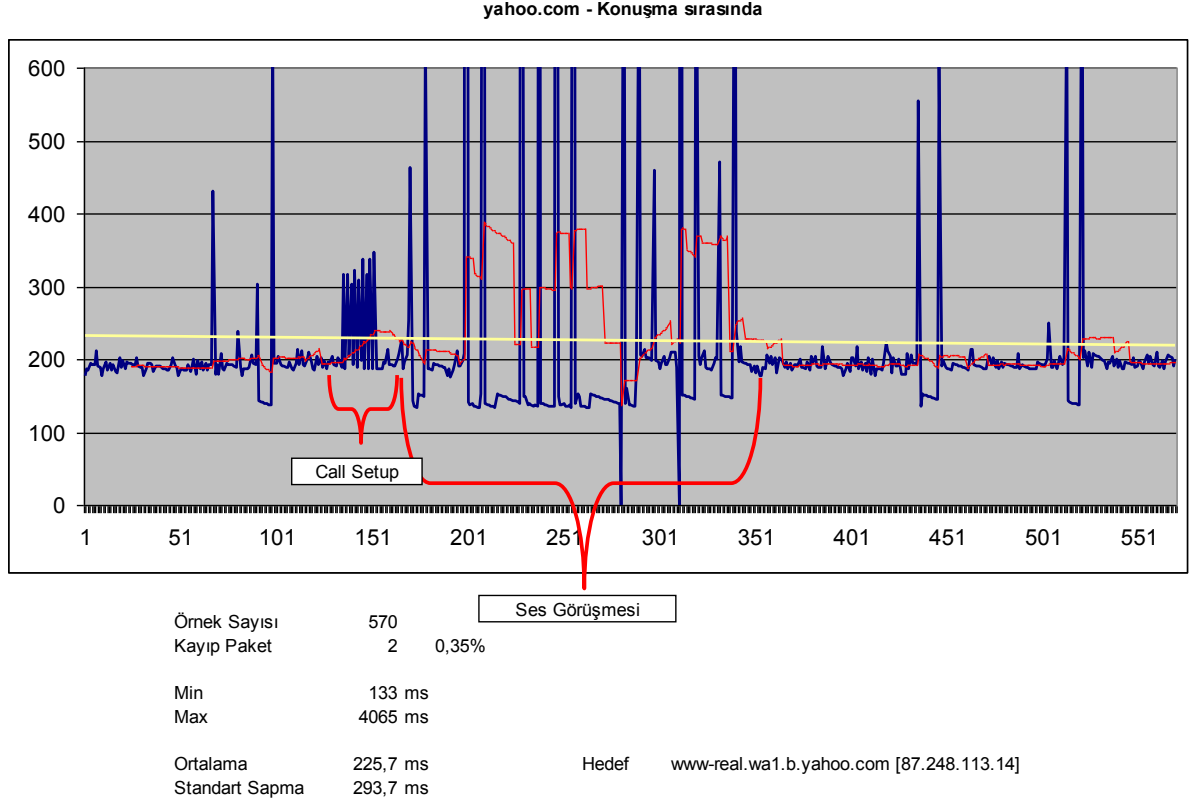
Şekil 7.1’de görüldüğü üzere ölçüm sonuçları A operatörü için IP seviyesinde ortalama 118ms gecikme ve %3’lük bir paket kaybı göstermiştir. TCP akışı kullanıldığında paket kayıpları kullanıcıya yeniden aktarım sebebiyle artan gecikme olarak yansımaktadır. Gecikmenin standart sapması 75ms mertebesinde dir.



Şekil 7.2 A operatörü için kullanıcı ekipmanı üzerinden bant genişliği ve gecikme ölçümü

TCP akışı ölçümleri A operatörü için şekil 7.2'deki gibi ortalama 500kbit'e varan bant genişliklerinin mümkün olduğunu öte yandan paket kayıplarının yeniden aktarımı ile birlikte gecikmenin en kötü durumda 2000ms'ye kadar çıkabildiğini göstermiştir.

Aşağıdaki verilerde de gözüktüğü gibi normalde 200ms civarında gecikmeli bir IP akışı, şekil 7.3'de görüldüğü gibi aynı anda bir ses çağrısı gelmesi durumunda büyük bir gecikme değişkenliğine uğramaktadır. Bu akış sırasında normalde görülmeyen paket kayıpları ses görüşmesi ile birleştiğinde iki kez gözlemlenmiş ama ses görüşmesinin bitmesine takiben bütün değerler normale dönmüştür.



Şekil 7.3 A operatöründe veri aktarımı sırasında ses görüşmesinin gecikmeye etkisi

Bir önceki ölçüm konumundan farklı olarak bu konumda yapılan ölçümlerde (eş zamanlı ses görüşmesi sırasında gerçekleşen iki istisna hariç) hiç paket kaybı gözlenmemiştir.

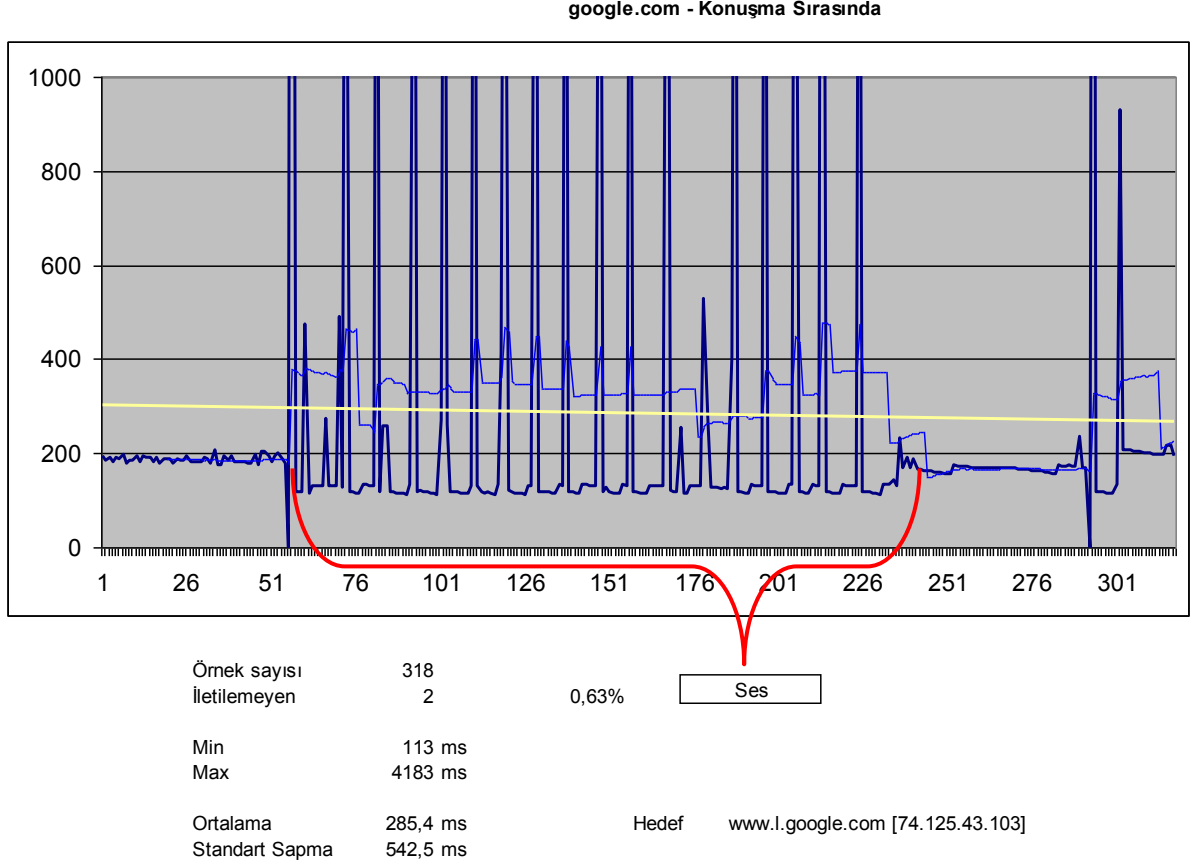
	Konum1	Konum2	Konum3	Konum4	Konum5
A	377	579	514	427	498

Şekil 7.4 A operatörü konumsal ölçüm sonuçları (kbit/sn)

Yukarıdaki şekil 7.4’de A operatörü için konuma göre bant genişliği ölçüm sonuçları görülmektedir. Takip eden bölümlerde de kullanılan bu beş konum *Konumsal Matematiksel Model* bölümünde görülen harita da işaretli olan noktalara denk gelmektedir.

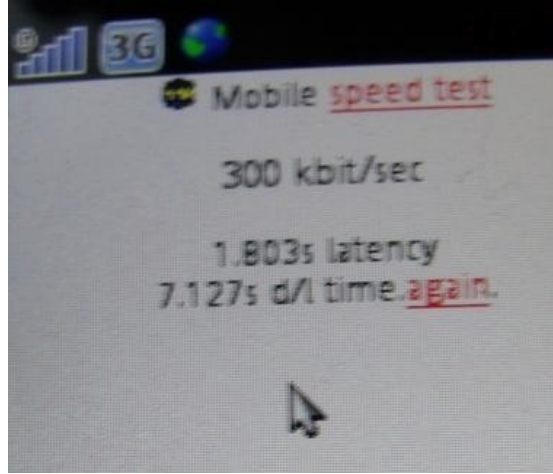
7.2.2 B Operatörü

Yaptığımız ölçüm ve gözlemler B operatörünün çok daha sınırlı bir 3G kapsama alanına sahip olduğunu; gecikme değerlerinin A operatörü ile karşılaştırılabilir bir seviyede olduğunu ama bant genişliği olarak A operatörünün gerisinde kaldığını göstermektedir.



Şekil 7.5 B operatöründe gecikme ve veri aktarımı sırasında ses görüşmesinin etkisi

Yukarıdaki şekil 7.5’deki verilerde de gözüktüğü gibi ortalama 200ms civarında gecikmeli IP akışı eş zamanlı ses çağrısı gelmesi durumunda A operatöründe gözlenene çok benzer bir gecikme değişkenliği göstermektedir. Bu durumun iki operatörde de gözlemlenmesi, bu fenomenin operatörden çok UMTS ağlarının doğasından yada ölçüm için kullanılan cihazların multitasking için yeterli işlem gücüne sahip olmamasından kaynaklandığını düşündürmektedir.



Şekil 7.6 B operatörü için kullanıcı ekipmanı üzerinden bant genişliği ve gecikme ölçümü

TCP akışı ölçümleri B operatörü için şekil 7.6'daki gibi ortalama 300kbit/sn mertebesinde bant genişliklerinin mümkün olduğunu ve gecikmenin bu operatörde de 2000ms'ye kadar çıkabildiğini göstermiştir.

	Konum1	Konum2	Konum3	Konum4	Konum5
B	583	270	412	486	450

Şekil 7.7 B operatörü konumsal ölçüm sonuçları (kbit/sn)

Yukarıdaki şekil 7.7'de B operatörü için konuma göre bant genişliği ölçüm sonuçları görülmektedir.

7.2.3 C Operatörü

C operatörü çok az konumda ölçüm yapmak için elverişli sinyal kapsama alanı sunduğu için bu lokasyonlar ile sınırlı ölçümler zaman zaman kendi içinde çelişen sonuçlar doğurdu. C operatörü için 100kbit ile 250 kbit arasında dalgalanmalar gösteren bir UDP akışı ölçülebilmıştır.

	Konum1	Konum2	Konum3	Konum4	Konum5
C	203	309	224	210	288

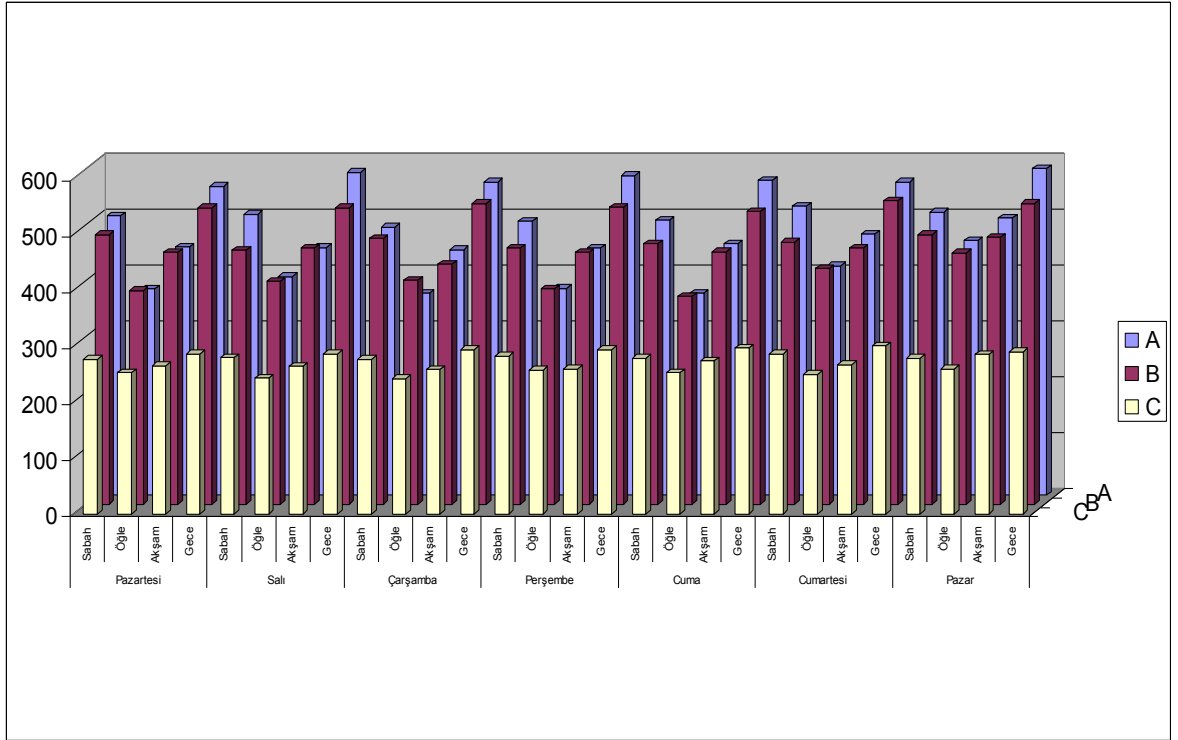
Şekil 7.8 C operatörü konumsal ölçüm sonuçları (kbit/sn)

Yukarıdaki şekil 7.8'de C operatörü için konuma göre bant genişliği ölçüm sonuçları görülmektedir.

ICMP paketleri ile de 400ms ile 3000ms arasında gecikmeler kaydedilmiştir. Paket kayıpları normalde %6'ya yakın değerlerde ve veri iletiminin drop edilmesinin hemen öncesinde %25'lere kadar yükselmektedir.

7.2.4 Üç Operatörün Karşılaştırılması

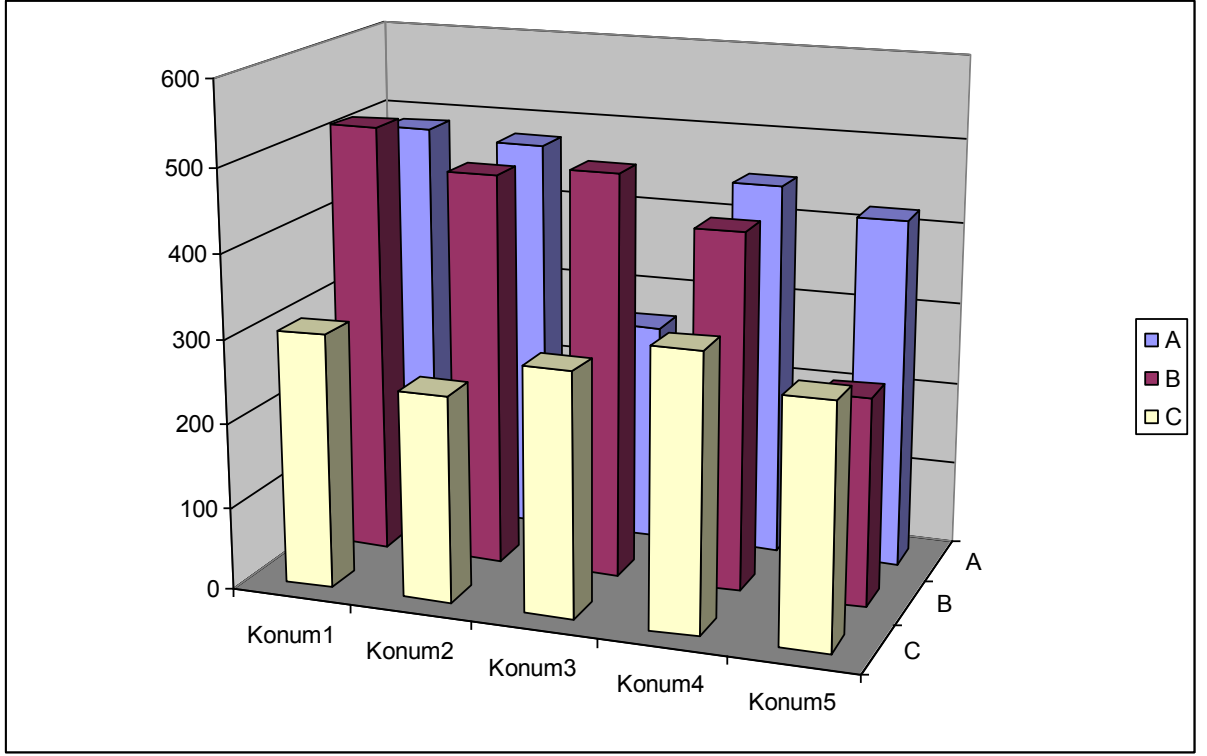
Yukarıdaki incelemelerde de görüldüğü gibi en yüksek performansı en gelişmiş altyapıya sahip olduğu için A operatörü sunmaktadır. B operatörü bir çok QoS değeri için A operatörünün hemen altında kalsa da 3G kapsama alanı olarak çok daha dar bir alanda hizmet veriyor olması B operatörünü hareket etmeyen kullanıcılar için verimli yapıyor; B operatörünün 3G baz istasyonlarına yatırım yapması halinde A operatörü ile eşdeğer performansta hizmet vermesi beklenir.



Şekil 7.9 Operatörlerin bant genişliği performansının zamansal karşılaştırması

Şekil 7.9'daki operatörlerin bant genişliği performanslarını zaman ekseninde karşılaştıran grafikten de görüldüğü gibi C operatörünün QoS parametreleri, kapsama alanı ve sinyal seviyesi gibi bütün alanlarda diğer iki operatörün çok gerisinde kaldığı gözlemlendi. Fiyat avantajı sayesinde elde ettiği A'dan az ama B'den çok olan toplam müşteri sayısı sebebiyle göstermelik olarak kamudan 3G lisansı aldığını düşünmek son derece yerinde gözükmemektedir, çünkü ne sıradan kullanıcılar için, ne de kritik iş uygulamalarını desteklemek için uygun bir hizmet yada alt yapı sunamamaktadır. Üç operatör için de ölçüm yapılabilen konumlardan beş

tanesinde karşılaştırma amacıyla toplanan veriler bize bu operatörleri mekansal anlamda da karşılaştırma imkanı vermektedir. Aşağıdaki şekil 7.10'daki grafikte de görüldüğü gibi, operatörler ekseninde görülen performans farkı zamansal grafikte gözlemlenen ile benzerlik göstermekte ama operatörlerin performansa göre sıralanmasında konuma göre değişkenlik zamansal olandan çok daha fazladır.



Şekil 7.10 Operatörlerin bant genişliği performansının mekansal karşılaştırması

Bu durumun iki olası açıklaması olabilir; bunlardan birincisi her bir konum için operatörlerin baz istasyonu mesafelerinin değişik olması sebebiyle farklı konumlar operatörlere farklı avantajlar sunabilmektedir.* Operatörlerin mekansal olarak heterojen performans dağılımını zamansal karşılaştırmada görülmemesi sağduyu ile uyumludur çünkü zaman ekseninde operatörlerin farklı konumlanmasından bahsedilemez. Yukarıdaki grafikte ölçüm sonuçları görülen beş konum, *konumsal matematik model* başlığında görülen haritada işaretlenmiş olan beş noktaya denk gelmektedir.

* Örneğin *Konum3*'te A operatörünün belirgin şekilde geride kalması, A operatörüne ait baz istasyonunun uzakta kalması ve buna bağlı olarak sinyal/gürültü oranının düşük olmasından kaynaklandığı düşünülebilir.

7.2.5 Konumsal Matematik Model

Bant genişliğinin değerlerini uzaysal olarak modellemek için İstanbul haritasının belirli bir kısmı 30'a 26'lık kare hücrelerden oluşan bir ızgara ile bölünmüş, ve A operatörü için yapılan ölçümlerde elde edilen bant genişliği değerleri denk geldikleri hücrelere değer olarak atandıktan sonra diğer hücreler özel bir interpolasyon/ekstrapolasyon ile hesaplanmıştır.

İlk aşamada sadece interpolasyon amacıyla kurduğumuz model basit bir ağırlıklı ortalama fonksiyonundan oluşmaktaydı. Bu modelde her hücrenin değeri, ölçüm değerlerinin uzaklıklarının tersinin karesi ile ağırlıklandırıldığı bir ortalama ile elde ediliyordu.

Uzaklık fonksiyonu

$$U(x, y, i, j) = \sqrt{(x - i)^2 + (y - j)^2} \quad (7.1)$$

şeklinde iki boyutlu kartezyen uzayda iki noktadan reel sayılara $R^2 \times R^2 \rightarrow R$ bir fonksiyon olsun. Ağırlıklı ortalamanın her terimi için kullandığımız katsayı fonksiyonu **K** da bu nokta için **U** fonksiyonuna göre uzaklığın karesinin tersi olsun.

$$K = \frac{1}{U^2} = \frac{1}{\left(\sqrt{(x - i)^2 + (y - j)^2} \right)^2} = \frac{1}{(x - i)^2 + (y - j)^2} \quad (7.2)$$

Bu durumda n adet noktada yapılan ölçüm değeri $g_{1..n}$ ise (x,y) noktasındaki interpolasyon değerini veren ağırlıklı ortalama fonksiyonu **A** şu şekilde olur.

$$A(x, y) = \sum_{t=1}^n K_t g_t \quad (7.3)$$

Bu modelde ölçüm noktaları arasında kalan alandaki değerler ölçüm değerlerine yakınsarken bu alandan uzaklaşıldıkça sonuçlar sıfıra yaklaşmaktadır.

Ölçüm noktalarının dışarıya doğru da etki gösterebilmesi için modeli doğrudan ölçüm değerlerinin ağırlıklı toplamları yerine, bu ölçümlerin ortalaması üzerine eklenen ağırlıklı sapma toplamları olarak değiştirirsek ızgara değerleri ölçüm noktaları arasında kalan alandaki değerler ölçüm değerlerine yakınsarken bu alandan uzaklaşıldıkça ölçümler ortalamasına yaklaşır.

Modeli bu şekilde güncelleştirmek için ağırlık katsayısı fonksiyonu **K** ve ölçüm değerleri **g** aynı kalırken; **g** değerlerinin ortalaması λ ve (x,y) noktasındaki değeri veren yeni **A** fonksiyonu şu şekilde olur :

$$\lambda = \frac{1}{n} \sum_{t=1}^n g_t$$

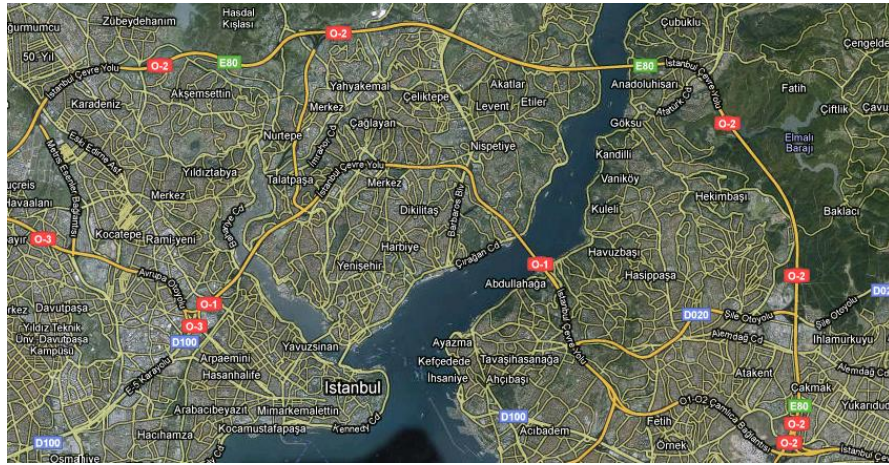
$$A(x, y) = \lambda + \sum_{t=1}^n K_t (g_t - \lambda) \quad (7.4)$$

Yukarıdaki 7.4 fonksiyonu kullanarak beş noktada yapılan ölçümlere göre hesaplanan değerler şekil 7.11'deki gibidir. Şekil 7.12 ve 7.13'de ise sonuçlar harita üzerinden bölgesel olarak gösterilmiştir.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
1	250,7	250,7	250,7	250,7	250,6	250,6	250,6	250,6	250,6	250,6	250,6	250,6	250,6	250,6	250,6	250,6	250,6	250,6	250,6	250,7	250,7	250,7	250,7	250,7	250,8	250,8	250,8	250,8	250,8	250,8	
2	250,7	250,7	250,7	250,6	250,6	250,6	250,6	250,6	250,5	250,5	250,5	250,5	250,5	250,5	250,5	250,5	250,6	250,6	250,6	250,6	250,6	250,7	250,7	250,7	250,7	250,8	250,8	250,8	250,8	250,8	
3	250,7	250,7	250,6	250,6	250,6	250,6	250,5	250,5	250,4	250,4	250,4	250,4	250,4	250,4	250,4	250,5	250,5	250,5	250,5	250,5	250,5	250,6	250,6	250,6	250,7	250,7	250,8	250,8	250,8	250,8	
4	250,7	250,6	250,6	250,6	250,6	250,5	250,5	250,4	250,3	250,2	250,2	250,2	250,1	250,2	250,2	250,3	250,4	250,4	250,4	250,4	250,4	250,4	250,4	250,5	250,6	250,6	250,7	250,8	250,8	250,8	250,8
5	250,6	250,6	250,6	250,6	250,5	250,5	250,4	250,3	250,2	250,0	249,8	249,7	249,8	249,8	250,0	250,1	250,2	250,3	250,3	250,2	250,2	250,2	250,2	250,4	250,5	250,6	250,7	250,7	250,8	250,8	250,8
6	250,6	250,6	250,6	250,5	250,5	250,4	250,3	250,2	249,9	249,5	248,9	248,4	248,9	249,5	249,9	250,1	250,1	250,1	249,9	249,8	249,7	249,9	250,1	250,3	250,5	250,7	250,7	250,8	250,8	250,9	250,9
7	250,6	250,6	250,5	250,5	250,4	250,4	250,2	250,1	249,7	248,8	246,2	241,8	246,2	248,8	249,7	250,0	250,0	249,8	249,4	248,9	248,7	249,0	249,6	250,1	250,4	250,6	250,8	250,8	250,9	250,9	250,9
8	250,6	250,5	250,5	250,4	250,4	250,3	250,2	250,0	249,5	248,3	241,7		241,8	248,8	249,6	249,9	249,8	249,4	248,4	246,7	245,6	246,9	248,6	249,8	250,3	250,7	250,8	250,9	250,9	250,9	250,9
9	250,6	250,5	250,5	250,4	250,3	250,2	250,1	249,9	249,5	248,7	246,1	241,7	246,1	248,8	249,7	249,9	249,8	249,1	246,9	239,9	228,1	240,0	247,1	249,5	250,4	250,7	250,9	251,0	251,0	251,0	251,0
10	250,5	250,5	250,4	250,3	250,3	250,1	250,0	249,8	249,6	249,2	248,6	248,2	248,7	249,5	250,0	250,1	250,0	249,1	246,0	228,4		228,6	246,4	249,6	250,6	250,9	251,1	251,1	251,1	251,1	251,1
11	250,5	250,4	250,4	250,3	250,2	250,0	249,9	249,7	249,5	249,3	249,2	249,3	249,6	249,9	250,3	250,5	250,4	249,8	247,8	240,9	229,3	241,2	248,3	250,4	251,1	251,3	251,3	251,2	251,2	251,2	251,1
12	250,5	250,4	250,3	250,2	250,0	249,9	249,6	249,4	249,2	249,1	249,2	249,4	249,8	250,2	250,7	251,0	251,2	251,0	250,4	249,1	248,4	249,8	251,3	251,9	251,9	251,8	251,6	251,4	251,3	251,2	251,2
13	250,4	250,4	250,2	250,1	249,9	249,6	249,3	248,9	248,5	248,4	248,6	249,1	249,7	250,4	251,1	251,8	252,4	252,6	252,8	253,3	254,3	255,1	254,9	254,0	253,1	252,4	252,0	251,7	251,5	251,3	251,3
14	250,4	250,3	250,2	250,0	249,7	249,3	248,7	247,8	246,9	246,5	247,1	248,2	249,4	250,5	251,8	253,5	254,8	255,0	255,4	257,4	261,5	264,4	261,5	257,3	254,6	253,2	252,3	251,9	251,6	251,4	251,4
15	250,4	250,3	250,1	249,9	249,5	248,9	247,8	245,9	242,9	241,0	243,1	246,4	248,8	250,6	252,9	257,9	265,8	260,0	258,2	262,7	279,9	309,3	279,7	262,0	256,1	253,8	252,7	252,0	251,7	251,4	251,4
16	250,4	250,3	250,1	249,8	249,4	248,6	246,9	242,9	230,7	210,4	231,0	243,5	248,0	250,5	253,5	265,0		267,5	259,8	266,1	309,8		309,5	265,1	256,9	254,1	252,8	252,1	251,7	251,5	251,5
17	250,4	250,3	250,1	249,8	249,3	248,4	246,5	240,9	210,3	210,6	241,4	247,5	250,2	252,8	258,0	266,0	260,3	258,5	263,1	280,4	309,7	280,0	262,3	256,4	253,9	252,8	252,1	251,7	251,5	251,5	251,5
18	250,4	250,3	250,1	249,8	249,4	248,6	247,0	242,9	230,8	210,5	231,0	243,4	247,9	250,1	251,8	253,7	255,2	255,6	256,2	258,5	262,6	265,4	262,4	257,9	255,1	253,5	252,6	252,0	251,7	251,5	251,5
19	250,4	250,3	250,2	249,9	249,6	249,0	247,9	246,0	243,0	241,0	243,2	246,4	248,7	250,1	251,2	252,2	253,0	253,7	254,4	255,4	256,6	257,2	256,5	255,1	253,8	252,9	252,3	251,9	251,6	251,4	251,4
20	250,4	250,4	250,2	250,0	249,8	249,4	248,8	247,9	247,1	246,7	247,2	248,3	249,4	250,3	251,0	251,6	252,2	252,7	253,2	253,7	254,1	254,3	254,1	253,5	252,9	252,4	252,0	251,7	251,5	251,4	251,4
21	250,5	250,4	250,3	250,2	250,0	249,7	249,4	249,0	248,7	248,6	248,9	249,3	249,9	250,4	250,9	251,3	251,7	252,1	252,4	252,7	252,9	253,0	252,9	252,6	252,3	252,0	251,8	251,6	251,4	251,3	251,3
22	250,5	250,5	250,4	250,3	250,1	250,0	249,8	249,7	249,5	249,5	249,7	249,9	250,2	250,5	250,9	251,2	251,4	251,7	251,9	252,1	252,2	252,2	252,2	252,1	251,9	251,8	251,6	251,5	251,3	251,2	251,2
23	250,6	250,5	250,4	250,4	250,3	250,2	250,1	250,0	250,0	250,0	250,1	250,2	250,4	250,6	250,8	251,1	251,2	251,4	251,6	251,7	251,8	251,8	251,8	251,7	251,6	251,5	251,4	251,3	251,3	251,2	251,2
24	250,6	250,5	250,5	250,5	250,4	250,3	250,3	250,3	250,2	250,3	250,3	250,4	250,6	250,7	250,8	251,0	251,1	251,3	251,4	251,4	251,5	251,5	251,5	251,5	251,4	251,4	251,3	251,2	251,2	251,1	251,1
25	250,6	250,6	250,6	250,5	250,5	250,5	250,4	250,4	250,4	250,4	250,5	250,6	250,6	250,7	250,8	250,9	251,0	251,1	251,2	251,3	251,3	251,3	251,3	251,3	251,3	251,3	251,2	251,2	251,1	251,1	251,1
26	250,6	250,6	250,6	250,6	250,6	250,5	250,5	250,5	250,5	250,5	250,6	250,6	250,7	250,8	250,8	250,9	251,0	251,1	251,1	251,2	251,2	251,2	251,2	251,2	251,2	251,2	251,1	251,1	251,1	251,1	251,0

Y	X	Sapma	Bant genişliği
16	22	59,2	310
16	17	14,2	265
17	10	-40,8	210
10	21	-23,8	227
8	12	-8,8	242
ortalama			250,8

Şekil 7.11 A operatöründe beş noktaya göre hesaplanmış ekstrapolasyon ızgarası



Şekil 7.12 Ölçümlerin yapıldığı bölge [4]



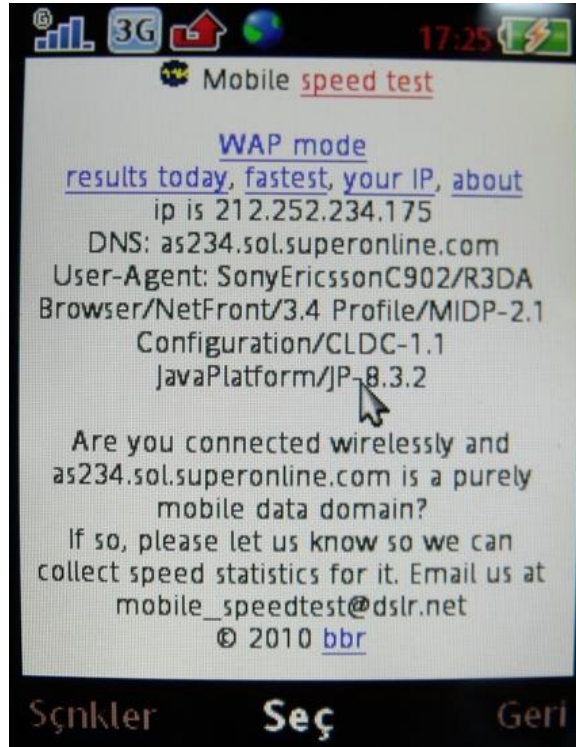
Şekil 7.13 A operatörü için gecikme ızgarası ve haritanın süperpozisyonu

Haritada gösterilen beş kırmızı alan sol üst köşeden başlayarak saat yönünde sırasıyla şu konumlara denk gelmektedir:

- İki çevre yolu arasındaki bağlantının Nurtepe kısmı
- YTÜ kampüsü
- GSÜ kampüsü
- Valikonağı Caddesi, Nişantaşı
- Birinci çevre yolu viyadüğünün başlangıcı

7.2.6 3G-Internet Ağ Geçidi

A operatörünün internet ağ geçidini* konumlandırmak için WAP üzerinden sunucuya request gönderen IP'yi ve bunun ters DNS çözümünü istemciye geri gönderen web hizmetinden yararlanılmıştır. Aşağıdaki şekil 7.14'deki ekran görüntüsünde de gözüktüğü gibi A operatörü Internet ağ geçitlerini kendi bünyesinde değil ama ticari bir internet hizmet sağlayıcısının sunucuları üzerinde barındırmaktadır. Buradan GSM operatörü ile internet hizmet sağlayıcısı arasındaki bağlantının da bir gecikmeye sebep olacağını varsaymak yerinde olacaktır.



Şekil 7.14 A operatörünün Internet bulutu tarafından gözüken ağ geçidi

* Internet Gateway

7.2.7 Kullanıcı Ekipmanının (UE) Modem Olarak Performansı

Ölçümler sırasında dikkatimizi çeken bir diğer konu, birçok kullanıcı ekipmanının* şekil 7.16’da görüldüğü gibi 3G ağından faydalansa bile, bilgisayara gerek kablo, gerek bluetooth ile bağlandığında kendisini layer 2’deki LLC (Link Level Control) ile megabit’ten daha yavaş bir iletişim cihazı olarak tanıtmaması oldu. Cihaz kendi iletim üst sınırını bağlantı anında şekil 7.15’deki gibi yaklaşık 500kbit/saniye olarak bildirdiği için, bağlandığı bilgisayarın işletim sistemi hiçbir zaman bu cihaza belirttiği hızın üstünde veri göndermeyi denemeyecektir.

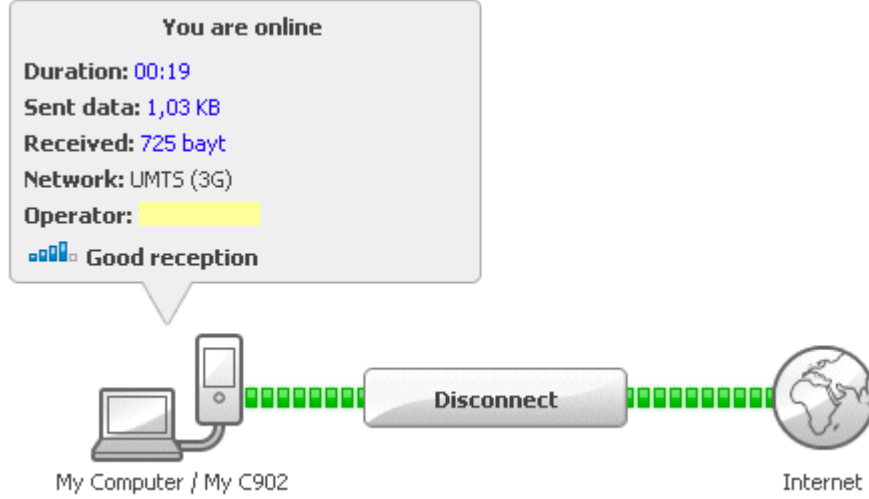


Şekil 7.15 Kullanıcı ekipmanının işletim sistemine bildirdiği kapasite

Bu durumun olası sebepleri hakkında spekülasyon yapmamız gerekirse; her ne kadar UMTS şebekesi çok daha yüksek hızlara izin veriyor olsa da, kullanıcı ekipmanı en baştan bunun sınırlı bir kısmını kullanacak şekilde tasarlanmış olabilir. Bir çok cep telefonu sınırlı bir işlem gücü ve bellek kapasitesine sahip olduğu için iki ağ arasında (UMTS <-> LAN IP) yönlendirme yapması gerektiğinde bildiğimiz anahtarlama yada yönlendirme cihazları gibi 10 megabitler seviyesinde hızlarda iletim yapamaz ve gereken tampon bellekleri tahsis edemez.

Benzer şekilde kullanıcı ekipmanının bilgisayar ile iletişimini düzenlemek için işletim sistemine yüklenen üreticiye özgü sürücü yazılımında tam bant genişliğinde hizmet verecek şekilde kodlanmadıysa, bu sürücü iletişim zincirinde bir dar boğaz oluşturabilir.

* Cep telefonu, USB modem



Şekil 7.16 Bilgisayarın internete çıkışında UMTS kullandığının teyidi

Bu bağlamda düşünüldüğünde UMTS ağından sonra sırasıyla cep telefonunun işlemcisi, tampon bellekteki kuyrukları, bilgisayar ile iletişim kurmak için kullandığı USB kablo yada bluetooth tabanlı protokolün kendi doğal üst sınırı ve bilgisayardaki işletim sistemine yüklenene sürücü bir dar boğazlar zinciri oluşturarak kullanılabilen efektif bant genişliğini bu zincirin en dar halkasına indirgemektedir.

7.2.8 UMTS Spesifik Baz İstasyonları

Bilindiği üzere UMTS şebekesini kullanmak için mevcut 2.5G altyapısına ait baz istasyonlarını kullanmak mümkün olmamaktadır; bu sebeple üç operatör de ülke genelinde 3G destekleyen yeni baz istasyonları kurdular. Telefonları 3G'yi destekleyen kullanıcılardan hatları için 3G'yi aktive edenler* GSM operatörü tarafında ilgili SIM kartın artık bu yeni baz istasyonları ile ilişkilendirilmesiyle bu hizmetlerden faydalanmaktalar. Bu yeni baz istasyonları A operatörü için ülke genelinde 16.000 adet gibi bir sayı ile başlamış olsa da B ve C operatörlerinde bu sayı kayda değer bir biçimde azdır.

A operatörü için bile, 3G hizmeti aktive edildikten sonra, kullanıcı ekipmanının bildirdiği sinyal seviyesinin bütün konumlarda eskisine göre daha düşük olduğu, ve çoğu konumda verimsiz bir şekilde sinyal gücünün yetersizliğinden dolayı hattın 2G baz istasyonuna devredilip hemen ardından yine 3G baz istasyonuna geri devredildiği ve bunun uzun süreler tekrarladığı gözlemlenmiştir.

* Genellikle operatöre gönderilen bir kısa mesaj ile o hizmet numarasının yeni bir set baz istasyonu ile ilişkilendirilmesi şeklinde aktive edilir.

Bu devirler sırasında mevcut bir ses çağrısı, video çağrısı yada data iletişimi varsa bu çağrıların drop edilmediği ama video çağrılarının görsel kısmının drop edilerek sadece ses çağrısına indirgendiği, ve mevcut veri iletişiminin de benzer şekilde kesilmeden kullanılabilen baz istasyonuna göre 3G, HSCSD ve EDGE kipleri arasında sürekli değiştiği gözlemlenmiştir.

8. SONUÇ VE ÖNERİLER

Üç farklı operatöre ait cep telefonu yada USB 3G modem kullanılarak yapılan ölçümlerde mobil noktadan aynı ulusal internet omurgası üzerindeki bir diğer ADSL çıkışı arasındaki efektif bant genişliği ortalama 150 ile 300 kbps arasında ölçülmüştür; bant genişliği dağılımı doğal dağılım (gauss dağılımı) ile örtüşmektedir. Bu ölçülen değer teorik 14 MBps tepe değeri bant genişliğinin çok altında kalmaktadır. Aynı operatörlerin 100 ile 130 kbps arasında bant genişliği sağlayan EDGE hizmeti ile karşılaştırıldığında, 3G taşınan birim veri başına fiyat avantajı sağlamazken, bant genişliği yaklaşık 1.5 kat yüksek ve ağ gecikmesi de ortalama 110ms değerindedir.

3G UMTS ağlarının performansı IP tabanlı veri iletimine özel olarak daha önceki 2.5G olarak nitelendirilen EDGE ile karşılaştırıldığında kullanıcılara getirdiği ekstra maliyeti haklı kılmamaktadır. EDGE ile 3G arasında tepe değeri IP paket bant genişliği açısından iki kat gibi bir fark olsa da, ortalama iletişim hızının çoğu kez EDGE ile elde edileni geçemediği ve 3G'nin gecikme değerleri sıklıkla EDGE ile aynı yada daha fazla gözlemlenmiştir. Aynı zamanda UMTS ve 3G şebekesinin çalışması için tamamen yeni baz istasyonlarının kurulmuş olması, bu yeni baz istasyonlarının sayısının azlığı nedeniyle 3G'ye ait kapsama alanının sınırlı kalmasına sebep olmuş ve sinyal seviyesini sıklıkla düşük olarak gözlemlememize sebep olmuştur. Bu düşük sinyal seviyesinin kullanıcı ekipmanını daha yüksek güçte sinyaller göndermeye zorlayacağını ve bu nedenle cihazın pile bağlı bekleme ve konuşma sürelerini düşürürken, oluşan elektromanyetik profili ve SAR değerlerini arttıracaklarını varsaymak mümkündür.

Bu çalışmanın devamı niteliğinde çalışmalar yapılması halinde bazı hususların daha detaylı incelenmesinde fayda vardır. Örneğin daha fazla noktada ve daha uzun süreleri kapsayan örneklemeler ile daha çok veri oluşturulması, mekansal ve zamansal modellerin birleştirilerek $A(x,y,t)$ formunda sabit bir fonksiyonun türetilmesi ya da x,y ve t 'nin sürekli değerleri için sürekli bir fonksiyon türetilmesi, kullanıcı ekipmanlarının ulaşabileceğinden daha alt seviyede verilere ulaşabilen operatörlere has cihazlar ile ölçüm yapılması, sonuçların Ns-2 gibi kabul görmüş ağ benzetimi yazılımlar ile yapılacak simülasyon bazlı karşılaştırılması göz önüne alınmalıdır. Daha da ileri düzeyde ise QoS parametlerinin 4G temelli ölçüm ve performansları değerlendirilebilir.

KAYNAKLAR

Büyükbaş, A., (2005), “Cdma Ve Umts: Üçüncü Nesil Mobil Haberleşme Teknolojilerinin Karşılaştırılması”, Uzmanlık Tezi, Telekomünikasyon Kurumu, Ankara.

Eckert, C., Mattheß, M., Ritscher, M., Schoo, P., Prasad, A., Laganier, J., Zugenmaier, A., Marx, R., Bayarou, K.M., (2006), “Experimental performance evaluation of secure, mobile IP-based, 3G-WLAN handover, 9th International Symposium on Wireless Personal Multimedia Communications (WPMC)”, San Diego, CA., USA.

Holma, H. ve Toskala, A., (2000), WCDMA for UMTS, Radio Access For Third Generation Mobile Communication, John Wiley & Sons, Inc., NewYork.

Inamura et al, H., (2002), “TCP over 2.5G and 3G Wireless Networks”, IETF Draft.

Karim, M.R. ve Sarraf, M., (2002), W-CDMA and CDMA2000 for 3G Mobile Networks, McGraw-Hill, New York.

Khafizov, F. ve Yavuz, M., (2000), “TCP Over CDMA2000 Networks”, Internet Draft.

Mun Choon, C. ve Ramachandran, R., (2002), “TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation”, Mobicom’02, Atlanta, Georgia.

Richardson, A., (2005), WCDMA Design Handbook, Cambridge University Press, UK.

Sari R.F., Purnamasari P.D., Zaini F., Gultom A.D., (2008), “Performance Evaluation of IP Based Multimedia Services in UMTS”, Informatica Economica Vol. XII/No.3, Romania, ISSN 1842 - 8088.

Sehier, P. ve Urie, A., (2001), “Standardization of 3G Mobile Systems”, Alcatel Telecommunications Review, Paris.

Wee Lum, T. ve OnChing, Y., (2005), “Measurement-based Performance Model of IP Traffic over 3G Networks”, Proc. IEEE TENCON, Melbourne, Australia.

Zagnal, R.Y. ve Khan, J.I., (2005), “EFSM/SDL modeling of the original TCP standard (RFC793) and the Congestion Control Mechanism of TCP Reno”, Technical Report, Internetworking and Media Communications Research Laboratories, Department of Computer Science, Kent State University.

Zheng, H., Rittenhouse, G., Recchione, M., (2003), “The Performance of Voice Over IP over 3G Downlink Shared Packet Channels Under Different Delay Budgets”, Vehicular Technology Conference, IEEE.

İNTERNET KAYNAKLARI

- [1] 3G Tutorial, UMTS Overview: [Eriřim Tarihi: 14.07.2009]
<http://www.umtsworld.com/technology/overview.htm>
- [2] 3GPP Resmi İnternet Sitesi: [Eriřim Tarihi: 14.07.2009]
<http://www.3gpp.org/>
- [3] International Engineering Consortium: [Eriřim Tarihi: 14.07.2009]
www.iec.org
- [4] Google Map: [Eriřim Tarihi: 14.07.2009]
<http://maps.google.com/>
- [5] Software Engineer Training [Eriřim Tarihi: 14.07.2009]
www.software-engineer-training.com/
- [6] UMTS Family Protocols: [Eriřim Tarihi: 14.07.2009]
<http://www.protocols.com/>

EKLER

Ek 1a	Program giriş noktası main() metodu
Ek 1b	tcpListenerThread Sınıfı
Ek 1c	tcpThread Sınıfı
Ek 1d	udpThread Sınıfı
Ek 1e	Sniffing tuşuna tıklanması durumunda çalışan kod bloğu
Ek 1f	Paket koklayıcı iş parçacığına ait ana fonksiyon
Ek 1g	Yakalanan paketlerin DNS çözümünü yapan ve kuyruğa ekleyen kod bloğu

Ek 1a Program giriş noktası main() metodu

```
public static void Main (string[] args)
{
    log("Startup");
    var tlt = new tcpListenerThread();
    log("tcp listener thread started");
    var ut = new udpThread();
    log("udp thread started");
}
```

Ek 1b tcpListenerThread Sınıfı

```
class tcpListenerThread{
    public Thread t;

    public tcpListenerThread(){
        t = new Thread(new ThreadStart(tcpListenerThreadFunction));
        t.Start();
    }

    public void tcpListenerThreadFunction(){
        TcpListener tl = new TcpListener(1337);

        tl.Start();
        while(true){
            var tcpt = new tcpThread(tl.AcceptSocket());
            //sonsuz döngüde gelen soket bağlantılarını kabul et
        }
    }
} //tcpListenerThread class
```

Ek 1c tcpThread Sınıfı

```
class tcpThread {
    Socket sck;
    Thread t;
    Stopwatch _stopWatch = new Stopwatch();

    public tcpThread(Socket s){
        sck = s;
        log("starting tcp thread for "+s.ToString());
        t = new Thread(new ThreadStart(tcpThreadFunction));
        t.Start();
    }

    public void tcpThreadFunction() {
        byte[] buffer = new byte[_bufferSize];
        int bufferLevel;
        int totalBytes=0;

        _stopWatch.Reset();
        _stopWatch.Start();

        while(sck.Connected){
            while(sck.Available > 0){
                bufferLevel = sck.Receive(buffer);
                sck.Send(buffer,bufferLevel,SocketFlags.None);
                totalBytes += bufferLevel;
                log("TCP - Bufferlength: "+bufferLevel.ToString()
                    +throughputStr(totalBytes,_stopWatch));
            }
            Thread.Sleep(0);
        }
        log("TCP " +sck.Handle.ToString()+" - Client Disconnected");
    } //Thread Function
} /
```

Ek 1d udpThread Sınıfı

```

class udpThread{

    public Thread t;
    public UdpClient udpc;

    public udpThread(){
        udpc = new UdpClient(1337);
        udpc.Client.ReceiveBufferSize = _bufferSize;
        udpc.Client.SendBufferSize = _bufferSize;
        t = new Thread(new ThreadStart(udpThreadFunction));
        t.Start();
    }

    public void udpThreadFunction(){

        byte[] buffer;
        IPEndPoint ipe = null;

        while(true){
            //Soket veri pompası, UDP Thread sonsuz döngüde
            while(udpc.Available > 0){
                buffer = udpc.Receive(ref ipe);
                udpc.Send(buffer,buffer.Length,ipe);
                log("UDP:" +ipe.ToString()+" - Buffer length :"+buffer.Length.ToString());
            }
            Thread.Sleep(1);
        }
    }
} //udpThread class

```

Ek 1e Sniffing tuşuna tıklanması durumunda çalışan kod bloğu

```

void sniffingButtonClick(object sender, EventArgs e)
{
    if(sniffButton.Text=="Start Sniffing"){
        _sniffing=true;
        sniffButton.Text="Stop Sniffing";
        snifferThread = new Thread(new ThreadStart(snifferThreadFunction));
        snifferThread.Start();
        bgw.RunWorkerAsync();
    } else {
        //Stop Sniffing
        sniffButton.Text="Start Sniffing";
        _sniffing=false;
    }
}

```

Ek 1f Paket koklayıcı iş parçacığına ait ana fonksiyon

```
private void snifferThreadFunction()
{
    try {
        try {
            var _socket = new Socket(AddressFamily.InterNetwork,
                                     SocketType.Raw,
                                     ProtocolType.IP);

            try {
                _socket.Bind(
                    new System.Net.IPEndPoint(System.Net.IPAddress.Parse(nicIP), 0)
                );

                _socket.SetSocketOption(SocketOptionLevel.IP,
                                         SocketOptionName.HeaderIncluded, 1);

                var _out = new byte[4];
                _socket.IOControl(unchecked((int)0x98000001),
                                 new byte[4]{1, 0, 0, 0}, _out);
                while (_sniffing) {
                    System.IAsyncResult ar;

                    ar = _socket.BeginReceive(_snifferBuffer,
                                              0,
                                              _snifferBufferSize,
                                              SocketFlags.None,
                                              new System.AsyncCallback(CallReceive),
                                              this);

                    while (_socket.Available == 0) {
                        Thread.Sleep(1);
                        if (!_sniffing) break;
                    }

                    if (!_sniffing) break;
                    int Size = _socket.EndReceive(ar);
                    ExtractBuffer();
                }
            } finally {
                if (_socket != null) {
                    _socket.Shutdown(SocketShutdown.Both);
                    _socket.Close();
                }
            }
        } finally {
            _sniffing=false;
        }
    } catch (System.Exception E) {
        System.Windows.Forms.MessageBox.Show("HATA:"+E.ToString());
    }
    _sniffing=false;
}
}
```

Ek 1g Yakalanan paketlerin DNS çözümünü yapan ve kuyruğa ekleyen kod bloğu

```
protected void ExtractBuffer()
{
    IPPacket IP = new IPPacket(ref _snifferBuffer);

    string SourceAddress = IP.SourceAddress.ToString();
    string DestinationAddress = IP.DestinationAddress.ToString();

    if (dnsResolveCheckbox.Checked) {
        string HostName = null;

        //Kaynak adres DNS sorgulama
        try {
            HostName = DNSCache[SourceAddress];
        } catch (Exception e) {
            ResultBox.Items.Add(e.Message);
        }

        if (HostName == null) {
            DNSCache[SourceAddress] = "";

            try {
                HostName = System.Net.Dns.GetHostEntry(SourceAddress).HostName;
            } catch (Exception e) {
                HostName = "";
                ResultBox.Items.Add(e.Message);
            }

            DNSCache[SourceAddress] = HostName;
        }
        if (HostName != SourceAddress) SourceAddress += " " + HostName;

        //Hedef adres DNS sorgulama
        HostName=null;
        try{
            HostName = DNSCache[DestinationAddress];
        } catch(Exception){

        }
        if (HostName == null) {
            DNSCache[DestinationAddress] = "";

            try {
                HostName = System.Net.Dns.GetHostEntry(SourceAddress).HostName;
            } catch {
                HostName = "";
            }

            DNSCache[DestinationAddress] = HostName;
        }
        if (HostName != DestinationAddress) DestinationAddress += " " + HostName;
    }

    _packetQueue.Enqueue(IP);
}
```

ÖZGEÇMİŞ

Doğum Tarihi	27.02.1982	
Doğum Yeri	İstanbul	
Lise	1996-2000	Eyüp Otakçılar Lisesi
Lisans	2001-2006	Kadir Has Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü (Burslu)
Yüksek Lisans	2006-2010	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Müh. Anabilim Dalı ve Programı

Çalıştığı Kurumlar

2007-2010	Nortel Netaş
2010-Devam Ediyor	Alcatel - Lucent Teletaş