

**KOCAELİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ  
ANABİLİM DALI**

**DOKTORA TEZİ**

**KABLOSUZ GERÇEK ZAMANLI UYGULAMALAR İÇİN GERİ  
BESLEMeye DAYALI YENİ BİR UDP YAKLAŞIMI: F-UDP**

**ONUR GÖK**

**KOCAELİ 2013**

**KOCAELİ ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRONİK ve HABERLEŞME MÜHENDİSLİĞİ**  
**ANABİLİM DALI**

**DOKTORA TEZİ**

**KABLOSUZ GERÇEK ZAMANLI UYGULAMALAR İÇİN GERİ  
BESLEMeye DAYALI YENİ BİR UDP YAKLAŞIMI: F-UDP**

**ONUR GÖK**


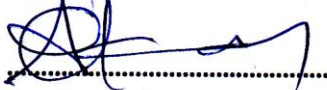



**Yrd.Doç.Dr.H.Engin DEMİRAY,**  
**Kocaeli Üniv.**

**Prof.Dr.Adnan KAVAK, Kocaeli**  
**Üniv.**

**Prof.Dr.Coşkun Sönmez, YTÜ**

**Doç.Dr.Mehmet YILDIRIM, Kocaeli**  
**Üniv.**

**Yrd.Doç.Dr.Mehmet YAKUT,**  
**Kocaeli Üniv..**

  
.....  
  
.....  
  
.....  
  
.....  
  
.....

**Tezin Savunulduğu Tarih: 25.04.2013**

## ÖNSÖZ VE TEŞEKKÜR

Bu çalışmada kablolu-kablosuz bilgisayar ağlarda gerçek zamanlı uygulamalar için problem olan paket kayıp oranının azaltılması için iletim katmanında geri beslemeli bir UDP metodu olarak F-UDP adlandırdığımız yeni bir protokol benzetimi önerilmiştir. İletim katmanında gerçek zamanlı uygulamalarda kullanılan UDP protokolü için geriye doğru hata düzeltme metodu kullanarak paket kayıp oranını düşürülmesi amaçlanmıştır. Ağ simülasyonu yardımı ile geliştirilen UDP temelli metod için paket kayıp oranı ve gerçek zamanlı iletim için önemli bir parametre olan gecikme değerleri ölçülerek sonuçlar gözlemlenmiştir.

Doktora eğitimim süresince değerli birikimlerini benimle paylaşan, tezimin her aşamasında sorunlarımı dinleyerek, çalışmalarına yön veren ve yoğun akademik yaşamında değerli zamanını her türlü problemimi çözmeye ayıran tez danışmanım saygı değer hocam Yrd. Doç. Dr. H. Engin Demiray'a, destek ve yardımlarında dolayı Prof. Dr. Adnan Kavak'a, Doc. Dr. Hacı Ali Mantar'a teşekkürlerimi sunarım.

Ayrıca maddi ve manevi desteklerini tüm hayatı boyunca esirgemeyen aileme şükranlarımı sunarım.

Nisan – 2013

Onur Gök

## İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR .....	i
İÇİNDEKİLER .....	ii
ŞEKİLLER DİZİNİ.....	iii
TABLolar DİZİNİ .....	iv
SİMGELEr DİZİNİ ve KISALTMALAR .....	v
ÖZET.....	vi
ABSTRACT .....	vii
GİRİŞ .....	1
1. TAŞIMA KATMANI VE KABLOSUZ YEREL ALAN AĞLARI.....	8
1.1. İnternet Ağları .....	8
1.2. TCP/IP Katmanları.....	9
1.2.1. Fiziksel katman.....	9
1.2.2. Ağ katmanı .....	10
1.2.3. Taşıma katmanı .....	11
1.2.4. Uygulama katmanı.....	11
1.3. Taşıma Katmanı Protokolü .....	12
1.4. Kullanıcı Datagram Protokolü .....	17
1.5. Kablosuz Yerel Alan Ağları.....	19
1.5.1. 802.11 mac .....	23
1.5.2. 802.11e protokolü.....	26
1.6. Kablosuz Ağlarda İletim Katmanı .....	28
1.6.1. Kablosuz ağlarda TCP.....	28
1.6.2. Kablosuz ağlarda UDP .....	35
1.6.3. Güvenilir UDP (Reliable UDP).....	37
2. GERÇEK ZAMANLI İLETİŞİM .....	39
2.1 Rtp.....	46
3. UDP TEMELLİ İLETİM KATMANI PROTOKOL TASARIMI.....	46
3.1. Geri Beslemeli UDP (F-UDP-Feedback base UDP).....	47
3.2. Simülasyon.....	61
3.3. Simulasyon Sonuçları .....	63
4. SONUÇLAR ve ÖNERİLER.....	69
KAYNAKLAR .....	71
KİŞİSEL YAYINLAR VE ESERLER .....	74
ÖZGEÇMİŞ .....	75

## ŞEKİLLER DİZİNİ

Şekil 1.1.	OSI modeli ile TCP/IP katmanları arasındaki ilişki .....	9
Şekil 1.2.	Taşıma katmanının diğer katmanlarla ilişkisi .....	14
Şekil 1.3.	TCP segmenti.....	16
Şekil 1.4.	UDP'nin çoğullanması.....	18
Şekil 1.5.	UDP datagramının formatı.....	18
Şekil 1.6.	Ağ ortamında Protokol genişlemesinin dinamik oluşumu.....	23
Şekil 1.7.	TCP bölümlenme .....	33
Şekil 1.8.	Protokol Şekli - RUDP (Reliable UDP) .....	38
Şekil 3.1.	UDP temelli protokol tasarımı(F-UDP).....	47
Şekil 3.2.	F-UDP temel istasyondaki sözde kodu .....	49
Şekil 3.3.	F-UDP alıcı tarafın sözde kodu .....	50
Şekil 3.4.	F-UDP göndericisi için durum diyagramı .....	51
Şekil 3.5.	F-UDP göndericisi durum diyagramı doğruluk analizi .....	52
Şekil 3.6.	F-UDP alıcısı için durum diyagramı.....	52
Şekil 3.7.	F-UDP alıcısı durum diyagramı doğruluk analizi.....	53
Şekil 3.8.	F-UDP için kablolu-kablosuz ağ paket akış zaman diyagramı.....	54
Şekil 3.9.	F-UDP için kablolu-kablosuz ağ paket akış zaman diyagramı.....	55
Şekil 3.10.	RTT ve sabit zaman için zaman diyagramı- hatalı ulaşımda tekrar istek .....	56
Şekil 3.11.	RTT ve sabit zaman tampon görünümü.....	57
Şekil 3.12.	Hatalı paket ulaştığında RTT ve sabit zaman tampon görünümü.....	58
Şekil 3.13.	Hatalı paket ve zaman aşımı için RTT ve sabit zaman için tampon.....	59
Şekil 3.14.	Geleneksel UDP için iletim zamanı sembolleri.....	60
Şekil 3.15.	Geleneksel UDP için iletim zamanı sembolleri.....	60
Şekil 3.16.	NS simülasyon dosya sistemi.....	62
Şekil 3.17.	F-UDP paket saydırma ile tamponlama için paket gecikmesi.....	64
Şekil 3.18.	F-UDP paket saydırma ile tamponlama için paket kaybı .....	64
Şekil 3.19.	F-UDP için tampon bekleme süresi RTT için paket gecikmesi.....	65
Şekil 3.20.	F-UDP için tampon bekleme süresi RTT paket kayıp oranı.....	65
Şekil 3.21.	F-UDP için tampon bekleme süresi sabit değer için paket gecikmesi.....	66
Şekil 3.22.	F-UDP için tampon bekleme süresi sabit değer için paket kayıp oranı .....	66
Şekil 3.23.	Test 1 için PSNR değerleri.....	68
Şekil 3.24.	Test 2 için PSNR değerleri.....	68

## **TABLolar DİZİNİ**

Tablo 1.1. Kablosuz yerel alan ađ standartları .....	20
Tablo 2.1. Gerçek zamanlı iletimde tek yön limitleri .....	44

## SİMGELER DİZİNİ VE KISALTMALAR

$T_{total}$	:	İletim için toplam geçen süre
$T_{wd}$	:	Kablolu ağda iletim süresi
$T_{wl}$	:	Kablosuz ağda iletim süresi
$T_{ftotal}$	:	F-UDP için toplam geçen süre
$T_{fwd}$	:	F-UDP için kablolu ağda iletim süresi
$T_{fwl}$	:	F-UDP için kablosuz ağda iletim süresi
$T_{nmb}$	:	F-UDP paketler numara yerleştirme işlemi için oluşan gecikme
$T_{feed}$	:	NACK kullanarak geriye doğru hata düzeltme işlemi gecikmesi
$T_{buffer}$	:	tampona paketi atma ve almak için gereken zaman
$V_{peak}$	:	Video parlaklık bit değeri

### Kısaltmalar

ACK	:	Acknowledgement (Geriye Dönük Olumlu Cevap)
AP	:	Kablosuz İşlem noktası(Access Point)
BS	:	Base Station (Temel İstasyon)
FEC	:	Forward Error Correction(İleriye Doğru Hata Düzeltme)
FH	:	Forwarding Host (Gönderici uç bilgisayar)
IP	:	Internet Protocol(İnternet Protokolü)
LAN	:	Local Area Network (Yerel Alan Ağı)
MH	:	Mobil Host (Mobil uç bilgisayar)
NACK	:	Negative Acknowledgement (Geriye Dönük Olumsuz Cevap)
PDU	:	Protocol Data Unit (Protokol Veri Birimi)
PSNR	:	Peak Signal-To-Noise Ratio(Doruk Sinyal Gürültü Oranı)
RLP	:	Radio Link Protokol (Radyo Bağlantı protokolü)
RTT	:	Round Trip Time (Dolanım Süresi)
RTP	:	Real Time Protocol (Gerçek zamanlı Aktarım Protokolü)
SEQ	:	Sequential (Sıralı)
TCP	:	Transmission Control Protocol (İletim Kontrol Protokolü)
UDP	:	User Datagram Protocol (Kullanıcı Datagram Protokolü)
WLAN	:	Wireless Local Area Network (Kablosuz Yerel Alan Ağı)

## **KABLOSUZ GERÇEK ZAMANLI UYGULAMLAR İÇİN GERİ BESLEMeye DAYALI YENİ BİR UDP YAKLAŞIMI: F-UDP**

### **ÖZET**

Gerçek zamanlı iletişiminde paket kaybı kalite için önemli bir parametredir. Gerçek zamanlı Aktarım Protokolü (RTP), IP ağları üzerinden gerçek zamanlı ses ve video sunmak için standart bir paketleme biçimidir ve RTP uygulamaları Kullanıcı Datagram Protokolü (UDP) üzerine inşa edilmiştir . Kablosuz ağlarda paket kayıp oranı genellikle kablolu ağlara göre daha yüksektir ve bu nedenle kablolu-kablosuz ağlarda gerçek zamanlı iletişim için önemli bir konudur. Uygulama katmanında ileri hata düzeltme yöntemleri kullanarak sorunu çözmek için bir kaç çalışma yapılmıştır. Çalışmamızda ise, geri hata düzeltme kullanarak paket kayıp oranı azaltmak için geri bildirim tabanlı adı verilen bir yöntem öneriyoruz(F-UDP).F-UDP' de, iletim iki taraf ayrılmıştır: kablolu ve kablosuz. Kablolu tarafta, geleneksel UDP iletim kullanılır. Kablosuz tarafında geriye hata düzeltme ise, negatif alındı, tamponlama ve numaralandırılmış UDP paketleri ile uyarlanarak kullanılmıştır. Kablosuz tarafta kayıp paketler için tampon ve tekrar iletim için akış kontrol mekanizması için kullanılmıştır. F-UDP paket kayıp oranı vadede UDP daha iyi performans olduğunu NS-2 simülasyon sonuçlarında F-UDP' nin paket kayıp oranını düşürdüğü gözlemlenmiştir.

**Anahtar Kelimeler:** Gerçek zamanlı iletim, iletim katman, UDP, WLAN.



## **A NEW FEEDBACK BASED UDP DESIGN FOR WIRELESS REAL TIME APPLICATIONS: FUDP**

### **ABSTRACT**

Packet loss in real-time communication is an important parameter for quality. The Real-time Transport Protocol (RTP) defines a standardized packet format for delivering audio and video over IP networks and the majority of the RTP implementations are built on the User Datagram Protocol (UDP). Packet loss rate in wireless networks usually higher than wired networks and therefore it is an important issue for real time communication in wired-cum-wireless networks. There have been a few studies which try to solve problem by using forward error correction methods at application layer. In our study, we propose a method so called Feedback based UDP (F-UDP) in which we try to decrease packet loss rate by using backward error correction. In F-UDP, the transmission is divided into two sides: wired and wireless. In wired side, traditional UDP transmission is used. In wireless side, backward error correction is adapted by using negative acknowledgement, buffering and numbered UDP packets. A buffer is used for loss packets and flow control mechanism generated for wireless side. The simulations results on NS-2 show that F-UDP outperform UDP in term of packet loss rate.

**Keywords:** Real time transmission, transmission layer, UDP, WLAN.

## **GİRİŞ**

Kablosuz iletişim teknolojisi, en basit tanımıyla, radyo sinyalleri kullanarak noktadan noktaya veya bir ağ yapısı şeklinde bağlantı sağlayan bir teknolojidir. Kablosuz Yerel Alan Ağları (Wireless Local Area Networks, WLAN), iki yönlü geniş bant veri iletişimi sağlayan, iletim ortamı olarak telsiz frekansı veya kızıl ötesi ışınları kullanan ve salon bina veya kampüs gibi sınırlı bir alanda çalışan iletim ağlarıdır. Kurulum kolaylığı ve hareket serbestliği gibi önemli avantajları vardır.

Kablosuz Yerel Alan Ağları havadan aygıtlar arasında radyo sinyalleri göndererek çalışır. Sinyaller 300 metreye kadar gönderilebilir ve metal olmayan duvarlardan ve engellerden geçebilir. Dizüstü bilgisayarlar, masaüstü bilgisayarlar ve sunucular Ethernet benzeri bir kart ile veya kablosuz sinyalleri alıp iletebilen kartlar ile kablosuz yerel ağ'a bağlanırlar. Bağlanırken kablosuz yerel ağ ile kablolu yerel ağ arasında köprü olarak görev yapacak bir Erişim Noktası'na (Access Point-AP) gereksinim duyarlar.

IEEE Kablosuz Yerel Alan Ağ standartlarının oluşmasında çalışmaları sonucunda tüm dünya tarafından kabul görmüş ve IEEE 802.11x (802.11, 802.11a, 802.11b, 802.11g) olarak yayımlanmıştır. Versiyon farklılıklarının çıkma nedeni şöyle açıklanabilir: Radyo iletişimde fiziksel bazı kurallar vardır. Gerekli standartlar oluşturulurken bu özellikler göz önüne alınmaktadır. Eğer yüksek hızda bir iletim gerçekleştirmek istenirse verinin tekrarlanmadan gönderebileceği alan düşük olmaktadır. İletim alanını arttırmak için güç tüketimini arttırabilir, ama bu durumda da cihazın pil ömrü azalmaktadır. İletim hızını arttırmanın diğer bir yolu da daha yüksek frekanslarda iletimi gerçekleştirmektir. Yukarıda bahsedilen iletim kuralları ve insanların ihtiyaçları baz alınarak IEEE tarafından çeşitli standartlar tanımlanmıştır.

Kablosuz ağların kablolu ağlarla iletişimi ortak protokolleri kullanmaları ile olabilmektedir, İnternet'in de temel standart ağ yapısı düşünüldüğünde, TCP/IP

internet protokol kümesi ortak dil olarak kullanılması kaçınılmazdır. TCP/IP protokol kümesi tasarlanırken kablolu ağ yapısı temel alındığında kablosuz ağın fiziksel yapısından meydana gelen problemler ortaya çıkmıştır. Bu problemler için TCP/IP protokol kümesi içerisinde ve kablosuz ağ standartları üzerinde değişimler ve gelişimler de kaçınılmaz hale gelmiş ve ağ performansını artırmak için farklı yapısal ve protokol mantığında değişikliklere neden olmuştur. TCP/IP'nin katmanlı mimarisi göz önüne alındığında, her katman için ayrı çalışmalar ve ayrı çözümler veya bileşik çözümler ortaya atılmıştır.

Çalışmaların olduğu katmanlardan biri TCP/IP'nin iletim katmanıdır. İletim katmanı, verilen servise göre iki tür protokol tipi mevcuttur, bunlar: TCP ve UDP.

Transmission Control Protocol (TCP), güvenilir, bağlantı yönelimli, çift yönlü, byte katarlı, ulaşım katmanı protokolüdür (Postel,1981). Uç kullanıcıların uygulamaları için uçtan uca akış ve tıkanıklık kontrol sağlayan bir protokoldür (Saltzer ve diğerleri 1984). İnternet trafiğinin büyük bir kısmı TCP oluşturur (Thompson ve diğerleri 1997). TCP, kablolu ağlar için tasarlanmıştır, TCP kablolu ağlarda çok verimli çalışırken, kablosuz ağlarda iyi performans verememiştir (Pentikousis, 2000). Kablosuz ağlardaki TCP uygulamalarındaki performans problemlerinin ana kaynağı, iletim hatalarıdır(Hassan ve diğeri, 2004). Paketlerdeki bazı hatalar, alt katmanlardaki FEC kodları tarafından düzeltilirken, daha fazla hata paketlerin bozulması olarak kabul edilir. Bozulan paketler TCP'nin eline geçmeden atılır. TCP, paket kayıplarını ağ tıkanıklığı olarak kabul ettiği için, tıkanıklık penceresini azaltarak tepki verir. Bu durumda, kablosuz iletim hataları, ağ tıkanıklığı ile ilişkilendirilir, böylece uygunsuz olarak tıkanıklık penceresinin küçülmesi, TCP uygulamaları için gereksiz kayba neden olur .

Ses, video ve resim gibi etkileşimli ve ağ tabanlı çoklu ortam uygulamaları internet ve kablosuz ağlarda yaygın olarak kullanılmaktadır. Çoğu internet tabanlı gerçek zamanlı çoklu ortam uygulamaları iletim katmanı protokolü olarak UDP kullanılmaktadır. TCP'nin kullanılmamasının nedeni ise TCP de tekrar gönderim sırasında ortaya çıkacak gecikmelerin bu tarz uygulamalarda istenmemesidir.

UDP (User Datagram Protocol), bağlantı kurulum işlemleri, tıkanıklık kontrolü, akış kontrolü ve tekrar iletim işlemlerini yapmayarak veri iletim süresini minimum protokol mekanizması ile en aza indiren iletim katmanı protokolüdür. UDP uçtan uca güvenli ve sıralı bir iletişim sağlamaz (Postel, 1980). UDP'nin bu özellikleri sayesinde gecikmeye karşı hassas olan gerçek zamanlı uygulamalarda UDP tercih edilmektedir. Bir UDP paketi başlık ve veriden oluşmaktadır. UDP' de isteğe bağlı olarak, yalnızca iletilen paketin sağlama bilgisi (checksum) ile bütünlük kontrolü yapılır. Sağlama bilgisinin içeriği UDP başlık ve veri kısmından elde edilir. UDP paketi alındığında sağlama bilgisi doğrulanmazsa, bunun anlamı paketin hatalı olduğu ve iletim katmanında yok edileceğidir.

Kablosuz ağlarda gerçek zamanlı uygulamalar için geleneksel ses ve video uygulamalarında, iletim katmanına hatalı olarak ulaşan bir paketin atılması yerine uygulamaya gönderilmesi tercih edilir. Bunun nedeni, çoğu zaman hataların bit hatasından kaynaklanması ve bu hatalarında uygulamanın kullanacağı bir hata düzeltme mekanizması ile giderilebilmesidir. Kablosuz ağlarda paket hatalarının çoğu ortam ve kanal hatalarından kaynaklanan tek bit hatalarıdır. Yapılan araştırmalara göre, sağlama bilgisi hataları nedeniyle, alıcı tarafındaki iletim katmanında paketlerin yaklaşık olarak %85'i atılmaktadır (Lam ve diğerleri, 2004). Hatalı paketlerin uygulama katmanına iletilmesinin bir yolu, UDP' de bütünlük kontrolünün kullanılmamasıdır. Bu durumda ise uygulamaya özel başlıkların doğrulanmadan geçirilmesi gibi önemli bir problemle karşılaşılabilir.

Kablosuz ağlarda gerçek zamanlı uygulamalar için TCP ve UDP yeterli derecede çözüm olamamışlardır. Bunun temel sebebi, kablosuz ağlardaki paket bozulmalarının kablolu ağa göre göz ardı edilmeyecek derecede fazla olmasıdır. Hem TCP tarafında hem UDP tarafında problemin çözümlerini arayan çalışmalar yapılmıştır.

Kablosuz ağlarda TCP için çok farklı metotlar denenmiş, TCP'nin farklı versiyonları ortaya çıkmıştır. Kablosuz TCP metotlarından bir tanesi de TCP bağlantısını kablolu ve kablosuz için bölümlendirmek olmuştur (Larzon ve diğerleri, 1999).

Kablosuz UDP' de ise yapılan çalışmalar, ileri doğru hata düzeltme çalışmalarıdır. Hatalı ulaşan paketler uygulama katmanına hatalı olmalarına rağmen, üst katmandaki uygulama ile düzeltimi yapılmaya çalışılmaktadır. UDP-Lite, UDP-Liter, C-UDP protokol çalışmalarında, gelen paketteki hata, alt katmanlarda göz ardı edilerek, uygulama katmanına iletilir, uygulama katmanında hata düzeltme algoritmaları yardımı ile hata düzeltilmeye çalışılır. Hata düzeltme işlemi için, gönderilen paketlerin hatalı da olsa alıcıya ulaşması gerekmektedir. Ulaşmayan paketlerin hata düzeltilmesi olamaması nedeni ile fiziksel ortam nedeni ile ortaya çıkan kayıplara bir etkileri olmayan çalışmalarıdır. İleriye dönük hata düzeltme metotlarının, göndericiden alıcıya ulaşmayan veri paketleri için herhangi çözümü yoktur.

Kablosuz Yerel Alan ağlarda fiziksel ortam nedeni ile oluşan paket kayıp oranının, kablolu iletme göre daha fazla olmaktadır. Günümüzde en çok kullanılan TCP/IP internet protokolleri ailesi içerisinde birçok çalışma yapılmış, paket kayıp oranından kaynaklanan iletim gecikmeleri en az seviyeye indirilmeye çalışılmıştır.

TCP/IP iletim katmanında, uçtan uca çözüm için TCP ve UDP için farklı çözüm önerileri yapmışlardır. TCP güvenilirliği sebebi ile iletimdeki paket kayıpları tekrar iletim ile düzeltmiş, bu düzeltmede bekleme süresini azaltmak için farklı önerilerde bulunulmuştur.

Gerçek zamanlı uygulamalar için iletim katmanında UDP protokolü, tasarımı gereği TCP'ye göre tercih edilmektedir. Bunun sebebi, UDP protokolünde gönderilen paketler için göndericiye paketlerin iletilmesine dair bildirim yapılmamasıdır. Bu sebeple peş peşe paketler gönderilebilmektedir. Kablolu ağlardaki paket kayıp oranının göz ardı edilecek derecede küçük olmasından dolayı, UDP protokolü gerçek zamanlı uygulamalar için ideal bir çözüm olmuştur. TCP/IP protokol ailesinin ilk olarak kablolu ağlar için tasarlanmış olmasından dolayı, kablosuz ağlardaki gerçek zamanlı uygulamalarda paket kayıp oranından kaynaklanan problemler yaşanabilmektedir.

İnternet tabanlı gerçek zamanlı uygulamalar için genellikle, Kullanıcı Veri iletim protokolü (UDP) ulaşım katmanı protokolü olarak kullanılır. İletim katmanının

protokolleri olan iletim kontrol protokolü (TCP) veya UDP' de alıcıyı gönderilen paketlerin kayıp oranı, kablosuz yerel alan uygulamalarında önemli bir sorundur. UDP'de sorunu çözmek için önerilmiş olan uygulama katmanında ileri hata düzeltme yöntemleri bulunur. TCP için de yine bağlantı bölümlene yöntemleri, akış kontrol, geriye dönük hata düzeltme yöntemleri kullanılarak paket kaybı yok edilmiş ve bekleme süreleri azaltılır. Önerilen ağ tasarımında, gerçek zamanlı uygulamalar için çok yaygın olarak kullanılan UDP protokolünün, kablosuz ağlardaki paket kayıpları ve bozulması durumunda geriye doğru hata düzeltme yapılması öngörülür.

Geleneksel UDP içinde kayıp olan paketlerin tekrar istenmesi, protokol dizaynı nedeniyle mümkün olamamaktadır. Bu nedenle, protokolün işleyişi ve paket başlığına eklenen bilgilerle UDP protokolünün kablosuz TCP metotlarından bağlantı bölme benzetimi sağlanarak güvenilirliği yükseltilmeye çalışılmış ve paket bekleme süresi azaltılması amaçlanmıştır. UDP için paket iletimi ortamı kablolu ve kablosuz olarak ikiye ayrılmış, kablolu tarafta geleneksel UDP çalışmış, kablosuz tarafta geriye dönük hata düzeltme yöntemi ile paket kayıp oranı düşürülmeye çalışılmıştır. Protokol tasarlanırken, paket kayıp oranının düşürülmesinin hedef olmasının yanında gerçek zamanlı uygulamalar için kullanılması dikkate alınarak bekleme süresi de geleneksel UDP ile yakın değerlerde olması göz önünde bulundurulmuştur.

Önerilen çalışmada kablosuz ağlarda kullanılan TCP'deki gibi ağı bölme işlemini UDP protokolü için kullanarak kablolu ve kablosuz ağı ikiye ayrılacak ve kablolu ağ tarafında geleneksel UDP bağlantısı kullanılacak. Kablosuz tarafta, tezde önerilen geri beslemeli UDP mekanizması kullanarak geriye hata düzeltme işlemi yaparak hatalı ulaşan paketler veya ulaşılmayan paketler tekrar gönderimle hata oranı düşürülmeye çalışılacaktır. Hatalı ulaşan veya ulaşmayan paketler için tüm paketlerin için gelindi bilgisini göndermek yerine sadece hatalı ulaşan veya ulaşmayan paketler için istekte bulunulacaktır. Kablolu ağ tarafında geri bildirim olmayacaktır. Kablosuz tarafta hata düzeltme yapılacağı için, hata için RTT(dolanım süresi round trip time) kısılacaktır. UDP kullanıldığı için ağda tekrarlı paketler olmayacaktır.

Gerçek zamanlı uygulamalarda TCP kullanılması iyi sonuç vermemesinin başlıca nedenleri:

- Uçtan uca bağlantı: uçtan uca bağlantıda paketlerin ulaşmaması durumunda tekrar gönderilmesi gerekecektir. Bu da ulaşmayan paketler için ve onay için bekleme süresi demektir.
- Güvenilir bağlantı: Her paket için işlem zamanı ve yayılım zamanı onay mekanizması sebebi ile 2 katına çıkar. Paketin ulaşmaması veya hatalı ulaşması durumunda, paket uygulama katmanına çıkmadan atılır ve paketin doğru gelmesi beklenir.

Kablosuz ağlarda durum TCP için yine aynı olacaktır. Aynı zamanda, fiziksel nedenlerle ulaşamayan paket, TCP tarafından tıkanıklık olduğu farz edilerek pencere boyutu küçülecek bu sebeple gönderilen paket sayısı düşürülerek, iletimi azaltacaktır. Kablosuz ağlarda fiziksel ortamdan kaynaklanan veri kaybı, kablolu ağlardan çok daha yüksektir (başarım kablolu ağlarda %99 civarı, kablosuz ağlarda yaklaşık olarak %60).Kayıp sebebi ile TCP pencere boyutunu küçültür. Peş peşe gönderdiği paket sayısı azalır. Onay mekanizması kullanır. Bu tamponlama demektir, onayın gecikmesi veya gelmemesi durumunda paket tekrar gönderilir.

Kablosuz taraf için TCP snooping veya splitting kullanılması durumunda, kablolu taraftaki gecikme zamanı azalır. RTT değeri azalır fakat kablosuz ağdan kaynaklanan sorun giderilemez: paketi iletim oranı düşüktür, TCP problemi tıkanıklık olarak düşünür.

SACK ve TCP kablosuz ağ tarafında UDP gibi bir performans verebilir. R-UDP ve TCP & SACK onay mekanizması bakımından birbirine benzemektedir. İkisinde de sadece ulaşmayan veriler için onay mekanizması çalışır fakat F-UDP kullanmak, bağlantı gerektirmez, paket boyu küçülür.

Çalışmanın getireceği avantaj Kablolu ağ tarafında geri bildirim olmayacaktır. Kablosuz tarafta hata düzeltme yapılacağı için, hata için RTT kısalmayacaktır. UDP kullanıldığı için ağda tekrarlı paketler olmayacaktır.

Bu tez Őu b6l6mlerden oluŐur:

- TaŐıma katmanı ve kablosuz yerel alan aĐları
- Gerçek zamanlı iletiŐim
- Bilgisayar aĐlarında gerçek zamanlı uygulamalar
- Udp temelli iletim katmanı protokol tasarımı ve benzetimi
- Sonuç ve yorumlar



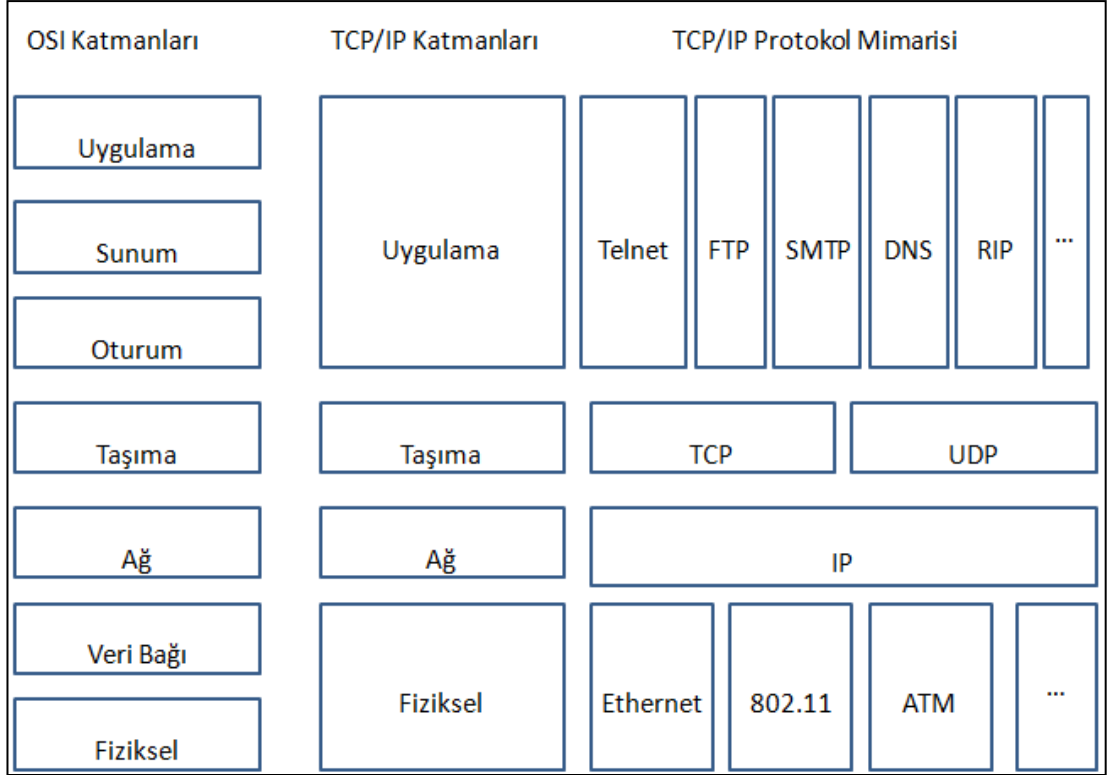
## 1. TAŞIMA KATMANI VE KABLOSUZ YEREL ALAN AĞLARI

### 1.1. İnternet Ağları

TCP/IP ilk olarak, 1970’li yılların başında Amerikan Savunma Bakanlığı’na bağlı İleri Araştırma Ajansı (Advanced Research Projects Agency, ARPA)’nın savunma amaçlı projelere destek vermek üzere paket anahtarlamalı ağ deneylerinde kullanılmaya başlanmıştır. Bu proje Amerika’da bulunan bilgisayarların bir felaket halinde bile ayakta kalabilmesini sağlamak ve iletişimini devam ettirmesi için gerekli unsurları kapsayacak şekilde tasarlanmıştır. Bu nedenle ARPA geliştirdiği paket anahtarlamalı deneysel çalışmaları ilk olarak Geniş Alan Ağları (Wide Area Network, WAN)’na uygulamaya başladı daha sonra geliştirilen bu sistemi Yerel Alan Ağ (Local Area Network, LAN) sistemlerine göre de adapte etmeye başladı.

TCP/IP, OSI standartlarını esas almak üzere toplam dört katmandan oluşmaktadır. OSI standardında bulunan yedi katman TCP/IP mimarisinde dört katmanda birleştirilmiştir ve benzer işlevlere sahiptir. Ancak TCP/IP protokolü kendi katmanları içinde birbirinden farklı görevlere sahip protokoller içeren bir protokol kümesidir.

Şekil 1.1.’de görülen mimariden anlaşılacağı gibi, OSI başvuru modeli katmanlarının TCP/IP başvuru modelinde tam bir karşılığı yoktur. TCP/IP’de olmayan sunum ve oturum katmanlarının görevleri, başka katmanlar tarafından yapılır. Başvuru modellerinin gelişimi itibarıyla OSI modelinde önce katman yapısı ardından ilgili protokoller tanımlanmıştır; fakat TCP/IP’de, Taşıma Katmanı Protokolü (Transmission Control Protocol, TCP) ve İnternet Protokolü başta olmak üzere, geliştirilen protokollerin başvuru modeline yerleştirilmesi ile aile modeli tamamlanmıştır. TCP/IP uygulama katmanı, zaman içerisinde geliştirilen çeşitli protokollere zenginleşmiş ve sistemin yaygınlaşmasına katkıda bulunmuştur.



Şekil 1.1. OSI modeli ile TCP/IP katmanları arasındaki ilişki

## 1.2. TCP/IP Katmanları

### 1.2.1. Fiziksel katman

TCP/IP modeli, host'un herhangi bir protokolü kullanıp, IP paketlerini gönderebilmesi gerektiğinin dışında, bu katmanda ne olduğu hakkında fazla bir şey belirtmemektedir. Çeşitli ağ protokolleri (ethernet, token ring, token bus, FDDI) kullanılarak veriler fiziksel şebeke üzerinden hedeflerine gönderilmektedir.

TCP/IP'nin tasarımı fiziksel katmanın fonksiyonunu kullanıcıdan gizler. Bu tasarım yeni ağ teknolojileri bulununca (ATM, Frame Relay) TCP/IP yığınının daha yüksek seviyelerinin yeniden yazılması ihtiyacını azaltır.

TCP/IP'nin kuvvetli yanlarından biri de ağdaki her bilgisayarı tek olarak tanıyan adresleme yöntemidir. IP adresi, datagramın iletileceği fiziksel ağda hangi fiziksel adres uygunsa o adrese dönüştürülür.

Fiziksel katman, yüksek seviyeli protokollerden farklı olarak, ağ katmanı protokolleri, paket yapısı, maksimum çerçeve büyüklüğü, kullanılan fiziksel adres yöntemi gibi ayrıntılar ile ilgilenir.

Fiziksel katman, internet başvuru modelinde en alt katmandır. Bu katman, bilgisayarın ağa bağlı cihaz ve bilgisayarlara veri gönderebilmesi için kullandığı protokolleri içerir. Bu katmandaki protokoller üç farklı fonksiyonu gerçekleştirir. Bunlar:

- Bu protokoller bir çerçeve iletmek için ağın nasıl kullanılacağını, fiziksel bağlantı üzerinden gönderilecek veri biriminin hangisi olduğunu tanımlar.
- Bilgisayar ve fiziksel ağ arasındaki veri alışverişini yapar.
- Aynı şebekedeki iki cihaz arasında veri taşır. Yerel ağda veri götürmek için ağ katmanı protokolleri, düğümlerin ağdaki fiziksel adreslerini (MAC adresi) kullanır. Bir fiziksel adres bir bilgisayarın ağ bağdaştırma kartında veya başka bir aygıtta saklanır ve değeri üreticisi tarafından bağdaştırıcı kartına donanımsal olarak kodlanmıştır.

### **1.2.2. Ağ katmanı**

Fiziksel katmanının üzerindeki katmana ağ katmanı denir. Bu katman mesajları ağlar arasında yönlendirmekten sorumludur. Mesajları ağlar arasında yönlendirmekten sorumlu olan iki tip aygıt vardır. İlk aygıt, iki ağ bağdaştırma kartına sahip olan “geçityolu (gateway)” denir. Bu aygıt, bir ağ bağdaştırma kartı üzerinden gelen ağ paketlerini kabul eder ve bu paketleri ikinci bir ağ bağdaştırma kartı aracılığıyla farklı bir ağa yönlendirir. İkinci aygıt, paketleri bir ağdan farklı bir ağa geçiren ve bu işe özel olarak adanmış olan “yönlendirici (router)”dir. Bu iki aygıtın paketleri ileme yetenekleri arasında belirgin farkları vardır.

İnternet katmanı protokolleri, bir datagram ağ hizmeti sağlar. Datagramlar başlık, veri ve kuyruk içeren “bilgi paketleri”dir. Başlık, hedef adres gibi ağın datagramı yönlendirmek için ihtiyaç duyduğu bilgiyi içerir. Bir başlık, kaynak adresi ve güvenlik etiketleri gibi başka bilgileri de içerebilir. Kuyruklar tipik olarak, verinin iletimde değişmemesini garantileyen bir kontrol toplamı (checksum) değeri içerir.

Datagram hizmetlerini kullanan haberleşme varlıkları (bilgisayarlar, işletim sistemleri, programlar, süreçler ve insanlar), iletilecek her mesaj için (kontrol bilgisini kullanarak) hedef adresi ve veriyi tanımlamalıdır. İnternet katmanı protokolleri mesajı, bir datagram içine paketler ve dışarı gönderir.

IP datagramı herhangi bir oturum veya bağlantı kavramını desteklemez. Mesajın bir kez alındığı veya gönderildiği durumda, haberleşilen varlıkla ilgili bellek tutulmaz. Eğer böyle bir bellek gerekirse ulaştırma katmanındaki protokoller bunu sağlar. Datagram hizmetlerinde verinin hatalara karşı kontrolü ve tekrar iletme isteği minimaldir veya yoktur. Eğer datagram hizmeti, iletim sırasında datagramın kontrol toplamı değerini kullanarak bir iletim hatası sezerse; basitçe, üst seviye varlığa haber vermeden datagramı ihmal eder.

### **1.2.3. Taşıma katmanı**

Ağ katmanının üzerindeki katman ulaştırma katmanıdır. Bu katman uçtan-uca veri bütünlüğü sağlamaktan sorumludur ve iki yönlü güvenilir iletişim gerçekleştirmek isteyen varlıklara haberleşme hizmeti sağlar.

Bilinen iletme ve alma fonksiyonlarına ek olarak, taşıma katmanı aç (open) ve kapa (close) komutlarını bir bağlantı başlatmak ve sonlandırmak için kullanır. Bu katman karakter akımı olarak iletilecek veriyi kabul eder ve bilgiyi alıcıya akım olarak gönderir.

Bağlantı, taşıma katmanının bir durumu olup, alıcı bilgisayar tarafından aç komutunun kabulü ile aynı bilgisayar tarafından kapa komutunun kabulü arasında geçen süre arasındaki zaman ile sınırlıdır.

Bu katmanda bağlantı yönelimli TCP protokolü ve bağlantısız UDP protokolü kullanılarak işlemler yapılmaktadır.

### **1.2.4. Uygulama katmanı**

TCP/IP modelinin en üst katmanı uygulama katmanıdır. Kullanıcı uygulamalarının ağ üzerinde haberleşmek için kullandığı hizmetleri sağlar ve kullanıcı ağ erişim süreçlerinin bulunduğu katmandır. Bu süreçler kullanıcıların

doğrudan etkileştiđi uygulamaları içerdiđi gibi, kullanıcıların haberi olmadıđı süreçleri de içerir.

Uygulama katmanı, taşıma katmanı protokollerinin veriyi iletmek için kullandığı tüm uygulama protokollerini içerir. Kullanıcı verisini işleyen diđer süreçler, veri şifreleme, şifre çözme, sıkıştırma, sıkıştırılmış veriyi açma da uygulama katmanında bulunur.

Uygulama katmanı birlikte çalışan uygulamalar arasındaki oturumları yönetir. TCP/IP protokol hiyerarşisinde, oturumlar ayrı bir katman olarak tanımlanamazlar ve bu fonksiyonlar ulaştırma katmanı tarafından gerçekleştirilir. Oturum terimini kullanmak yerine TCP/IP, birlikte çalışan uygulamaların haberleştikleri yolu tanımlamak için “soket ve port” terimlerini kullanır.

Uygulama katmanındaki uygulama protokollerinin çođu kullanıcı hizmetleri sağlar ve yeni kullanıcı hizmetleri sıkça eklenmektedir. Birlikte çalışan uygulamaların veri alış-verişini sağlamaları için, verinin temsil biçimi üzerinde anlaşmaları gerekir.

### **1.3. Taşıma Katmanı Protokolü**

TCP yedi-katmanlı OSI modelinin aktarım katmanında yer alır (İnternet katmanları oturum ve sunum katmanlarını içermez). TCP kullanıcı veri transferini desteklemek için yönlendiricilere yüklenmemiştir. TCP, host bilgisayarda veya kullanıcı verisinin uçtan-uca entegrasyonu ile görevli bir cihazda bulunur. Eğer TCP, yönlendiricide çalıştırılırsa; ağ yönetimi, yönlendiriciyle terminal oturumu, vs. aktiviteleri desteklemek için çalıştırılır (birkaç istisna dışında).

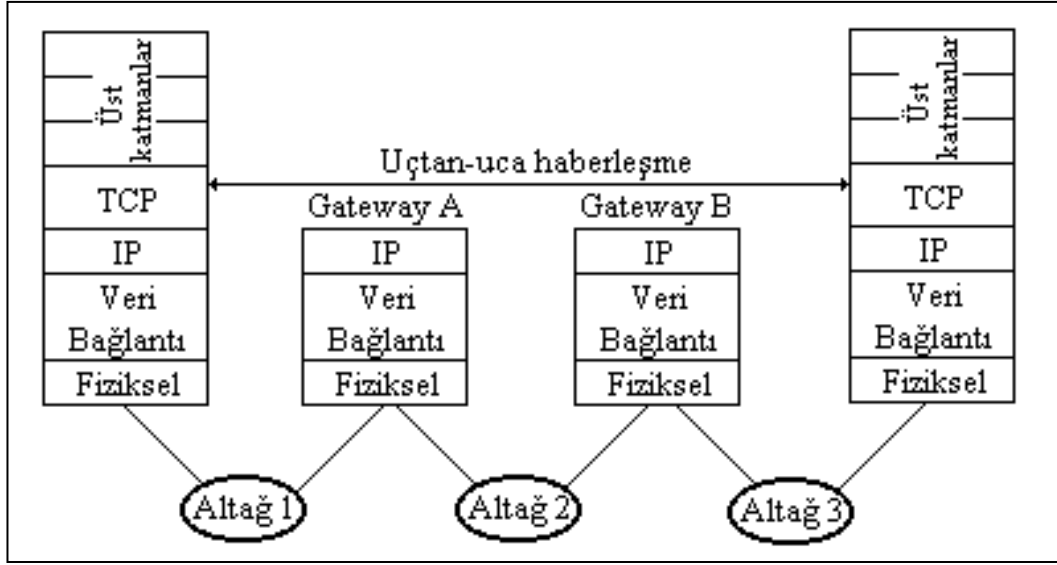
FTP, SMTP gibi uygulama protokolleri TCP'nin servislerinden yararlanırlar. Çođu TCP fonksiyonu (akış kontrolü, güvenilirlik ve sıralama gibi) bir uygulama programı ile başarılabilir. Ancak, her bir uygulama programı için bu fonksiyonları tekrar tekrar kodlamak hoş değildir. Tercih edilen yaklaşım geniş kullanım alanına sahip fonksiyonları sağlayan genelleştirilmiş bir yazılım geliştirmektir.

TCP üst katmanlara aşağıdaki servisleri sağlar:

- Bağlantı-yönlendirmeli veri yönetimi,
- Güvenilir veri transferi,
- Nehir-yönlendirmeli (bayt-bayt) veri transferi,
- Push fonksiyonları,
- Yeniden-sıralama (resequencing),
- Akış kontrolü (kayan pencereler),
- Çoğullama,
- Tam-duplex iletim,
- Öncelik ve güvenlik,
- Hoş bağlantı kapatma.

TCP bağlantı-yönlendirmeli bir protokoldür. Yani TCP, modülüne giren veya çıkan her bir “kullanıcı verisi nehir-akışı” ile ilgili durum ve konum bilgilerini sağlar. TCP aynı zamanda bir ağ veya çoklu ağlar boyunca yerleşmiş bir alıcı kullanıcı uygulaması ile (veya diğer üst katman protokolü) uçtan-uca veri transferi yapılmasından sorumludur. Şekil 1.2.’ de TCP’nin, veriyi üç ağ boyunca iki host arasında ilemesi gösterilmiştir. TCP iletim yaparken sıra numaraları ve pozitif onay (acknowledgment, ACK)’lar kullanır.

İletilen her bir bayt için bir sıra numarası atanır. Alıcı TCP modülü bir toplamsal-hata rutini kullanarak verinin iletim boyunca bir hasara uğrayıp uğramadığını kontrol eder. Eğer veri kabul edilebilir ise TCP, gönderici TCP modülüne bir pozitif onay gönderir. Eğer veri hasarlı ise, alıcı TCP veriyi yok eder ve bir sıra numarası kullanarak, gönderici TCP’ye sorun hakkında bilgi gönderir. TCP zamanlayıcıları tedavi ölçümleri yapmadan önce zaman kaymasının aşırı olmadığından emin olurlar. Tedavi ölçümleri, alıcı siteye onay (pozitif onay anlamında kullanılacaktır) gönderilerek veya veriyi gönderici siteye yeniden göndererek yapılır.



Şekil 1.2. Taşıma katmanının diğer katmanlarla ilişkisi

TCP, veriyi bir üst katman protokolünden nehir-yönlendirmeli biçimde alır. Nehir-yönlendirmeli protokoller ayrık karakterler (blok, çerçeve veya datagram değil) göndermek üzere tasarlanmamışlardır. Baytlar bir üst katman protokolünden nehir temelli, yani bayt-bayt gönderilir. Baytlar TCP katmanına varınca, TCP segmentleri olarak gruplaşırlar. Bu segmentler daha sonra diğer varışa iletilmek üzere IP'ye (veya başka bir alt katman protokolüne) geçirilir. Segment uzunluğuna TCP karar verir. Ancak bir sistem geliştiricisi, TCP'nin bu kararı nasıl vereceğine karar verebilir.

TCP ayrıca ikilenmiş veri kontrolü yapar. Eğer gönderici TCP veriyi tekrar yollarsa, alıcı TCP tüm ikilenmiş gelen veriyi yok eder. Örneğin; alıcı TCP onay trafiğini belli bir zamanda gerçekleştirmezse, gönderici TCP veriyi yeniden gönderir ve veri ikilenmiş olur.

TCP, onaylar için sıra numaraları kullanır. TCP bu sıra numaralarını aynı zamanda, segmentlerin son varışa sırası ile varıp varmadıklarını kontrol etmek üzere, segmentleri yeniden sıralamada kullanır. TCP bağlantısız bir sistemin üzerinde yer aldığı için ki bu sistem internet içerisinde dinamik, çoklu rotalar kullanılabilir, internette ikilenmiş datagramların oluşması muhtemeldir. TCP ikilenmiş datagramlar içerisinde taşınmış, ikilenmiş segmentleri yok eder.

TCP her bir oktete sıra numarası verir. Daha sonra iletildiği bu oktetlere karşılık onay bekler. Eğer belirli aralıklarla beklenen onayı almazsa, onay almadığı kısımları yeniden varış host'a iletir. TCP olumsuz bir geri bildirim mekanizması kullanmaz.

Alıcı TCP modülü gönderici verisi üzerinde akış kontrolü yapabilir. Böylece tampon overrun ve alıcı cihazın doyması (saturation) gibi sorunlar engellenir.

Akış kontrolü, göndericiye bir "pencere" değeri verilmesine dayanır. Gönderici bu pencere ile belirlenmiş sayıda bayt iletebilir, pencere kapanınca gönderici veri göndermeyi durdurmalıdır.

TCP'nin bir özelliği de, host cihazı üzerindeki çoklu kullanıcı oturumlarını çoğullayabilmesidir. Çoğullama, TCP ve IP modüllerindeki portlar ve soketler için basit isimlendirme anlaşmaları kullanılarak gerçekleştirilir. TCP, iki TCP varlığı arasında tam-duplex iletim sağlar. Böylece bir dönüş işareti beklemezsizin (half-duplex'te gereklidir) eşzamanlı iki-yönlü iletim yapılıır.

TCP, kullanıcının bağlantı için güvenlik ve öncelik seviyeleri belirleyebilmesine olanak tanır. Bu iki özellik, tüm TCP ürünlerinde bulunmayabilir, ancak TCP DOD standardında tanımlanmışlardır. TCP iki kullanıcı arasında hoş close sağlar. Hoş close bağlantı koparılmadan önce tüm trafiğin onaylarının oluşturulduğundan emin olunmasını sağlar.

TCP'nin her bir bağlantı trafiğini sağlamak için tek bir yöntemi vardır. Diğer birçok protokolün tersine, belirgin bir negatif onaya (Negative Acknowledgement, NACK) sahip değildir; ancak, ileten varlığın pozitif onayını (ACK) almadığı veriler için bir timeout ve yeniden-iletim (retransmit) verisi yayınlamasına dayanır. ACK'yı geri gönderir.

İki TCP modülü arasında değiştirilen PDU'lara "segment" denir. Şekil 1.3'te bir segmentin formatı verilmiştir.

Segment, başlık ve veri olmak üzere iki parçaya ayrılır. Şekil 1.3'de gösterildiği gibi veri, başlığın ardından gelir. Segmentin ilk iki alanı kaynak port ve varış



portudur. Bu 16-bit alan, TCP bağlantısını kullanarak üst-katman uygulama programlarını tanımlamada kullanılır.

Sıra numarası (Sequence number, SEQ) olarak etiketlenmiş alanın değeri, ileten modülün bayt-nehrinin yerini belirtir. TCP üst katmandan aldığı veriyi segmentlere böler. Bu segmentlerin her biri genellikle tek bir IP paketi içinde taşınır. TCP her bir segmente bir numara verir. Amaç, ağlar üzerinde dolaşan bu segmentlerin hedefe varış sıralarının karışması durumunda hedef host'ta çalışan TCP protokolünün bunları tekrar uygun şekilde birleştirip, üst katmana sunabilmesinin sağlamaktır (segment boyları sabit değildir).

TCP, karşı TCP ile bağlantıyı ilk kurduğunda, ilk gönderdiği segmente bir numara verir. Bu numaraya başlangıç gönderi sırası (Initial Send Sequence, ISS) denir. Sıra numarası 0 ile 231 değeri arasında olabilmektedir.

Kaynak Portu (16 bit)				Varış Portu (16 bit)				
Sıra Numarası (32 bit)								
ACK Numarası (32 bit)								
Veri Offset (4 bit)	Rezerve (6 bit)	U R G	A C K	P S H	R S T	S Y N	F I N	Pencere (16 bit)
Checksum (16 bit)				Acil İşaretçisi				
Opsiyonlar (Değişken)				Dolgu				
Veri (Değişken)								
...								

Şekil 1.3. TCP segmenti

TCP, verideki baytları gruplayarak segmentleri oluşturur ve her bir segment ayrı bir numara ile numaralandırılır. Bir segment, bir numara aldığı anda bu segment numarasını içinde barındırdığı ilk oktete verir. İçinde barındırdığı diğer oktetlere ise bu numaraların artanlarını verir. Bu segmentten sonra gelen segmentin alacağı numara, bir önceki segmentin içindeki en son oktetin aldığı numaranın bir fazlası olacaktır. Bu sıra numaraları segment başlığı içinde taşınır.

ACK numarasına bir değer atanarak önceden alınan verilerin onaylanması sağlanır. Bu alandaki değer, ileticiden gelmesi beklenen, bir sonraki baytın sıra numarası değerini belirtir. Bu numara beklenen oktet için set edildiğinden, dahili bir onay kapasitesi sağlar. Şöyle ki, bu değer bu numaraya kadar olan oktetleri ve bu numaralı oktet de onaylar.

Veri offset alanı, TCP başlığını oluşturan, 32-bit sıralı kelimelerin sayısını belirtir. Bu alan veri alanının nerede başladığının tespitinde kullanılır.

#### **1.4. Kullanıcı Datagram Protokolü**

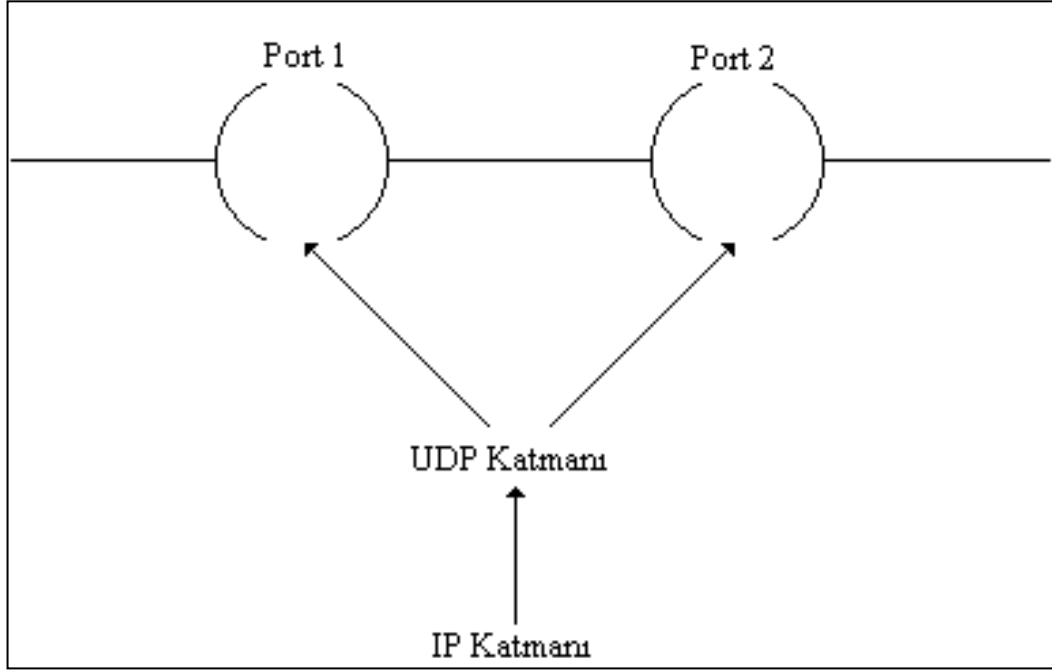
Bağlantısız protokol güvenilirlik ve akış-kontrol mekanizmaları sağlamaz, aynı zamanda hiçbir hata bulma prosedürü de yoktur. UDP, işletim sistemi her bir aktif UDP soketi ile ilgili bilgileri sağlamak zorunda olmasına rağmen, bağlantısız bir protokol olarak sınıflandırılır. UDP'nin daha iyi bir tanımı bağlantı-yönlendirmelidir denerek yapılabilir, ancak UDP normal olarak bağlantı-yönlendirmeli protokollerde geniş olarak kullanılan konum yönetim işlemlerini yürütmez. UDP bazen TCP'nin tüm servislerinin gerekli olmadığı yerlerde kullanılır. Örneğin; önemsiz FTP (Trivial FTP, TFTP), basit ağ yönetim protokolü (SNMP) ve RPC (Remote Procedure Call) UDP'yi kullanır.

UDP, IP'ye basit bir uygulama arabirimi olarak hizmet eder. Güvenilirlik, akış-kontrol veya hata-bulma ölçümleri olmadığından; prensip olarak, IP'nin alışverişi ve uygulamaların trafiği için bir port multiplexer/demultiplexer gibi hizmet eder. Şekil 1.4'de UDP'nin IP'den gelen datagramları nasıl kabul ettiği gösterilmiştir. UDP, datagramları doğru üst katman uygulamalarına yönlendirmek için "port" kavramını kullanır. UDP datagramı, bir varış port numarası ve bir kaynak port numarası içerir. UDP modülü varış numarasını trafiği doğru alıcıya teslim edebilmek için kullanır.

Bu protokolü açıklamak için en iyi yol, mesajını ve mesajındaki alanları incelemektir. Şekil 1.5.'deki gibi format oldukça basittir ve aşağıdaki alanları içerir:

- Kaynak portu: Bu değer gönderici uygulama işleminin portunu tanıtır. Bu alan opsiyoneldir ve eğer kullanılmazsa, buraya "0" değeri yerleştirilir.
- Varış portu: Bu değer varış host cihazındaki alıcı işlemi tanıtır.
- Uzunluk: Bu değer, başlık ve veri de içinde olmak üzere, kullanıcı datagramının uzunluğunu gösterir.
- Toplamsal-hata (Checksum): Bu opsiyonel değer; sözde-IP başlığı, UDP başlığı ve verinin 1'lerinin toplamının tümleyeninin, 16-bit 1'e tümlemesini

içerir. UDP aynı zamanda herhangi bir doldurma (padding) için de bir toplamsal-hata sağlar.



Şekil 1.4. UDP'nin çoğullanması

Sözde-başlık (aynı zamanda TCP'de de kullanılır) UDP veri biriminin doğru varış adresine varmasını sağlar. Böylece, sözde-başlık IP adresleri içerir ve toplamsal-hata hesabına katılır. Son varış sözde-başlığa (UDP veri biriminin kalanına) tamamlayıcı bir toplamsal-hata sağlayarak, trafiğin değişmediğini ve doğru varış adresine vardığını kanıtlar. UDP birçok geçiş-tabanlı (transaction-based) uygulama sistemlerinde kullanılan servislerin minimal bir seviyesidir, şöyle ki eğer tüm TCP servislerine ihtiyaç yoksa UDP oldukça kullanışlı olur.

Kaynak Portu (16 bit)	Varış Portu (16 bit)
Uzunluk (16 bit)	Checksum (16 bit)
Veri (Değişken)	

Şekil 1.5. UDP datagramının formatı

## 1.5. Kablosuz Yerel Alan Ağları

Kablosuz iletişim ağları iki veya daha fazla bilgisayar veya sayısal cihazın birbirleri ile kablosuz veri iletişimi sağlamaları ile oluşan yapıdır. Kablosuz iletişim ağlarını hizmet yapısı, çalışma prensibi veya mimarisine göre farklı şekillerde gruplandırmak mümkündür. Genel yaklaşıma göre kablosuz iletişim ağları 4 sınıfta toplanabilir.

- Kablosuz geniş alan ağları: Bir ülke ya da dünya çapında yüzlerce veya binlerce kilometre mesafeler arasında iletişim sağlayan ağlardır. Örnek olarak GSM, GPRS, 3G verilebilir.
- Kablosuz metropol alan ağları: Bir şehri kapsayacak şekilde yapılandırılmış iletişim ağlarına veya birbirinden uzak yerlerdeki yerel bilgisayar ağlarının birbiri ile bağlanması ile oluşur. Örnek olarak yeni gelişen Wimax verilebilir. IEEE 802.16 standardı bu ağlar için geliştirilmektedir.
- Kablosuz yerel alan ağlar: bir bina, kampüs gibi sınırlı bir coğrafi alanda kurulan ve çok sayıda kişisel bilgisayarın yer aldığı ağlardır. IEEE 802.11x bu ağlar için kullanılmaktadır.
- Kablosuz kişisel alan ağları: ev ya da küçük iş yerinde birkaç bilgisayar ve çevre biriminden oluşan ağlara denir. Bluetooth , bu ağların öncülüğünü yürütmektedir.

Kablosuz WLAN çalışmalarında IEEE (Elektrik ve Elektronik Mühendisleri Enstitüsü-Institute of Electrical and Electronic Engineers) standart geliştirerek bu teknolojinin daha fonksiyonel olarak kullanılmasını sağlamıştır. İlk olarak IEEE 802 LAN/MAN standart komitesi 1997'de IEEE 802.11 standardını yayımlamıştır. Bu standarda göre 2.4 GHZ frekans bandında çalışmaktaydı. Aynı zamanda, FHSS(Frequency Hopping Spread Spectrum) ve DSSS (Direct Sequence Spread) kullanılmaktaydı ve 2 Mbps'e kadar data iletişimi gerçekleştirebilmekteydi (Agilent Technologies, 2005). 802.11 standardındaki temel amaç mevcut kablolu LAN'ların kablosuz hale gelmesini sağlamak ve kablolu sistemlerle mobil sistemleri uyumlu olarak çalıştırmaktır. Zaman içerisinde IEEE tarafından yayımlanan standardın başarılı olması ile 802.11x ile yeni standartlar yayımlanmıştır.

Tablo 1.1. Kablosuz yerel alan ađ standartları

Kategori /Standart	Max. Veri Oranı (Data Rate)	Frekans (Hz)	Mesafe (Bina İçi)	Mesafe (Bina Dışı)
IEEE (1997)	802.11 2 Mbps	2.4GHz	20 m	100 m
IEEE (Wi-Fi)	802.11a 54 Mbps	5.2GHz	35 m	120 m
IEEE (Wi-Fi)	802.11b 11 Mbps	2.4GHz	38 m	140 m
IEEE (Wi-Fi)	802.11g 54 Mbps	2.4GHz	38 m	140 m
IEEE (Haziran 2009)	802.11n 248 Mbps	2.4GHz, 5.2GHz	70 m	250 m
IEEE (Haziran 2008)	802.11y 54 Mbps	3.7GHz	50 m	5000 m
IEEE (WiMAX)	802.16 70 Mbps	10-66 Ghz	?	50 000 m
IEEE (WiMAX)	802.16a 70 Mbps	2-11 Ghz	?	?
HiperLAN1	20 Mbps	5.2GHz	?	?
HiperLAN2	54 Mbps	5.2GHz	?	?
HomeRF	10 Mbps	2.4 Ghz	45 m	-
Bluetooth	1 Mbps	2.4 Ghz	10 m	-

IEEE 802.11b, 802.11a ile aynı tarihte çıkmıştır. DSSS modülasyon tekniğini kullanmaktadır. 2.4 Ghz bandında 2400-2483.5 Mhz aralığında 11 Mbps'a kadar veri iletişim hızlarına sahip olabilmektedir. Dizüstü ve masaüstü bilgisayarlar da yaygın bir şekilde kullanılmıştır. Ancak zaman içerisinde diğer standartların üstün özellikleri kullanım oranlarını düşürmüştür. 802.11b standardının en büyük problemlerinden biriside enterferasyon olasılığıdır. Bu durum veri iletişim hızını düşürmekte yada kopmalara neden olabilmektedir.

IEEE 802.11a Standardı, OFDM modülasyon tekniği kullanarak 5 GHz frekans bandında 54 Mbps veri iletişim hızına sahiptir. Bu teknoloji 802.11b'ye göre daha üstün bir standarttır. 802.11b'de 11Mbps olan veri iletişim hızı 802.11a'da 54 Mbps'e çıkmıştır. Hızın artması ise videolarda kesintisiz görüntü elde etmede önemli olduğundan 802.11a standardında daha etkili sonuçlar elde edilebilmiştir. Radyo frekansları vericiden alıcıya doğru giderken OFDM tekniği sinyal iletiminde daha verimli sonuçlar doğurmaktadır. Çünkü OFDM sinyal yansımalarından daha az etkilenmektedir. 802.11b'ye göre kapasitesi daha yüksektir. 5 Ghz bandında enterferans yapmayan 12 kanal WLAN sistemleri için ayrılmıştır. 2,4 Ghz bandında ise sadece 3 kanal bulunmaktadır. 802.11a'da veri iletişim hızı 54 Mbps olduğu için 802.11b'ye göre 5 kat daha hızlıdır. 802.11b, Orthogonal Frequency Division Multiplexing (OFDM) modülasyon tekniğini kullanmakta ve 5 Ghz frekans bandında çalışmaktadır. Bu durum enterferans riskini 802.11b'ye göre düşürmektedir. temi'nden bahsedilmiştir.

IEEE 802.11g, 802.11b gibi 2.4 GHz bandında çalışmaktadır. Bu standart toplam 54 Mbps bant genişliği sunar. OFDM modülasyonunu kullandığından yüksek hızda çalışmaktadır. Bu modülasyon 802.11a standardında da kullanılmaktadır. 802.11g'nin dezavantajları 802.11b benzerleri ile aynı olup üst üste binmeyen sadece üç kanal kullanılması ayrıca yüksek frekans yayılımı yapan mikrodalga fırın gibi sistemlerden etkilenmesidir.

IEEE 802.11 standardı kablosuz istasyonlar ve kablolu ağ altyapısının bağlantısını geliştirmek için kullanılan önemli bir iletişim standardıdır. IEEE 802.11 protokolü ve bağlantı teknolojisi aracılığıyla toplantı odaları, koridor, lobiler, kafeteryalar

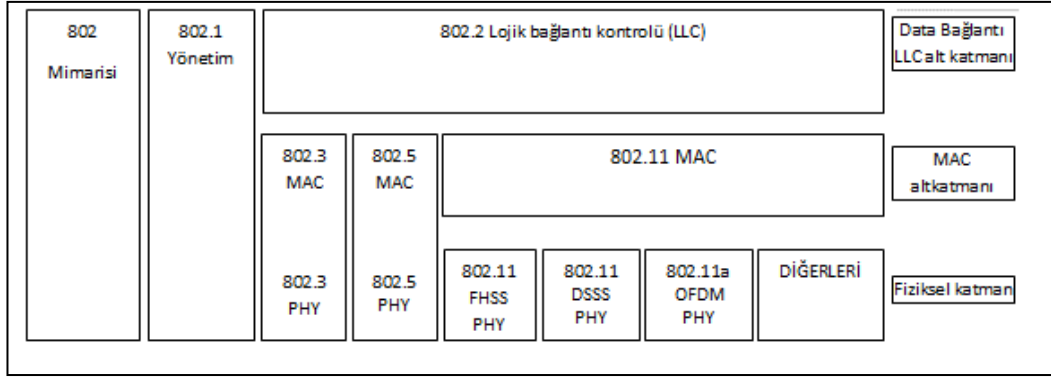
vb. mekanlarda ağ erişimi sağlanabilir. OSI modeli bağlı kalınarak kablosuz mimariyle çalışan sistemlerin tasarımı yapılabilmektedir.(Gast,2002)

Şekil 1.6' da gösterildiği gibi, kablosuz ağlar farklı yapılara sahip olabilirler. Bazılarında ortam bir Access point istasyonu tarafından yönetilir, bazılarında ise bilgisayarlar birbirlerine kablosuz bağlantıyla bağlanabilir. 802.11 standartları her iki konfigürasyonu da dikkate alacak şekilde tasarlanmıştır.

IEEE 802 özellikleri OSI modelinin iki alt katmanı üzerinde odaklanmıştır, çünkü OSI model fiziksel ve veri katmanları birlikte çalışmaktadır. MAC, data göndermek ve erişim araçlarının nasıl düzenleneceğinin kurallarını ayarlar, fakat iletişimin detayları ve kabulü fiziksel katmana bırakılmıştır.

802 serisinin bireysel özellikleri ikinci numarasıyla düzenlenmiştir. Örneğin 802.3, Ethernet'le ilgili çarpışma algılama (Collision Detection-CSMA/CD ) ile Carrier Sense Multipli Access network olarak isimlendirilmiştir ve 802.5 Token Ring'dir. Diğer özellikler 802 protokol katmanının başka kısımlarını oluşturmaktadır. 802.2 herhangi bir düşük katman LAN teknolojisinde kullanılabilir LLC, ortak bir link katmanıdır. 802 ağları için yönetimsel özellikler 802.1'de tanımlanmıştır. 802.1'in çok sayıda hükümleri arasında, köprüleme(802.1d) ve sanal LAN'lar veya VLAN'lar (802.1d) bulunmaktadır. (Gast, 2002)

OSI modelinin temel tanımlamaları yapılırken, fiziksel katmanın üst katmanlardan gelen veriyi elektriksel formata çevirerek bit formatında iletmekte olduğu tanımlanmıştır. 802.11 kablosuz protokol standardını belirttiği için verinin radyo sinyalleriyle gönderileceğini vurgulamaktadır. Şekil 1.7.'de 802.11, 802.11a, 802.11b olarak kablosuz ağ türleri sıralanmıştır. Burada sona gelen a, b gibi ifadeler kablosuz ağın teknolojik özelliğini belirtmektedir. Buna göre, fiziksel katmanda kullanılacak modülasyon teknikleri değişeceğinden Şekil 2.7'de frequency hopping spread-spectrum (FHSS), a direct-sequence spread-spectrum (DSSS) vb. kullanılmıştır. Diğerleri olarak belirtilen, 802.11g DSSS/OFDM, 802.11n MIMO/OFDM kullanılmaktadır.



Şekil 1.6. Ağ ortamında Protokol genişlemesinin dinamik oluşumu

802.11 kablosuz iletişim modelinde veriler karşıya MAC adreslerine göre teslim edilecekleri için fiziksel katman ile MAC katmanı organizeli çalışmaktadırlar.

### 1.5.1. 802.11 mac

802.11 tanımlamasının anahtarı MAC, yani Media Access Control'dür. Bu, tüm fiziksel katmanları çalıştırır ve kullanıcı bilgisinin havadan aktarımını kontrol eder. Bu, çekirdek çerçeveleme operasyonlarını sunar ve kablolu ağ omurgası ile etkileşimi sağlar. Birbirleriyle bir arada çalışabilmeleri gereken farklı fiziksel katmanlar farklı aktarım hızları sunabilirler.

802.11, daha önceki IEEE 802 versiyonlarından radikal bir şekilde ayrılmamaktadır. Bu standart, Ethernet türündeki ağ yapısını başarılı bir şekilde radyo bağlantılarına adapte etmiştir. Ethernet gibi 802.11 de aktarım ortamına erişimi kontrol etmek için Carrier Sense Multiple Access (CSMA) şemasını kullanmaktadır. Ancak çarpışmalar çok değerli aktarım kapasitesini boşa harcayabilmektedirler. Bu nedenle Ethernet'te kullanılan Collision Detection (CSMA/CD) yerine 802.11'de Collision Avoidance (CSMA/CA), yani çarpışmadan kaçınma şeması kullanılmaktadır. Ethernet de olduğu gibi 802.11 de merkezi bir denetçisi olmayan dağıtık erişim şemasını kullanmaktadır. Tüm 802.11 istasyonları, ortama erişebilmek için aynı yöntemi kullanırlar. 802.11 ile Ethernet arasındaki en önemli farklar ise temelde yatan ortamdaki kaynaklanmaktadır.

Kablosuz ortama erişim, koordinasyon fonksiyonları tarafından kontrol edilmektedir. Ethernet'tekine benzer şekilde CSMA/CA erişimi Distributed



Coordination Function (DCF) yani dağıtık koordinasyon fonksiyonu tarafından sunulmaktadır. Eğer çekişmesiz bir servis gerekiyorsa DCF'in üzerinde oluşturulan Point Coordination Function (PCF) yani nokta koordinasyon fonksiyonu kullanılarak sunulabilir. Çekişmesiz servisler sadece altyapılı ağlarda sunulabilmektedir.

DCF, standart CSMA/CA erişim mekanizmasının temelidir. Ethernet'te olduğu gibi gönderim öncesinde ilk olarak radyo bağlantısının temiz olup olmadığını kontrol eder. İstasyonlar, çarpışmalardan kaçınmak için, ilk vericinin kanalı tutması ile her çerçeve sonrasında rastgele bir geri çekilme kullanır. DCF, bazı durumlarda çarpışma ihtimalini daha da azaltmak için CTS/RTS temizleme tekniğinden faydalanabilir.

PCF Nokta koordinasyonu, çekişmesiz servisler sunar. Çekişme olmayan bir ortamın sunulduğundan emin olmak için Point Coordinators adındaki özel istasyonlar kullanılır. Point Coordinators, erişim noktalarında bulunurlar ki bu, PCF'in sadece altyapılı ağlarda kullanılabileceği anlamına gelmektedir. Standart çekişme bazlı servislere karşı üstünlük sağlamak için PCF, istasyonların çerçevelerini belirli bir zaman aralığı sonrasında göndermelerine izin verir. PCF yaygın olarak kullanılan bir yöntem değildir.

DCF'de Hata tespiti ve düzeltme, atomik çerçeve değişimini başlatan istasyona bağlıdır. Bir hata tespit edildiği zaman verinin sahibi olan istasyonun çerçeveyi yeniden göndermesi gereklidir. Gönderen istasyon hataları tespit etmek zorundadır. Bazı durumlarda gönderen, alıcıdan gelen pozitif onaydaki hatadan çerçevenin kaybolduğunu anlayabilir. Çerçeveler yeniden gönderildikleri sürece yeniden deneme sayacı da arttırılır.

Her bir çerçeve veya parçanın kendisi ile ilişkili tek bir yeniden deneme sayacı vardır. İstasyonların ise Short Retry Count (Kısa Yeniden Deneme) ve Long Retry Count (Uzun Yeniden Deneme) olmak üzere iki sayaçları vardır. RTS eşliğinden küçük olan çerçeveler kısa ve büyük olanlar ise uzun olarak düşünülürler. Çerçeve, boyutlarına bağlı olarak istasyon üzerindeki uzun veya kısa sayaç ile ilişkilendirilirler. Çerçeve yeniden deneme sayacı 0'dan başlar ve çerçeve aktarımında yaşanan her sorunda bir arttırılır.

Kısa yeniden deneme sayacı aşağıdaki koşullarda sıfırlanır:

- Aktarılan bir RTS'e cevap olarak CTS çerçevesi alındığında.
- Parçalanmamış bir aktarım sonrasında bir MAC katmanı onayı alındığında.
- Bir Broadcast veya Multicast çerçevesi alındığında.

Uzun yeniden deneme sayacı aşağıdaki koşullarda sıfırlanır:

- RTS eşiğinden daha uzun olan bir çerçeve için MAC katmanı onayı alındığında.
- Bir Broadcast veya Multicast çerçevesi alındığında.

İlişkilendirilmiş olan yeniden deneme sayacına ek olarak parçalara da MAC tarafından bir maksimum yaşama süresi verilir. İlk parça aktarıldığı zaman yaşama süresi sayacı başlatılır. Yaşama süresi sınırına ulaşıldığı zaman çerçeve atılmıştır ve geri kalan çerçevelerin gönderilmesi için herhangi bir deneme de bulunulmaz.

Diğer bir çok ağ protokolünde olduğu gibi 802.11 de güvenilirlik için yeniden gönderim fonksiyonunu kullanmaktadır. Veri aktarımı, atomik sekans sınırları içerisinde gerçekleştirilir ve aktarımın başarı ile sonuçlanabilmesi için tüm sekansın tamamlanması gereklidir. İstasyon veri gönderdikten sonra, aktarım sırasında sorun yaşanmadığından emin olabilmek için mutlaka onayını almış olmalıdır. Hatalı aktarımlar, çerçeve (veya parça) ile ilişkilendirilmiş olan yeniden deneme sayacının arttırılmasına neden olur. Eğer yeniden deneme limiti aşılsa ilgili çerçeve atılır ve kaybolduğu üst seviye katmanlara bildirilir.

Kısa ve uzun çerçevelerin var olmasının bir nedeni, ağ yöneticilerine sahip oldukları ağın sağlamlığını farklı çerçeve uzunlukları için ayarlayabilmelerine izin vermektir. Büyük çerçeveler daha çok tampon alanına ihtiyaç duyarlar. Böylece iki farklı yeniden deneme limitine sahip olan potansiyel bir uygulama uzun deneme limitini düşürerek ihtiyaç duyulan toplam tampon alanını da küçültebilecektir.

### 1.5.2. 802.11e protokolü

Kablosuz ağlarda iletişim ortamı paylaşımlıdır ve çarpışma (collision) kaçınılmazdır. Kablosuz ağlardaki alıcı ve verici istasyonlar aynı frekansı kullandığından tek yönlü (halfduplex) yapıda çalışır. Bir kablosuz ağın çift yönlü (full duplex) yapıda çalışabilmesi için veri gönderiminin ve alımının farklı yapılması gerekir. Bu pratikte mümkün olsa da IEEE 802.11 standartları buna izin vermemektedir. Kablosuz ağlara her geçen gün artan talep doğrultusunda çoklu iletişim uygulamalarının sorunsuz gerçekleşmesinde IEEE 802.11 standardı yetersiz kalmaktadır. Günümüz şartlarını göz önüne alınarak geliştirilen teknikler ile bu ağlarda servis kalitesi sağlanarak sorunsuz bir iletişim sağlanabilmektedir.

Aynı anda veri gönderen iki kablosuz istasyonun sinyalleri karışır, çarpışma meydana gelir ve gönderilmek istenen veri ağda gürültü veya hata olarak algılanır. Bir çarpışma olup olmadığını tespit etmenin kesin bir yolu yoktur. Veri gönderen istasyon bile bunu anlayamayacaktır. Çünkü o sırada alıcıları kapalıdır. Basit bir geri besleme olarak göndericinin her çerçevesine karşılık alıcı bir onaylama çerçevesi (frame) gönderir. Onaylama (ack) çerçeveleri temel düzeyde çarpışma tespit araçları olsalar da çarpışmaları ilk yerinde önlemeyi sağlayamazlar.

DCF'nin gelişmiş bir versiyonu olarak IEEE 802.11e standartları ile birlikte EDCF kullanılmaya başlanmıştır. Bu gelişimin en önemli kısmı CWmin ve CWmax rastlantısal değerlerinin trafik sınıflandırması baz alınarak ayarlanabilmesi oluşturur. EDCF kullanımı ile DCF kullanımdaki tüm trafik aynı DIFS kadar beklemesine rağmen rastlantısal backoff eğrinin yaratılmasında kullanılan CWmin değeri trafik sınıflandırılmasına bağlı değişir. EDCF DIFS yerine AIFS kullanır. AIFS nin DIFS ten tek farkı her izin kategorisi için farklı değerde olmasıdır. Yüksek öncelikli trafik daha küçük bir CWmin değerine sahip olurken, best-effort trafik çoğunlukla daha uzun bir rastlantısal backoff değeri yaratan CWmin değerine sahip olur. Her istasyon gönderim süresinin en üst sınırını (TXOP) belirler. TXOP, istasyonun iletim hakkına sahip olduğu süre aralığı olup gönderim başlangıç zamanı ve maksimum süre ile ifade edilir (Kong,2004). EDCF, istasyonlar için 8 kullanıcı önceliği ve 4 izin grubu sunar.

Bir uzun grubunda birden fazla kullanıcı izinleri tanımlanabilir. Ağ üzerindeki her istasyon iletim çevrelerinde bulundukları erişim kategorileri ile sınıflandırılır.

EDCF operasyonu:

1. X istasyonu çerçeve iletirken, diğer istasyonlar rastlansal bir süre beklerler.
2. Voice 1 ve Voice 2 istasyonları, ses çerçeveleri ilettikleri için düşük bir CWmin değeri seçer (3), böylelikle de rastlansal bekleme süreleri kısılır. Bunun aksine Best Effort 1 ve Best Effort 2 istasyonları ise daha yüksek CWmin değeri seçtikleri için (31) daha uzun bir rastlansal süre belirler.
3. Voice 1 en düşük rastlansal süreye sahip olduğu için ilk iletimi gerçekleştirir. Voice 1 çerçeve iletirken diğer istasyonlar bekler. Ağa yeni katılan Voice 3 istasyonu da rastlansal bir süre belirler.
4. Voice 1 iletimi tamamlandıktan sonra, her istasyon AIFSD süresi kadar bekler ve rastlantısal sürelerini azaltırlar. Voice 2 rastlansal süresi il ksona eren istasyon olarak ilettime başlar.
5. Voice 2 istasyonu iletimi tamamlayınca, AIFSD süresi kadar bekleyen istasyonlar bekleme sürelerini azaltırlar ve bu süreyi ilk tamamlayan Best Effort 2 ilettime başlar. Fakat bu sırada daha yüksek önceliğe sahip olan Voice 3 istasyonu çerçeve gönderememiş olur. Bunun nedeni ise ağa sonradan dahil olan Voice 3 istasyonun seçtiği düşük rastlansal sürenin, Best Effort 2 istasyonun zamanla azalttığı bekleme süresinden yüksek olmasıdır.
6. Best Effort 2 istasyonundan sonra diğer istasyonlar AIFSD süresi kadar beklerler ve rastlansal sürelerini azaltırlar. Bu süreyi tamamlayan Voice 3 çerçeveyi iletir. Süreç bu şekilde devam eder(Szigeti, 2004).

Wi-Fi organizasyonu tarafından servis kalitesinin sağlanması adına gerçekleştirilen kablosuz multimedya (WMM), IEEE'nin 802.11e standardı ile kullanılmaktadır. WMM, trafiği dört kategoride ses, video, best-effort ve arka plan olarak inceler ve veri önceliğini bu kategorilere göre yapar. Kablosuz ağlarda kullanılan çarpışmadan kaçınma tekniğinin geliştirilmiş halidir. WMM, kaynakların bekleme sürelerini, uygulamaların kategorine göre belirler ve öncelikli paketler için bekleme süresi daha kısa olur.

## 1.6. Kablosuz Ağlarda İletim Katmanı

### 1.6.1. Kablosuz ağlarda TCP

Transmission Control Protocol(TCP), güvenilir, bağlantı yönelimli, çift yönlü, byte katarlı, ulaşım katmanı protokolüdür. Uç kullanıcıların uygulamaları için uçtan uca akış ve tıkanıklık kontrol sağlayan bir protokoldür. İnternet trafiğinin büyük bir kısmı TCP oluşturur. TCP, kablolu ağlar için tasarlanmıştır, TCP kablolu ağlarda çok verimli çalışırken, kablosuz ağlarda iyi performans verememiştir. Kablosuz ağlardaki TCP uygulamalarındaki performans problemlerinin ana kaynağı, iletim hatalarıdır. Paketlerdeki bazı hatalar, alt katmanlardaki FEC kodları tarafından düzeltilirken, daha fazla hata paketlerin bozulması olarak kabul edilir. Bozulan paketler TCP'nin eline geçmeden atılır. TCP, paket kayıplarını ağ tıkanıklığı olarak kabul ettiği için, tıkanıklık penceresini azaltarak tepki verir. Bu durumda, kablosuz iletim hataları, ağ tıkanıklığı ile ilişkilendirilir, böylece uygunsuz olarak tıkanıklık penceresinin küçülmesi, TCP uygulamaları için gereksiz kayba neden olur.

Araştırmacılar 1990'ların başlarından ortalarına doğru, kablosuz hatlarda verilen yüksek bit hata oranları ve olası el verme kaybı ile TCP'nin tıkanıklık kontrolü cevabının kablosuz kurulumlarda problemlere sahip olabileceğini fark ettiler. Üç geniş yaklaşım sınıfı, bu problemle başa çıkarmak için olasıdır (Kurose,2010).

Yerel Kurtarma: Yerel kurtarma protokoller biti hatalarını, meydana geldikleri yerde ve zamanda kurtarır. 802.11 ARQ protokolü veya hem ARQ'yu hem FEC'yi kullanan daha gelişmiş yaklaşımlar gibi (Ayanaoğlu ve diğerleri, 1995) .

TCP göndericisi kablosuz hatların farkındadır: Yerel kurtarma yaklaşımlarında TCP göndericisi, segmentlerinin kablosuz bir hattı içinden geçtiğinden habersizdir. Alternatif bir yaklaşım, TCP göndericisinin ve alıcısının kablolu ağda meydana gelen tıkanıklık kayıplarını ve kablosuz hatta meydana gelen bozulma/kayıbı birbirinden ayırması için ve sadece tıkalı kablolu hat kayıpları durumunda tıkanıklık kontrolünü çağırması için kablosuz hattın varlığından haberdardır (Balakrishnan ve diğerleri, 1997). Uç sistemlerin bu ayırımı yapabileceğini varsayarak TCP'nin çeşitli tiplerini araştırır. (Wei 2005) bir uçtan

uca yolun kablolu ve kablosuz segmentleri üzerinde kayıpları ayırt etmek için teknikleri araştırır.

Bağlantı bölme yaklaşımları: Bağlantı bölme yaklaşımında (Bakre ve Badrinath 1995) , gezici kullanıcı ve diğer uç nokta arasındaki uçtan uca bağlantı, iki taşıma katmanı bağlantısına bölünür: Bir gezici ana sistemden kablosuz erişim noktasına, diğeri kablosuz erişim noktasından diğer iletişim ucuna. Böylece uçtan uca bağlantı, bir kablosuz kısım ile bir kablolu kısmın birleşmesinden meydana gelir. Kablosuz segment üzerindeki taşıma katmanı, standart bir TCP bağlantısı olabilir veya UDP üzerinden özel olarak kuyruklanmış hata kurtarma protokolü olabilir. (Yavatkar 1993) kablosuz bağlantı üzerinde, bir taşıma katmanı seçici tekraralama protokolünün kullanımını araştırır. (Wei 2006)'da bildirilen ölçümler, bölünen TCP bağlantılarının, hücresele ver ağlarında yoğun bir şekilde kullanıldığını ve önemli ölçüde gelişmelerin, bölünen TCP bağlantılarının kullanımı ile yapabileceğini belirtir.

Kablosuz Ağlarda TCP Problemleri sınıflandırması şu şekildedir.

Rastgele Kayıp: Kablosuz ağlarda, sinyal azalması nedeniyle kayıp olarak görülen veri segmentleri de, ağ tıkanıklığı sebebi ile olduğu düşünülür. Kod bölmeli çoklu erişim (CDMA)'da kayıp yarı yarıyadır (Mitzel, 2000). Bu yanlış anlamadan meydana gelen segment kaybı ile beraber, tıkanıklık kontrol mekanizmaları, TCP bağlantı gönderim oranını azaltır ve bazı segmentler gereksiz yere tekrar iletilir. Bu performansı düşürür.

Patlamalı Kayıp: Sinyal azalmasının başlaması sebebiyle patlamalı kayıp oluşabilir. Uzun süreli kontrol edilmeyen kanal engellemeleri ilişkili paket kayıplarına neden olabilir. Mobil haberleşmede, hareket halindeki mobil alıcı başka bir istasyondan bağlantı kurarken geçen zaman IEEE 802.11b kablosuz ağlar için yaklaşık 1-2 sn dir (Velayos, 2004). Handoff olayı da patlamalı kayba neden olabilir (Leoung ve diğerleri 2006).

Paketin Yeniden İstenmesi: Göndericinin peş peşe gönderdiği paketlerin, alıcı tarafta farklı sırada gelmesi ağın davranışına göre paketi yeniden istemesine neden olabilir. Son çalışmalar gösteriyor ki, paketin yeniden istenmesi nadir gerçekleşen

bir olay değildir. Israrcı ve kati paketin yeniden istenmesi durumunda, TCP' deki trafik kontrol mekanizmalarını bozmaktadır. Mobil haberleşmede de handoff olayı esnasında, eski istasyondan paket yeniden istenebilir.

Kablosuz ağlarda TCP çözümleri 4 farklı stratejiye ayrılmıştır: Tıkanıklık Tespiti Yaklaşımı, Durum Asma Yaklaşımı, Cevap Erteleme Yaklaşımı, Hibrit Yaklaşım.

Tıkanıklık Tespit Yaklaşımı: Ağın durumunu ölçen ve bu duruma göre eğer ağda tıkanma oluştuysa, ölçümlere göre doğru trafik kontrol stratejisini seçen metotlardan oluşur. Bir başka deyişle, doğru trafik kontrolü ile, tıkanıklığı ve tıkanıklık olmayanı birbirinden ayırmaya çalışmaktır.

Uygulamalar: TCP-Peach, TCP-Probing, TCP Westwood, TCP Venno, TCP Jersey, JTCP, TCP-Casablanca,

Durum Asma Yaklaşımı: Ağın o anki durumunu tespit ederek, ağda tıkanıklıktan dolayı meydana gelmeyen kayıplardan kaçınmak için, TCP haberleşme aktivitesinin ne zaman askıya alınacağına ne zaman devam ettirileceğine karar vermek için oluşturulan bir grup tekniktir. Bağlantının durumu, askıya alınınca, ağın durumuna göre belki eski haline dönebilir belki dönmeyebilir.

Uygulamalar: Freze-TCP, ILC-TCP, TCP-Feedback, ELFN, TCP Door

Cevap Erteleme Yaklaşımı: TCP istemcilerinin, trafik kontrol cevaplarını geciktirerek kablosuz ağdaki problemleri azaltmaya çalışır.

Uygulamalar: Del-Ack, TCP-ADA, TCP-DCR, ATCP

Hibrit Yaklaşım: Yukarıda sözü edilen 3 yaklaşımın beraber kullanıldığı yöntemlerdir. TCP istemcisi yukarıdaki mekanizmaların kombinasyonunu da kullanabilir.

Kablolu-kablosuz hibrit ağlarda TCP performansını artırma araştırmalarını 4 grupta toplayabiliriz: Bölümlü Bağlantı, Proxy-tabanlı, Veri bağı katmanı hata kontrol ve uçtan uca (Mondal ve diğerleri, 2007).

Bölümlü bağlantı yaklaşımında, ana istasyonda bağlantı, kablolu ile kablosuz arasında bölünüyor (Bakre ve diğerleri, 1995). Kablosuz ağ ile güvenilir bağlantı kuruluyor. Bağlantının kablolu bölümü, kablosuz ağın kaprisinden korunmuş oluyor. Bu yaklaşımın iki dezavantajı vardır: Uçtan uca TCP tanımına uymuyor ve ana istasyonda durumun normal sürdüğünü bildiren bir başlık ekleniyor.

Proxy tabanlı yaklaşımda, örneğin Snoop, ana istasyonda bulunan Proxy ile tek bir bağlantı kurulur. Proxy, kablosuz bağlantıyı denetler ve lokal iletimdeki kablosuz kayıpları düzeltmeye çalışır.

Veri bağı katmanı hata kontrol yaklaşımı, ileri hata düzeltme (FEC) ve otomatik tekrar cevabı (ARQ) kullanarak, kablosuz kayıpları azaltmaya çalışır.

AIRMAIL protokolü, ileri doğru hata kontrolü (FEC-Forward Error Correction) ve otomatik tekrarlı cevap (Automatic Repeat Request-ARQ) kullanan asimetrik güvenilirlik sağlayan bir uygulamadır (Ayanoglu ve diğerleri,1995). Asimetriklik, mobil istasyon haberleşmesi ve hesaplama karmaşıklığını azaltarak mobil istasyonun gücünü azaltmaktır. Haber alma ağırlığı temel istasyondadır. Bu , mobil ucun tekrarlı onayı , paket bloğu aldıktan sonra göndererek aşılmaya çalışılır. Pencere yönetimi ve durum transfer şeması, handoff problemi için adapte edilir. AIRMAIL kullanılan ağlarda, zaman sayaç yönetimi, durum mesajlarının yönetimi ve karar verme işlemi temel istasyonda halledilir. Temel istasyon durumunu mobil istasyona periyodik olarak bildirir. Mobil istasyon birkaç durum mesajını bir tek mesaj olarak birleştirerek gönderir bu sayede gücünü az kullanmış olur. Temel istasyon periyodik olarak bildirdiği durum için bir zamanlayıcı tutar. Zamanlayıcı paket kaybının tespiti için kullanılır. Mobil istasyon, mesaj göndermeden önce cevap isteği gönderir, temel istasyondan cevap gelirse, blok olarak mesaj gönderir ve zamanlayıcı tutarak durum mesajı bekler. Durum mesajı geldiğinde tekrar blok mesaj gönderir. Mobil istasyon mesaj sonunda mesajın bittiğine dair durum mesajı gönderir, temel istasyon CRC hata kodlamasını kullanarak bu mesajlar için hata kontrolü yapar, hatalı paketler yeniden istenir.

TULIP, TCP habersiz, servis haberli bir bağı katmanı protokolüdür. TULIP' teki ana fikir; kablosuz ağ bant genişliğini ve TCP'deki zaman aşımını kullanarak avantaj sağlayan, TCP'nin durumundan haberdar olmayan açık bir bağı katmanı



sağlamaktır (Parse ve diğerleri, 1999). TULIP, temel istasyon yönlendirmeli kablosuz ağlarda bağlantı güvenilirliğini artırmak için kullanılan bir yoldur

TULIP, sonlu zamanda, tekrarlı olmayan, sıralı ve güvenilir veri bağı paketleri(RLP-radyo link protokolü) kullanarak alıcıya güvenilir servis sağlar. TULIP, UDP, bağı katmanı onayları ve TCP onayları için güvenilir olmayan kapsüle edilmiş RLP paketleri kullanır.

TULIP işlemleri kablosuz ağların özelliği olan half-dupleks üzerine kurulmuştur. Aynı anda hem gönderim hem alım olmadığı için çakışma ihtimali de azaltılmıştır. MAC katmanında aynı anda sadece bir paket iletilir. MAC katmanı TRANS ve WAIT isimli iki sinyal kullanır. Paket gönderimine başlandığında TRANS sinyali gönderilir. İkinci paketi göndermeden önce TRANS sinyalini almak için t1 saniye beklenir. Eğer paket boyu büyük olursa ve zaman t1 zamanı yeterli değil ise MAC katmanı bunu TULIP'e t1 den daha büyük bir zaman beklemesi için bildirir. Böyle bir durumda MAC katmanı TULIP'e WAIT sinyali gönderir. Gönderici akış kontrol ve hata düzeltmeyi yönetir. Hata düzeltme seçmeli onay mekanizması ile olur.

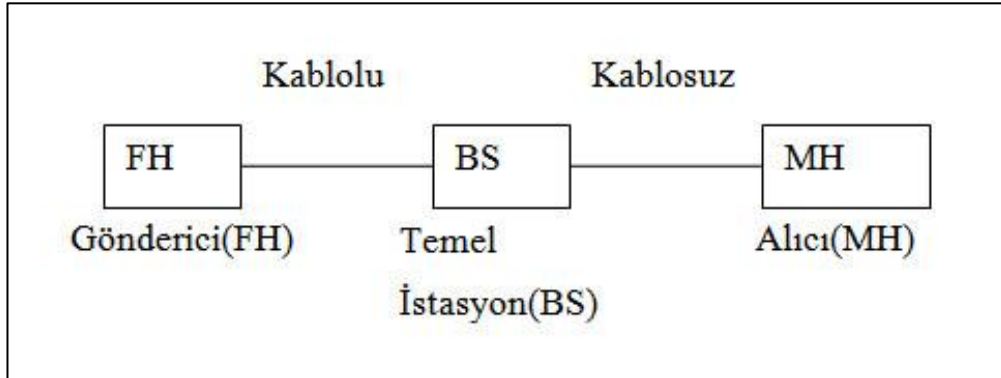
Kablolu-kablosuz hibrit ağlarda, TCP performansını artırma yöntemleri iki sınıfa ayrılabilir. İlk yaklaşımda, rast gele kablosuz kayıpları tıkanıklık kayıplarından ayırt etmeye çalışılır ve farklı katmanlardaki değişikliklerle TCP kendi için düzenler. İkinci yaklaşımda, sınır bağlantı kapasitesi hesaplanır ve gönderim oranına göre ayarlanır.

En yaygın kullanılan iki tip TCP katmanında bölümlenme şekli vardır.

I-TCP: TCP bağlantısı kablolu ve kablosuz olarak ikiye bölünür. Kablosuz ağdaki tıkanıklık sebebi ile oluşmamış hatalı veya ulaşmamış paketlerin, kablolu ağdaki gönderici tarafından tekrarlı olarak gönderilmesi engellenir. Kablolu –kablosuz iki tarafta da geleneksel TCP kullanılır.

İ-TCP &SACK: I-TCP'nin kablosuz tarafında SACK kullanılarak yapılır. Kablosuz ağ tarafında ulaşamayan veya hatalı giden paketlerin tekrar istenmesinde seçmeli onay (Selective Ack) gönderilir. Bu sayede tekrarlı

onaylardan ve tekrarlı paket gönderimleri azaltılır. I-TCP'den daha iyi performans verir.



Şekil 1.7. TCP bölümlenme

Bağlantı bölmenin avantajları:

- BS ile MH bağlantısı, FH ile BS bağlantısından bağımsız olarak optimize edilebilir. Farklı akış kontrol ve hata kontrol yapılabilir.
- Lokal hata düzeltme işlemi yapılabilir. Kablosuz ağdaki RTT zamanı daha az olduğu için hızlı hata düzeltme işlemi yapılabilir.
- Kendine uygun BS ile MH bağlantısı kullanılarak iyi performans elde edilebilir. Örneğin seçmeli onay metodu geleneksel TCP'den daha iyi sonuç verir.

Dezavantajları:

- Uçtan uca anlamı kaybolmaktadır. BS'ten gönderilen onay mesajı göndericiye ulaştığında, alıcı taraf paketi BS'ten almamış olabilir.
- BS zor durumda kaldığında problemler yaşanacaktır. BS'den onay gönderilen fakat alıcıya ulaşmayan paketin BS'de tutulmadığı durumda veri kaybı olacaktır ve TCP güvenilir olmayacaktır.
- BS, handoff olduğu zaman zor durumda kalabilir ve paket kaybı olabilir.
- Her TCP bağlantısı için BS'de tampon alanı olması gerekir.
- BS'te ekstra veri kopyalanır, FH ile BS soketi, BS ile MH soketine veri transferi yapılırken kopya veriler olur.
- Veri ve onay farklı bir yol üzerinden giderse bölmenin bir yararı olmaz.

TCP katmanında müdahale olmadan ağ bölümlene uygulamaları da mevcuttur. TCP'nin uçtan uca bağlantısının yapısını bozmadan veri bağı katmanı kontrollü bölümlene de yapılabilir. Veri bağı katmanı kontrollü TCP, bağlantı bölme ile aynı amaca sahiptir. Tek farkı TCP'nin uçtan uca anlayışını değiştirmeden bu işi yapmasıdır.

Veri bağı tekrar iletimini sağlamak için veri paketlerini BS'te tamponlar. MH'tan, BS'e Tekrarlı onaylar(dupacks) geldiği zaman, eğer paket tamponda varsa, kablosuz bağlantıdan tekrar iletim yapılır. BS'te tekrarlı onayları keserek, gönderici tarafta daha hızlı TCP tekrar iletimi sağlar.

Avantajları:

- Eğer seçmeli onay mekanizması kullanılırsa, hızlı bir gönderim sağlanabilir.
- Kablosuz kayıpları lokal olarak düzeltebilir.
- Uçtan uca kavramı korunmuş olur.
- BS'te oluşabilecek bir sorunda performans düşebilir fakat paket kaybı olmaz.

Dezavantajları:

- BS' teki veri bağı katmanı TCP'nin varlığından haberdar olmak zorundadır.
- Kodlanmış TCP başlıkları için yararı olmaz.
- TCP veri ve onay farklı yollardan giderse bir yararı olmaz.

Sonuç olarak kablosuz ağlardaki TCP uygulamalarındaki performans problemlerinin ana kaynağı, iletim hatalarıdır. Paketlerdeki bazı hatalar, alt katmanlardaki FEC kodları tarafından düzeltilirken, daha fazla hata paketlerin bozulması olarak kabul edilir. Bozulan paketler TCP'nin eline geçmeden atılır. TCP, paket kayıplarını ağ tıkanıklığı olarak kabul ettiği için, tıkanıklık penceresini azaltarak tepki verir. Bu durumda, kablosuz iletim hataları, ağ tıkanıklığı ile ilişkilendirilir, böylece uygunsuz olarak tıkanıklık penceresinin küçülmesi, TCP uygulamaları için gereksiz kayba neden olur.

## 1.6.2. Kablosuz ağlarda UDP

UDP, TCP/IP protokol takımının iki aktarım katmanı protokolünden birisidir. Verileri bağlantı kurmadan yollar.

Gelişmiş bilgisayar ağlarında paket anahtarlamalı bilgisayar iletişimde bir datagram modu oluşturabilmek için UDP protokolü yazılmıştır. Bu protokol minimum protokol mekanizmasıyla bir uygulama programından diğerine mesaj göndermek için bir prosedür içerir. Bu protokol hareket yönlendirmelidir. Paketin teslim garantisini isteyen uygulamalar TCP protokolünü kullanır.

Geniş alan ağlarında (WAN) ses ve görüntü aktarımı gibi gerçek zamanlı veri aktarımlarında UDP kullanılır. UDP bağlantı kurulum işlemlerini, akış kontrolü ve tekrar iletim işlemlerini yapmayarak veri iletim süresini en aza indirir. UDP ve TCP aynı iletişim yolunu kullandıklarında UDP ile yapılan gerçek zamanlı veri transferinin servis kalitesi TCP'nin oluşturduğu yüksek veri trafiği nedeniyle azalır. UDP güvenilir olmayan bir aktarım protokolüdür: ağ üzerinden paketi gönderir ama gidip gitmediğini takip etmez ve paketin yerine ulaşip ulaşmayacağına onay verme yetkisi yoktur. UDP üzerinden güvenilir şekilde veri göndermek isteyen bir uygulama bunu kendi yöntemleriyle yapmak zorundadır.

UDP, bağlantı kurulum işlemleri, tıkanıklık kontrolü, akış kontrolü ve tekrar iletim işlemlerini yapmayarak veri iletim süresini minimum protokol mekanizması ile en aza indiren iletim katmanı protokolüdür. UDP uçtan uca güvenli ve sıralı bir iletişim sağlamaz. UDP'nin bu özellikleri sayesinde gecikmeye karşı hassas olan gerçek zamanlı uygulamalarda UDP tercih edilmektedir. Bir UDP paketi başlık ve veriden oluşmaktadır. UDP' de isteğe bağlı olarak, yalnızca iletilen paketin sağlama bilgisi (checksum) ile bütünlük kontrolü yapılır. Sağlama bilgisinin içeriği UDP başlık ve veri kısmından elde edilir. UDP paketi alındığında sağlama bilgisi doğrulanmazsa, bunun anlamı paketin hatalı olduğu ve iletim katmanında yok edileceğidir.

Çoğu ses ve video uygulamalarında, iletim katmanına hatalı olarak ulaşan bir paketin atılması yerine uygulamaya gönderilmesi tercih edilir. Bunun nedeni, çoğu zaman hataların bit hatasından kaynaklanması ve bu hatalarında

uygulamanın kullanacağı bir hata düzeltme mekanizması ile giderilebilmesidir. Kablosuz ağlarda paket hataların çoğu ortam ve kanal hatalarından kaynaklanan tek bit hatalarıdır. Yapılan araştırmalara göre, sağlama bilgisi hataları nedeniyle, alıcı tarafındaki iletim katmanında paketlerin yaklaşık olarak %85'i atılmaktadır. Hatalı paketlerin uygulama katmanına iletilmesinin bir yolu, UDP' de bütünlük kontrolünün kullanılmamasıdır. Bu durumda ise uygulamaya özel başlıkların doğrulanmadan geçirilmesi gibi önemli bir problemle karşılaşılabilir.

Kablosuz UDP'de yapılan çalışmalar, ileri doğru hata düzeltme çalışmalarıdır. Hatalı ulaşan paketler uygulama katmanına hatalı olmalarına rağmen, üst katmandaki uygulama ile düzeltimi yapılmaya çalışılmaktadır. UDP-Lite, UDP-Liter, C-UDP protokol çalışmalarında, gelen paketteki hata, alt katmanlarda gözardı edilerek, uygulama katmanına iletilir, uygulama katmanında hata düzeltme algoritmaları yardımı ile hata düzeltilmeye çalışılır.

UDP-Lite, sağlama bilgisi kontrolünü parçalı olarak yapabileme imkanı vermektedir. Bu sayede UDP'ye göre daha esnek ve verimlidir. UDP-Lite protokolünde amaçlanan, UDP kullanan gerçek zamanlı uygulamalardaki çoğu tek bit hatalı paketleri, uygulama katmanına taşıyabilmektir. Bu protokolde paketler, hassas ve hassas olmayan kısımlar olarak iki kısımda değerlendirilir. UDP-Lite geleneksel UDP ile aynı başlık yapısını kullanır. Aralarındaki tek fark UDP de yer alan "UDP Length" alanı yerine "Coverage" yer almaktadır (Larzon ve diğerleri). Bu alan, paketteki hataya karşı hassas olacak verinin kaç byte olduğunu belirtir. Kısmi bütünlük kontrolü, paketin hassas olan bölümünde yapılır. Eğer bu kısımda bir hata varsa paket yok edilir. Hassas olmayan kısımda oluşacak hatanın kontrolü yapılmaz ve bu sayede veri hatalı olsa bile uygulama katmanına geçebilir. Kullanılan kodlama algoritmalarının çoğu bu hataları düzeltebilmektedir. Veri kısmında oluşan bit hatası kodlama algoritmalarında düzeltilemezse uygulamaya bağlı olarak ses veya görüntüde sorun oluşturmayacak küçük kesintiler yaşanabilir. UDP ile UDP-Lite başlık yapılarında çok fazla fark olmamasına rağmen UDP-Lite, geleneksel UDP kullanan uygulamaları desteklememektedir (Larzon ve diğerleri).

UDPLite'in performansını artırmak için farklı protokoller önerilmiştir. UDP-Liter olarak adlandırılan protokolün amacı, UDPLite ile artırılan verimin daha da iyileştirilmesi ve geleneksel UDP ile tam uyumlu olmasıdır. UDP-Liter protokol önerisinde, UDP başlığında herhangi bir değişiklik yapılmamaktadır ve daha da önemlisi UDP'deki bütünlük kontrol sınaması devam etmektedir. UDP ile tek farkı bütünlük sınamasında hatalı olarak tespit edilen paketi atmak yerine uygulama katmanına "hatalı paket" bilgisi ile göndermektir. Bu sayede uygulama katmanında çalışacak olan uygulama hatalı ve hatasız veriler için farklı algoritmaları çalıştırarak performansı ve kaliteyi artırabilir. UDPLiter'in kullanılabilmesi için UDP paket yapısında bir değişiklik yapılmamış, ancak BSD Soket API'lerinde eklentiler yapılmıştır. Örnek olarak socket() fonksiyonunda SOCK\_DGRAM\_LITER adıyla yeni bir soket tipi eklenmiştir. Buna ek olarak, recvfrom() fonksiyonuna benzer recvfrom\_liter fonksiyonu eklenmiştir. Bu fonksiyon sayesinde ek bir parametre ile uygulama katmanına gönderilen verinin hatalı veya hatasız olduğu bildirimi yapılabilir (Lam ve Liem).

C-UDP protokolünde, hata düzeltme ve kontrol işlemleri uygulama katmanına bırakılmıştır. Mac katmanında herhangi bir hata kontrolü yapılmamaktadır. Bunun yerine hata kontrolünü ve düzeltme işlemlerini uygulama katmanında hata düzeltme algoritmaları ve yazılımlara yapmaya çalışılmıştır(Mondal ve Luqman).

### **1.6.3. Güvenilir UDP (Reliable UDP)**

Güvenilir UDP (Reliable UDP-RUDP)(RFC 1151) , RTP clientleri için paket kaybı ve tıkanıklık problemlerinin çözümü için geliştirilmiş bir protokoldür. TCP'deki tekrar iletim ve tıkanıklık için kullanılan algoritmalara baz alınarak, UDP paket başlığında yapılan değişiklikler ile güvenilirlik sağlanmaktadır(Şekil 2.9). RUDP şu özellikleri içermektedir:

- Client için paket onayı
- Pencere ve tıkanıklık kontrol (server tarafından band genişliği garantisi yok)
- Gerçek zamanlı uygulamalarda tam tamponlamadan daha hızlı

1	2	3	4	5	6	7	8	16 bit
SYN	ACK	EAK	RST	NUL	CHK	TCS	0	Header Length
Sequence number							Ack number	
Checksum								

Şekil 1.8. Protokol Şekli - RUDP (Reliable UDP)

## 2. GERÇEK ZAMANLI İLETİŞİM

Gerçek zamanlı iletişim uygulama sınıfı, insanların birbirleriyle gerçek zamanlı iletişim için ses/videoyu kullanmalarına izin verir. İnternet üzerinden gerçek zamanlı etkileşimli sese genellikle İnternet telefonu denir, çünkü kullanıcının bakış açısından geleneksel devre anahtarlı telefon servisine benzer. İnternet telefonu son derece ucuza, özel santral alışverişi, yerel ve uzun mesafeli telefon servisi sunabilir. Ayrıca varlık tespiti, grup iletişimi, arayan filtreleme, Web telefon bütünleşmiş gibi geleneksel devre anahtarlı ağlar tarafından kolayca desteklenmeyen yeni servislerin dağıtımını da kurabilir. Gerçek zamanlı etkileşimli bir ses/video uygulamasında bir kullanıcının herhangi bir anda konuşabileceğine veya kafasını çevirebileceğine dikkat edilir. Birden çok konuşmacı arasındaki etkileşimli bir sohbet için, bir kullanıcının konuşmasından veya hareket etmesinden, eylemin alıcı ana sistemlerde ortaya çıkmasına kadar geçen süre gecikmenin, birkaç yüz milisaniyeden az olması gerektiğine dikkat edilir. Ses için 150 milisaniyeden küçük olan gecikmeler, bir dinleyici tarafından fark edilemez. 150 ile 400 milisaniye arasındaki gecikmeler, bir dinleyici tarafından fark edilemez, 150 ile 400 milisaniye arasındaki gecikmeler kabul edilebilir ve 400 milisaniyeyi geçen gecikmeler konuşmayı tamamen anlaşılabilir hal getirmiyorsa bile can sıkıcıdır.

Günümüz İnternet'te dağıtılmış olan IP protokolünün, taşıdığı tüm datagramlara en iyi çaba servisini sunduğunu hatırlayın. Diğer bir deyişle İnternet, her datagramı göndericiden alıcıya mümkün olduğunca çabuk göndermek için elinden gelenin en iyisini yapar ama bir paket için uçtan uca gecikme ile ilgili söz vermez. Ayrıca bir paket akışı içerisindeki paket gecikmesi çeşitliliği ile ilgili de herhangi bir söz vermez. Ayrıca bir paket akışı içerisindeki paket gecikmesi çeşitliliği ilgili de herhangi bir söz vermez. TCP ve UDP, IP üzerinde çalıştığından bu taşıma protokollerinin ikisi de çağrıda bulunan uygulamaya bir gecikme garanti vermemesine uyar. Paketlerin zamanında teslimi ile ilgili özel bir çabanın yokluğu



İnternet için başarılı multimedya ağ uygulamaları geliştirmek son derece zorlu bir problemdir. Bununla birlikte İnternet üzerinden multimedya, günümüze kadar hatırı sayılı bir miktarda başarı elde etmiştir. Örneğin akan depolanmış ses/video 5 ile 10 saniye arası kullanıcı etkileşimi gecikmesi artık İnternet’te sıradandır. Ancak trafiğin tepe noktalarında, performans tatmin edici olmayabilir, özellikle aradaki hatlar tıkalı olduğunda (okyanus geçen hatların tıkalı olması gibi).

İnternet telefonu ve gerçek zamanlı etkileşimli video da geniş bir kullanım bulmuştur; örneğin, verilen herhangi bir anda rutin olarak, çevrimiçi yedi milyondan fazla Skype kullanıcısı vardır. Gerçek zamanlı etkileşimli ses ve video paket gecikmesi ve paket seğirmesi üzerine katı kısıtlamalar getirir. Paket seğirmesi, aynı paket akışındaki paket gecikmelerinin çeşitliliğidir. Gerçek zamanlı ses ve video, bant genişliği bol olduğunda ve dolayısıyla gecikme ve seğirme en az olduğunda iyi çalışabilir. Ancak gerçek zamanlı ses veya video, paket akışının bir dereceye kadar tıkalı bir hatta rastlandığında kalite kabul edilemez seviyelere inebilir.

Multimedya uygulamalarının tasarımı, bir çeşit birinci sınıf ve ikinci sınıf internet servisleri var olsaydı ve birinci sınıf paketler sayıca sınırlı olsalar ve yönlendirici kuyruklarında öncelik servisi alsalardı kesinlikle daha kolay olurdu. Böylece bir birinci servis, gecikme duyarlı uygulamalar için tatmin edici olabilirdi. Ancak bugüne kadar İnternet, yönlendirici kuyruklarında paket zamanlamasına genellikle eşitçi bir yaklaşım sergilemiştir. Tüm paketler eşit servis alır, gecikme duyarlı ses ve video paketleri de dahil olmak üzere hiçbir paket yönlendirici kuyruklarında özel bir öncelik almaz. Ne kadar paranız olursa olsun veya ne kadar önemli olursanız olun, sıranın sonuna girmeli ve beklemelisiniz.

Böylece zamanın başlangıcından beri en iyi çaba servisi ile yaşamak zorundayız. Ancak verilen bu kısıtlama ile birçok tasarım kararı verebiliriz ve bir multimedya ağ uygulamasının kalitesini kullanıcı açısından artırmak için birkaç incelik uygulayabilir. Örneğin, ses ve videoyu UDP üzerinden göndererek TCP, yavaş başlama fazına girdiğinde, TCP’nin düşük akışının üzerinden gidebiliriz. Alıcıdaki oynatmayı 100 milisaniye veya daha fazla geciktirerek ağ tarafındaki

seğirme etkisini azaltabiliriz. Göndericideki paketlere zaman damgası koyarak alıcının, paketlerin ne zaman oynatılacağını bilmesini sağlayabiliriz.

İnternet'in ağ katmanı protokolü olan IP, en iyi çaba servisini sunar. Bu, servisin her datagramı kaynaktan hedefe mümkün olduğunca hızlı bir şekilde taşımak için en iyi çabayı sarf edeceği anlamına gelir. Bununla birlikte bir paketin uçtan uca gecikmesinin boyutu veya paket akışı içerisindeki paket seğirmesi ve paket kaybı boyutu ile ilgili herhangi bir söz vermez. Gecikme ve seğirme ile ilgili bir garanti yokluğu paket gecikmesine, seğirmeye veya kayba şiddetli bir şekilde duyarlı olan İnternet telefonu ve gerçek zamanlı video konferansı gibi gerçek zamanlı multimedya uygulamalarının tasarımı için önemli bir zorluk ortaya çıkarır.

Bu ayırmda multimedya uygulamalarının performansının, bir en iyi çaba ağı üzerinden geliştirilebileceği birçok yol vardır. Bu yolların yanında en iyi çaba servisinin kısıtlamalarından da bahsetmek gerekir.

Yukarıda en iyi çaba servisinin paket kaybına, uçtan uca aşırı gecikmeye ve paket seğirmesine neden olabileceğinden bahsettik. Şimdi de bu sorunları daha ayrıntılı olarak inceleyelim. Tartışmayı somut tutmak için bu mekanizmaları, aşağıda tanımlanan bir İnternet telefon uygulaması bağlamında tartışalım. Durum gerçek zamanlı video konferansı uygulamalarında benzerdir (Bolot ve Turletti, 1994).

İnternet telefonu örneğinde konuşmacı, değişen konuşma faaliyetleri ve sessiz dönemlerinden oluşan bir ses sinyali üretir. Bant genişliğini korumak için İnternet telefonu uygulamamız, sadece konuşma faaliyeti esnasında paket üretir. Bir konuşma faaliyeti esnasında gönderici, saniyede 8000 byte hızında paketler üretir ve her 20 milisaniyede gönderici, byteları parçalar halinde bir araya getirir. Böylece bir parçadaki byte sayısı  $(20 \text{ milisaniye}) \cdot (8000 \text{ byte/sn}) = 160 \text{ byte}$  olur. Her parçaya, içeriği aşağıdaki tartışılan özel bir başlık eklenir. Parça ve başlığı, soket arabirimine yapılan bir çağrı üzerinden UDP segmenti içinden sarmalanır. Böylece bir konuşma faaliyeti sırasında bir UDP segmenti her 20 milisaniyede bir gönderilir.

Paketlerin tümü alıcıya varırsa ve küçük, sabit bir uçtan uca gecikmesine sahipse paketler alıcıya bir konuşma faaliyeti esnasında periyodik olarak her milisaniyede

bir varır. Bu ideal koşullarda alıcı, her parça varır varmaz onu çalabilir. Ancak ne yazık ki bazı paketler kaybolabilir ve çoğu paket az bir şekilde tıkalı olan İnternette bile aynı uçtan uca gecikmeye sahip olmaz. Bu nedenle alıcı bir parçayı ne zaman çalacağını kayıp bir parça için ne yapacağını belirlerken çok dikkatli olmalıdır.

**Paket Kaybı:** İnternet telefonu uygulamamız tarafından üretilen UDP segmentlerinden birini ele alalım. UDP segmenti bir IP datagram içinde sarmalanır. Datagram ağ boyunca gezinirken, çıktı hatlarına erişmek için yönlendiricilerin tamponları üzerinden aktarılır. Göndericiden alıcıya doğru olan yollardaki bir veya birden fazla tamponun dolu olması veya IP datagramını kabul etmemesi mümkündür. Bu durumda IP datagramı atılır, alıcı uygulamaya hiçbir zaman ulaşmaz.

Kayıp paketleri UDP yerine TCP üzerinden göndererek eklenebilir. TCP'nin, hedefe varmayan paketleri yeniden ilettiğini hatırlayalım. Bununla birlikte yeniden iletim mekanizmaları genellikle İnternet telefonu gibi etkileşimli gerçek zamanlı ses uygulamaları için kabul edilemez olarak düşünülür, çünkü uçtan uca gecikmeleri artırır (Bolot ve Vega-Garcia 1996). Daha da ötesi TCP tıkanıklık kontrolü yüzünden, göndericideki iletim hızı, bir paket kaybının ardından alıcıdaki çekme hızından daha düşük bir hıza düşebilir. Bu nedenlerden dolayı var olan çoğu İnternet telefonu uygulaması, UDP üzerinden çalışır ve kayıp paketlerin yeniden iletimi ile ilgilenmez (Baset ve Schulzrinne 2006). UDP'nin, kullanıcı bir NAT veya UDP segmentlerini engelleyen bir güvenlik duvarı arkasında olmadığı müddetçe (bu durumda TCP kullanılır) Skype tarafından kullanıldığı bildirilmiştir.

Ancak paket kaybı, düşünülebileceği gibi bir felaket değildir. Aslında yüzde 1 ile 200 arasındaki paket kaybı sesin nasıl kodlandığına ve iletiildiğine ve kaybın alıcıda da nasıl gizlendiğine bağlı olarak hoş görülebilir. Örneğin ileri hata düzeltimi (FEC), paket kaybını gizlemeye yardım edebilir. Bununla birlikte gönderici ve alıcı arasındaki bir veya daha fazla hat ciddi şekilde tıkalıysa ve paket kaybı yüze 10 ile 20'yi aşmışsa (bu oranlar iyi hazırlanmış İnternet ağlarda nadiren gözlenmiş olsa da ) kabul edilebilir ses kalitesini yakalamak için

gerçekten yapılabilecek bir şey yoktur. Açıkçası en iyi çaba servisinin sınırları vardır.

**Uçtan Uca Gecikme:** Uçtan uca gecikme yönlendiricilerdeki, iletim, işleme ve kuyruklama gecikmelerinin; hatlardaki yayılma gecikmelerinin ve uç sistem işleme gecikmelerinin toplamıdır. İnternet telefonu gibi yüksek derecede etkileşimli ses uygulamaları için 150 milisaniyeden küçük, uçtan uca gecikmeler insanlar tarafından algılanamaz; 150 ve 400 milisaniye arası gecikmeler kabul edilebilir ama ideal değildir ve 400 milisaniyeyi geçen gecikmeler ses konuşmalarındaki etkileşimi ciddi biçimde engelleyebilir. Bir internet telefonu uygulamasının alıcı tarafı tipik olarak örneğin, 400 milisaniyeden fazla olan belirli bir eşikten fazla geciken paketleri almaz. Böylece eşikten daha fazla geciken paketler aslında kayıptır.

**Paket Seğirmesi:** Uçtan uca gecikmenin can alıcı bir bileşeni, yönlendiricilerdeki rastgele kuyruklama gecikmeleridir. Ağ içerisindeki bu değişen gecikmelerden dolayı bir paketin kaynakta üretilmesinden, alıcı tarafından alınmasına kadar geçen süre paketten dalgalanabilir. Bu olaya seğirme denir.

Bir örnek olarak İnternet telefonu uygulamamızda, bir konuşma faaliyeti içerisinde iki ardışık paket ele alalım. Gönderici ilk paketi gönderdikten 20 milisaniye sonra ikinci paketi gönderir. Ancak alıcıda bu paketlerin arasındaki boşluk 20 milisaniyeden fazla olabilir. Bunu görmek için ilk paketin bir yönlendiricideki, neredeyse boş kuyruğa vardığını ama ikinci paketin kuyruğa varmasından hemen önce, diğer kaynaklardan çok sayıda paketin aynı kuyruğa geldiğini varsayalım. İlk paket küçük bir kuyruklama gecikmesi yaşayacağı ve ikinci paket bu yönlendiricide daha büyük bir kuyruklama gecikmesi yaşayacağı için, ilk ve ikinci paketlerin arasında boşluk 20 milisaniyeden fazla olur. Ardışık paketler arasındaki boşluk ayrıca 20 milisaniyeden küçük da olabilir.

Alıcı seğirmenin varlığını yok sayarsa ve parçaları gelir gelmez çalarsa sonuç ses kalitesi alıcıda kolay anlaşılabilir hale gelebilir. Seğirmeyi ortada kaldırmak için , her parçaya dizi numarası ekleyerek , her parçaya bir zaman damgası ekleyerek ya da alıcı parçaların çalımını geciktirerek veya üçünü bir arada yaparak seğirmeden kaynaklı problem çözülebilir.

Gerçek zamanlı ses ve görüntü iletimi olan interaktif uygulamalar, genel olarak tek yönlü bir gecikme kullanıcı için gerçek zaman tepkilerinin bir izlenim vermek amacıyla kısa olması gerekmektedir. Tablo 2.1’de gerçek zamanlı iletim için tek yönlü gecikme sınırlarının durumları gösterilmiştir.

Tablo 2.1. Gerçek zamanlı iletimde tek yön limitleri

Tek yönlü iletim zamanı	Gecikmenin etkileri
0 to 150 ms	Kabul edilebilir
150 to 400 ms	Kabul edilebilir, etkilenme olur
400 ms and fazlası	Kabul edilemez

## 2.1. Rtp

Bir multimedya uygulamasının gönderici tarafının, ses/video parçalarını taşıma katmanına aktarmadan önce onlara başlık alanları ekler. Bu başlık alanları, dizi numaralarını ve zaman damgalarını içerir. Çoğu multimedya ağ uygulaması, dizi numaralarını ve zaman damgalarını kullanabileceği için ses/video verileri, dizi numarası, zaman damgası ve diğer potansiyel olarak kullanışlı alanları içeren standartlaşmış paket yapısına sahip olmak uygundur. RFC 3550’de tanımlanan RTP böyle bir standarttır. Günümüzde RTP, yüzlerce üründe ve araştırma prototipinde yaygın olarak uygulanmaktadır. SIP ve H.323 de dahil olmak üzere diğer önemli gerçek zaman etkileşimli protokoller için tamamlayıcıdır (Schulzrinne ve diğerleri, 2007).

RTP tipik olarak UDP üzerinden çalışır. Gönderici taraf bir medya parçasını, bir RTP paketi içerisinden sarmalar; ardından paketi, bir UDP segmenti içerisinden sarmalar ve ardından segmenti IP’ye verir. Alıcı taraf RTP paketini UDP segmentinden çıkarır, ardından medya parçasını RTP paketinden çıkarır ve ardından çözme ve görüntüleme için ortam oynatıcısına aktarır.

Bir örnek olarak sesi taşımak için RTP’nin kullanımını ele alalım. Ses kaydının 64 kbps’de PCM ile kodlandığını varsayalım. Uygulamanın kodlanmış verileri 20

milisaniyelik parçalar halinde yeni bir parçada 160 byte olarak topladığını varsayalım. Gönderici taraf ses verilerinin her parçasının önüne ses kodlamasının tipini, bir dizi numarasını ve bir zaman damgasını içeren bir RTP başlığı ekler. RTP başlığı normalde 12 byte'dır. RTP başlığı, ses parçası ile birlikte RTP paketini oluşturur. Ardından RTP paketi UDP soket arabirimine gönderilir. Alıcı tarafta uygulama, RTP paketini soket arabiriminden alır. Uygulama, RTP paketinden ses parçasına çıkarır ve uygun şekilde ses parçasının kodunu çözmek ve çalmak için RTP paketinin başlık alanları kullanılır.

RTP'nin verilerin zamanında teslimini veya diğer servis kalitesi garantisinden emin olmaya yarayan bir mekanizma sunmadığını vurgulamak gerekir, paket teslimi garantisini veya sıra dışı teslimin önlenmesini de garantilemez. Aslında RTP sarmalaması, sadece uç sistemlerde olur. Yönlendiriciler, RTP paketlerini taşıyan IP datagramları ile taşımayan IP datagramları arasında bir ayrım yapmaz.

RTP iki alt protokolü tanımlar:

Veri Transfer Protokolü: gerçek zamanlı çoklu ortam verisinin transferiyle ilgilidir. Bu protokol tarafından sağlanan bilgi senkronizasyon için tarih bilgisi, kaybolan paketlerin denetimi için sıra numarası ve verinin kodlanmış formatını gösteren payload formatını içerir.

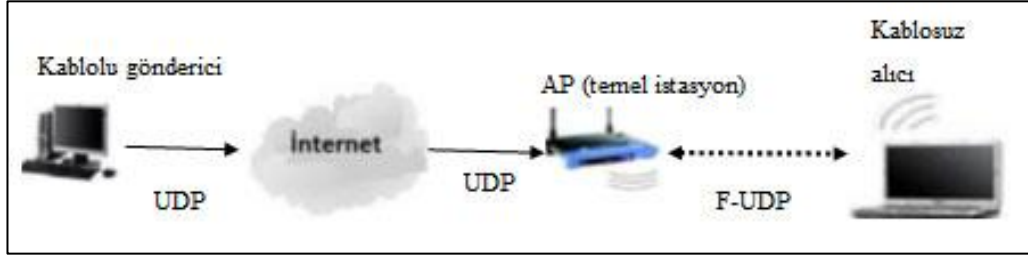
Gerçek Zamanlı Kontrol Protokolü: servis önceliği ile ilgili geribildirimler ve ortam streamleri arasında senkronizasyonu belirtmek için kullanılır.

### 3. UDP TEMELLİ İLETİM KATMANI PROTOKOL TASARIMI

İnternet ağlarında gerçek zaman etkileşimli ses/video haberleşmesinde, veri paketlerinin belirli zaman aralığında ve en az hata ile ulaşması önemlidir. Yine gerçek zaman etkileşiminde UDP protokolü, bağlantısız hizmet sunduğu için, tercih edilen bir protokoldür. TCP/IP protokol kümesi tasarımı kablolu ağlar için yapıldığı için, paket iletiminde kablosuz ağ tarafında tasarlanamayan ve istenmeyen durumlara neden olmaktadır. Bunlardan bir tanesi de, kablosuz ağdaki paket kayıp oranının kablolu ağlara göre göreceli olarak daha yüksek değerler almasıdır. Ulaşım katmanı protokolleri için TCP’de geriye dönük hata düzeltme olduğu için paketlerin ulaşmasında bir sorun olmamaktadır. Fakat UDP’de güvenilirlik yönünden bir mekanizmaya sahip olmadığından, hatalı ulaşan veriler için uygulama katmanında hata düzeltme kodlamaları kullanarak düzeltmeler yapmakta, ulaşmayan paketler için herhangi bir müdahale yapmamaktadırlar. Bu sebeple paket kaybının yüksek olduğu bir iletişimde gerçek zaman etkileşimli ses veya video iletişiminden istenen verim alınamamaktadır. Önerilen ağ tasarımında, gerçek zamanlı uygulamalar için çok yaygın olarak kullanılan UDP protokolünün, kablosuz ağlardaki paket kayıpları ve bozulması durumunda geriye doğru hata düzeltme yapılması öngörülmüştür Burada alıcı tarafa ulaşmayan paketlerin geriye dönük cevabı alınarak tespiti ve beslemesi yapılmıştır(Feeding UDP: F-UDP). Geleneksel UDP içinde kayıp olan paketlerin tekrar istenmesi, protokol dizaynı nedeniyle mümkün olamamaktadır. Bunun yerine uygulama katmanında hata düzeltme işlemleri yapılması ve ileriye doğru hata düzeltme çalışmaları yapılmış fakat geriye doğru hata düzeltme ile ilgili bir öneri bulunmamaktadır. Bu bağlamda, protokolün işleyişi ve paket başlığına eklenen bilgilerle UDP protokolünün TCPSack ve ağı bölme benzetimi sağlanarak paket kayıp oranının düşmesi sağlanmıştır.

Kablosuz-kablolu ağ tasarımında kablolu ağ tarafında gönderici ve geri bildirim yapılan ulaşmayan paket cevabının yayılım gecikme maliyeti göz önünde bulundurularak çift taraflı F-UDP kullanmaya gerek olmadığı düşünülmüştür.

Bunun yerine güvenilirlik yüzdesi düşük olan kablosuz ağ tarafında F-UDP prototipi kullanılmıştır.



Şekil 3.1. UDP temelli protokol tasarımı (F-UDP)

Burada kritik noktanın kablosuz erişim noktasının ne kadar süre ile veya boyutta tamponlama yapması gerektiği ve paketlerin geliş sıralarına göre paket numarası vermesidir. Paket numarasının yanı sıra dikkat edilmesi gereken husus, iki uç arasındaki noktadan noktaya haberleşme yerine, erişim noktası ile yalıtılmış iki farklı özelliği olan ağın bölünmüş olması durumudur. TCP/IP'nin ulaşım katmanındaki uçtan uca haberleşme yerine bölünmüş iki farklı iletim şekli ortaya çıkmıştır.

### 3.1. Geri Beslemeli UDP (F-UDP-Feedback base UDP)

Gerçek zamanlı uygulamalar için kullanılan UDP protokol tasarımı, kablolu ağlar için tasarlandığı için, kablosuz ağlardaki iletimdeki paket kayıp oranının kablolu ağa göre yüksek olması nedeniyle istenmeye durumlar oluşmaktadır. Fiziksel yapısından dolayı kablosuz ağdaki paket kayıp oranının fazla olması, UDP gibi güvenilirlik mekanizması olmayan bir protokolü çaresiz bırakabilir. Hata oranını düşürmek için ileriye dönük hata düzeltme yöntemleri kullanılmış olsa da, ulaşmayan paketler veya düzeltilmeyecek kadar çok hatalı paket ulaşması durumunda ileriye doğru hata düzeltme metotlarının yanında, geriye dönük hata düzeltme metotlarının da kullanılabilirliği araştırılmalıdır. Bu sebeple tıpkı TCP'deki gibi geriye dönük hata düzeltme işleminin yapılabileceği metot araştırılmıştır. Fakat geriye dönük hata düzeltme işlemindeki gecikme ve bağlantı kurma işleminin alacağı süre göz önünde bulundurulduğunda, gerçek zamanlı uygulamalar için ek bekleme süresine neden olacağı açıktır. Uçtan uca bağlantı yerine hata oranının yüksek olduğu tarafta yani kablosuz tarafta bir çözüm



geliştirilirse, gecikmenin daha az olacağı görülmektedir. Kablosuz ağda geriye dönük hata düzeltme için hem bağlantı kurmadan UDP uyarlamasının yapılması hem de gerçek zamanlı uygulamalar için bekleme süresi eşliğini geçmemesi için bir tasarım düşünülmüştür. F-UDP, kablosuz ağ tarafında çalışan, UDP alıcısını kablosuz uçta besleyerek, kablosuz ağdaki paket kayıp oranını düşürmeyi amaçlayan bir tasarımdır. F-UDP tasarımı yaparken bir taraftan UDP alıcı tarafında geri besleme yaparken, diğer taraftan gerçek zaman uygulaması olduğu göz ardı edilmemelidir. Bu sebeple besleme süresi gerçek zaman için tanımlanan eşik değerini geçmemesi dikkate alınmıştır.

F-UDP tasarımında kablosuz ağa giriş ucunda geri besleme mekanizması tasarlarken, dikkate alınması gereken en önemli husus bekleme süresi olmuştur. F-UDP geri beslemeyi ne kadar süre yapacağı, tasarımı oluştururken en önemli parametre olarak karşımıza çıkmıştır. Bunun için 3 farklı bekleme periyodu (timeout) tasarlanmış ve simüle edilmiştir:

- Paket saydırarak
- Sabit bir değer vererek
- RTT hesaplanarak

bekleme süreleri karşılaştırılmıştır.

F-UDP'nin gönderici ve alıcı taraftaki sözde kodları Tablo 3.1 ve Tablo 3.2'de gösterilmiştir.

Göndericiye gelen paket için F-UDP protokolüne gelen UDP paketleri aynı alıcı için oluşturulan soket tarafından hizmet açılır ve böylece iki uç arasındaki F-UDP iletiminde gerçekleşmesi olası olaylar kontrol edilir. Eğer veri paketi geliyorsa, sıra numarası yerleştirilir ve paket tampona atılarak zamanlayıcı tutulur. Eğer zaman aşımı olayı gerçekleşirse çöpe atılır. Eğer F-UDP hizmetine NACK paketi gelirse, NACK numarasına göre tamponda paket varsa tekrar gönderilir(Şekil 3.2).

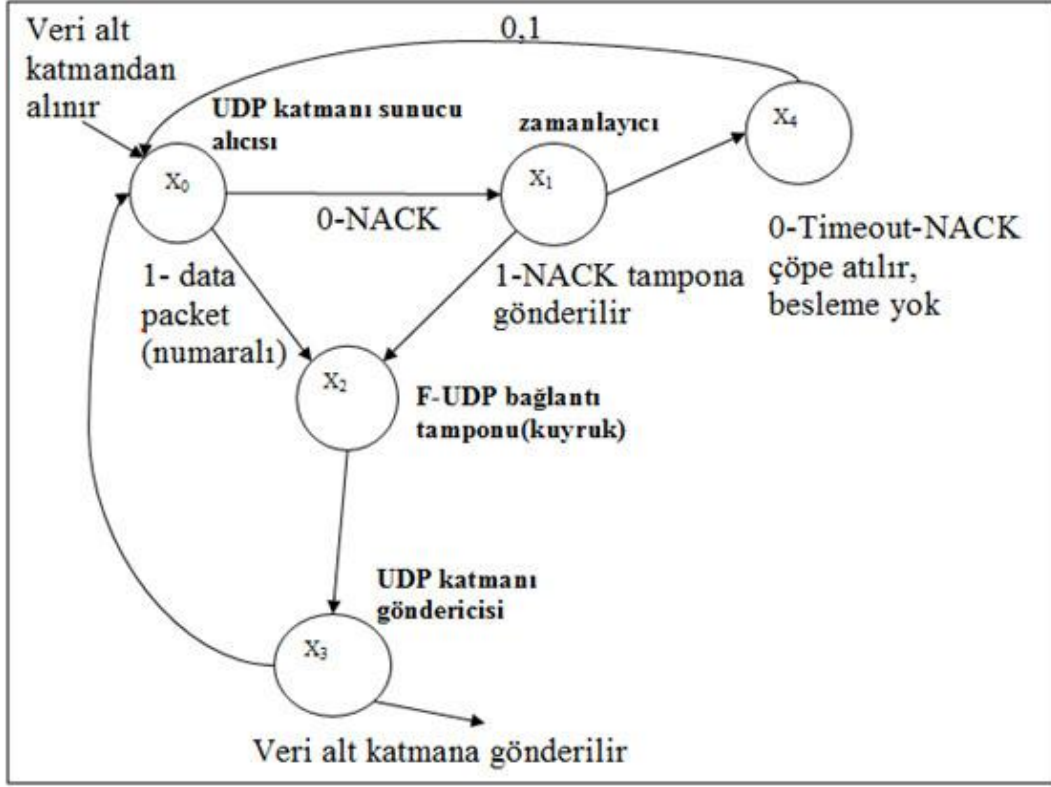
<p><b>Gönderici:</b></p> <p>SiradakiSayi=İlksıradakinumara</p> <p>Loop (sonsuz)</p> <p>{</p> <p>Switch(Olay)</p>
<p>Olay: veriler yukarıdaki uygulamadan alındı</p> <p>Y.SiradakiSayi dizi numarasına sahip UDP segmenti oluştur</p> <p>İf zamanlayıcı çalışmıyorsa</p> <p style="padding-left: 40px;">Zamanlayıcı çalıştır</p> <p>Y.Segmenti Y tampona at</p> <p>Y.Segmenti IP katmanına gönder</p> <p>Y.SiradakiSayi=Y.SiradakiSayi+length(data)</p>
<p>Olay: zamanlayıcı zaman aşımı</p> <p>İf Y.Segment.SiradakiSayi NACK mesajı gelmedi</p> <p>Segmenti tampondan çıkar</p>
<p>Olay: Y ACK alanı değerine sahip NACK alındı</p> <p>İf zamanlayıcı zaman aşımı&lt;timeout</p> <p>Y numaralı Segmenti IP katmanına gönder</p> <p>} //döngü sonu</p>

Şekil 3.2. F-UDP temel istasyondaki sözde kodu

Alıcı tarafta F-UDP hizmeti alan uç, hizmet için oluşturduğu tampona paketleri alır. Tampona paket numarasına göre aldığı için eğer, sıralı gelmeyen paket olayı gerçekleşirse NACK paketini temel istasyona göndererek tekrar istekte bulunur. Eğer zaman aşımı olayı gerçekleşirse gelmeyen paketi beklemekten vazgeçer ve tüm tamponu üst katmana gönderir (Şekil 3.3.).

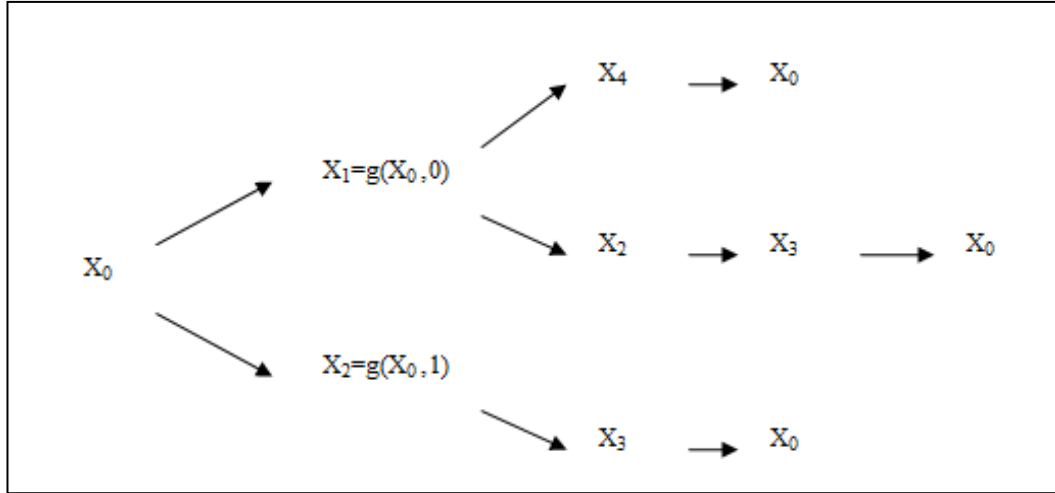
<pre>SiradakiSayi=İlksiradakinumara  Loop (sonsuz) { Switch(Olay)</pre>
<pre>Olay: veriler IP katmanından alındı      If X.UDPsegment =X.SiradakiSayi          Segmenti yukarıdaki uygulamaya gönder      Else If X.Zamanlayıcı&lt; X.SiradakiSayi.timeout          {              UDP segment'i tampona at              Göndericiye SiradakiSayi NACK gönder          }</pre>
<pre>Olay: zamanaşımı      X.Zamanlayıcı&gt;timeout      Tampondaki tüm segmentleri yukarı uygulamaya gönder  }</pre>

Şekil 3.3. F-UDP alıcı tarafın sözde kodu



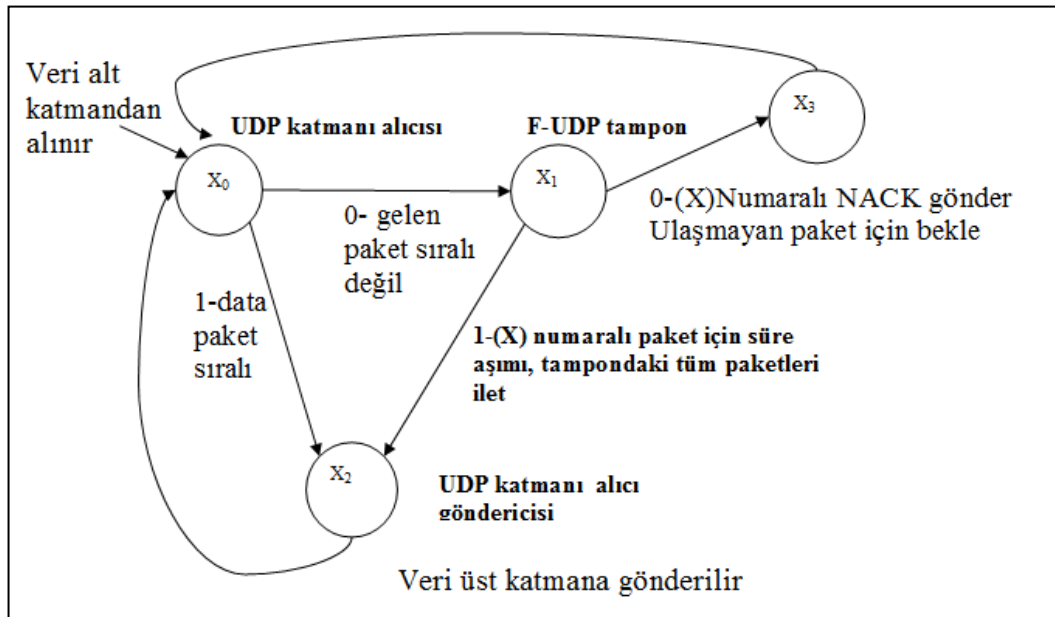
Şekil 3.4. F-UDP göndericisi için durum diyagramı

F-UDP gönderici durum diagramında  $X_0$  durumu, protokolün kablosuz ağa giriş noktasındaki UDP başlıklı gelen paketin protokol katmanına girişini gösterir. F-UDP gelen paket için veri paketi veya ACK paketi olup olmadığını tespit ettikten sonra durum  $X_1$  veya  $X_2$  durumuna geçer.  $X_1$  durumunda paket için tamponda zamanlayıcı kontrol edilir. Zamanlayıcı sona ermişse  $X_4$  durumuna geçer ve ACK çöpe atılır, sona ermemişse ACK tampona gönderilir.  $X_2$  durumunda ise paket tampondan alt katmana gönderilme durumuna iletilir.  $X_3$  durumunda veri alt katmana iletilir ve  $X_0$  durumuna geçer (Şekil 3.4.).



Şekil 3.5. F-UDP göndericisi durum diyagramı doğruluk analizi

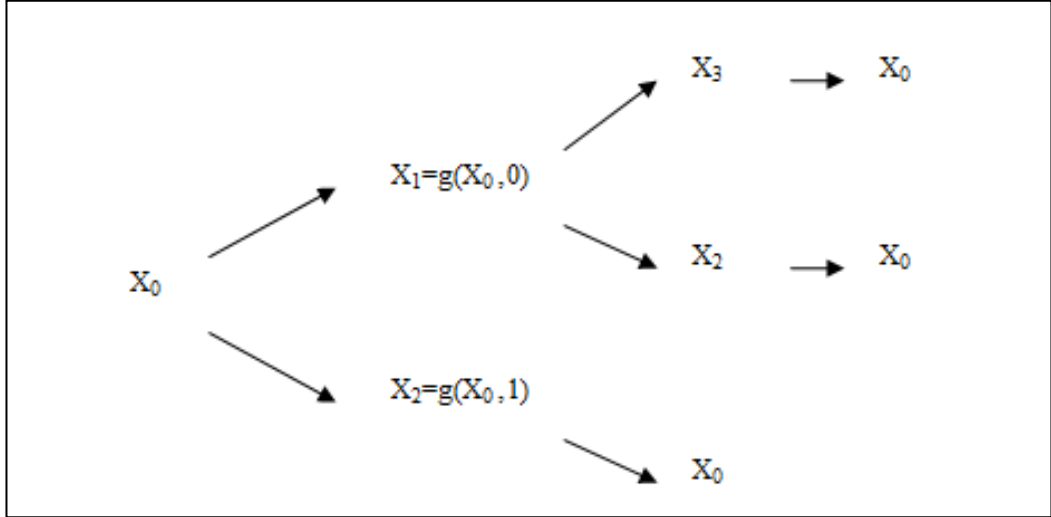
F-UDP göndericisinin durum diagramı için tüm durumlara ulaşma şekli görülmektedir. Analiz sonucunda tüm durumlara hangi durumlardan ulaşılacağı görülmektedir (Şekil 3.5).



Şekil 3.6. F-UDP alıcısı için durum diyagramı

F-UDP alıcı durum diagramında  $X_0$  durumu, protokolün kablosuz alıcıya giriş noktasındaki UDP başlıklı gelen paketen protokol katmanına girişini gösterir. F-UDP gelen paket için sıralı veya yanlış sırada olup olmadığını tespit ettikten sonra durum  $X_1$  veya  $X_2$  durumuna geçer.  $X_1$  durumunda paket yanlış sırada gelmiştir ve bundan sonra gelecek olan tüm paketler tampona atılır ve bekletilir..

Zamanlayıcı sona ermişse  $X_2$  durumuna geçilir, tüm bekletilen paketler üst katmana iletilir ve tekrar  $X_0$  durumuna geçer. Zamanlayıcı sona ermemişse  $X_3$  durumunda her yeni paket geldiğinde ACK gönderilir (Şekil 3.6.). Şekil 3.7.'te ise durum diyagramı için erişebilirlik analizi gösterilmiştir.

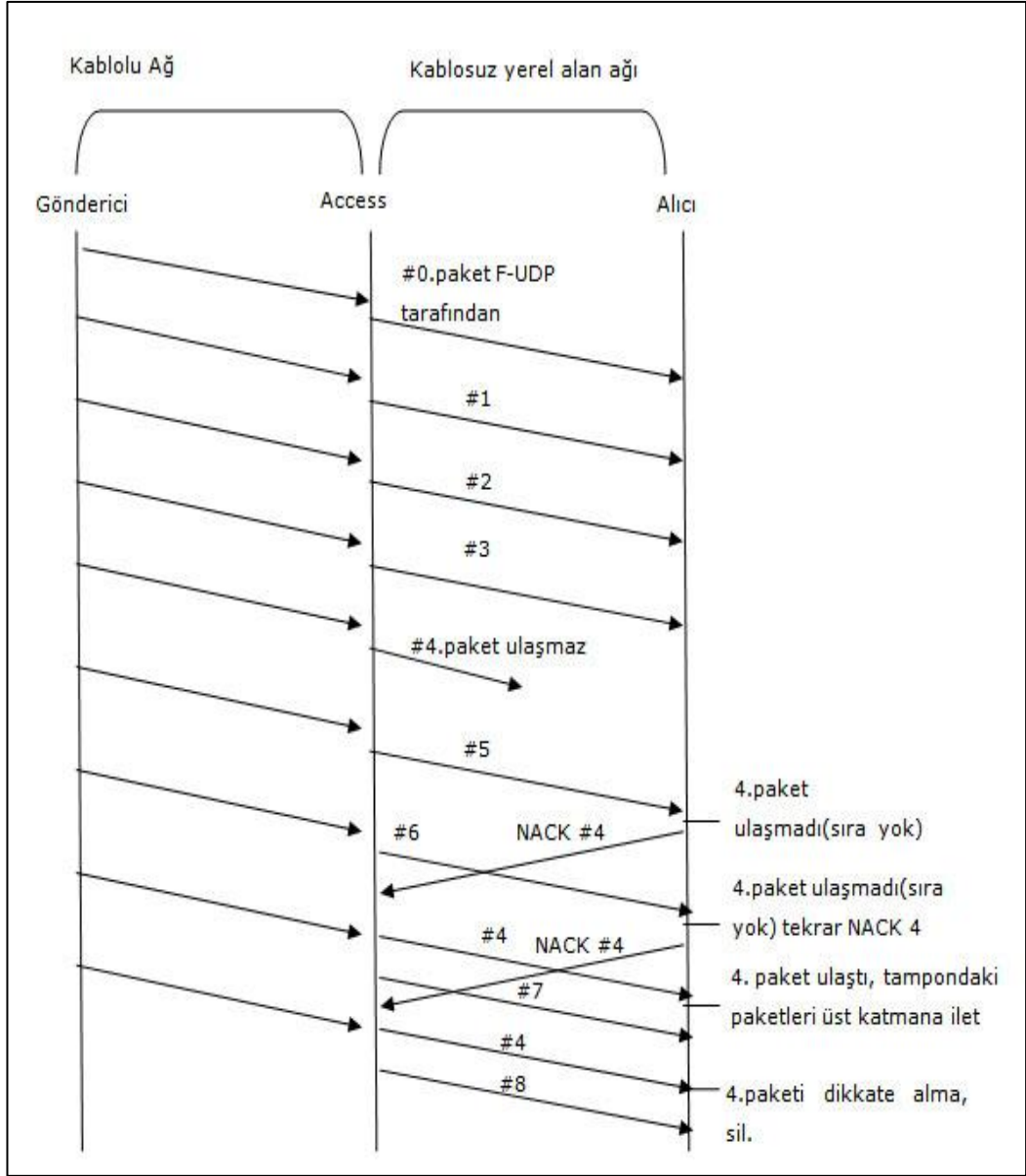


Şekil 3.7. F-UDP alıcısı durum diyagramı doğruluk analizi

Alıcı ve erişim noktasındaki gönderici için paketlerin bekleme süresi için 3 farklı metot düşünülmüştür ve bu metotlar için gecikme süreleri ve paket kayıp oranları simüle edilip sonuçları gözlenmiştir. Bu metotlar, paketlerin sayma, RTT ve zaman sabitleridir. Bahsedilen metotlar birbirleri kıyaslanarak en iyi sonuçlara göre PSNR değerleri hesaplanmıştır.

Yukarıdaki kullanılan algoritma için aynı zamanda, alıcı tarafın erişim noktası, göndericinin kablosuz ağdaki bir bilgisayar olması durumunda aynı algoritma geçerli varsayılmıştır.

Algoritma için oluşabilecek durumlara göre senaryolar için çıkarılan zaman diagramları Şekil 3.8 ve Şekil 3.9'da gösterilmiştir.

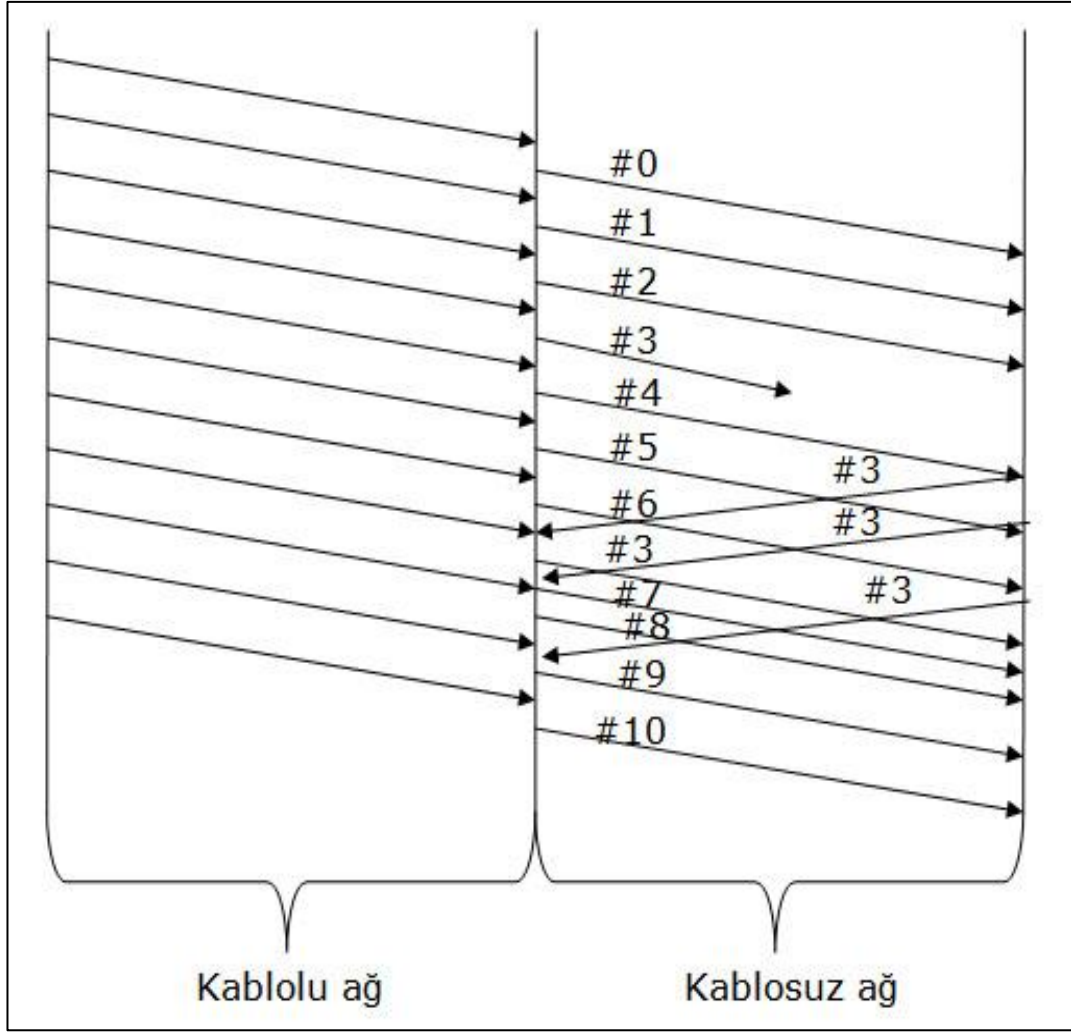


Şekil 3.8. F-UDP için kablolu-kablosuz ağ paket akış zaman diyagramı

F-UDP için kablolu-kablosuz ağ paket akış zaman diyagramında(Şekil 3.8.) kablosuz ağa girişte çalışan F-UDP'de AP'de yapılan tamponlama ile birlikte kablolu uçtaki göndericiden gelen paketlere sıra numarası yerleştirilir. Gelen pakete verilen sıra numaralarına göre akışta bir problem olup olmadığı tespiti yapılır. Yukarıdaki örnekte 4 numaralı paketin alıcıya ulaşmama durumu, 5 numaralı paket geldikten sonra tespit edilir. Tespit edildikten sonra sadece 4 numaralı paket için ulaşmadı bilgisi AP'ye tekrar gönderilir. 6 numaralı paket geldiğinde istek tekrarlanır. 4 numaralı paket ulaştıktan sonra akış kaldığı yerden devam eder. 5 ve 6 numaralı paketler bekletilen tampondan üst katmana iletilir.



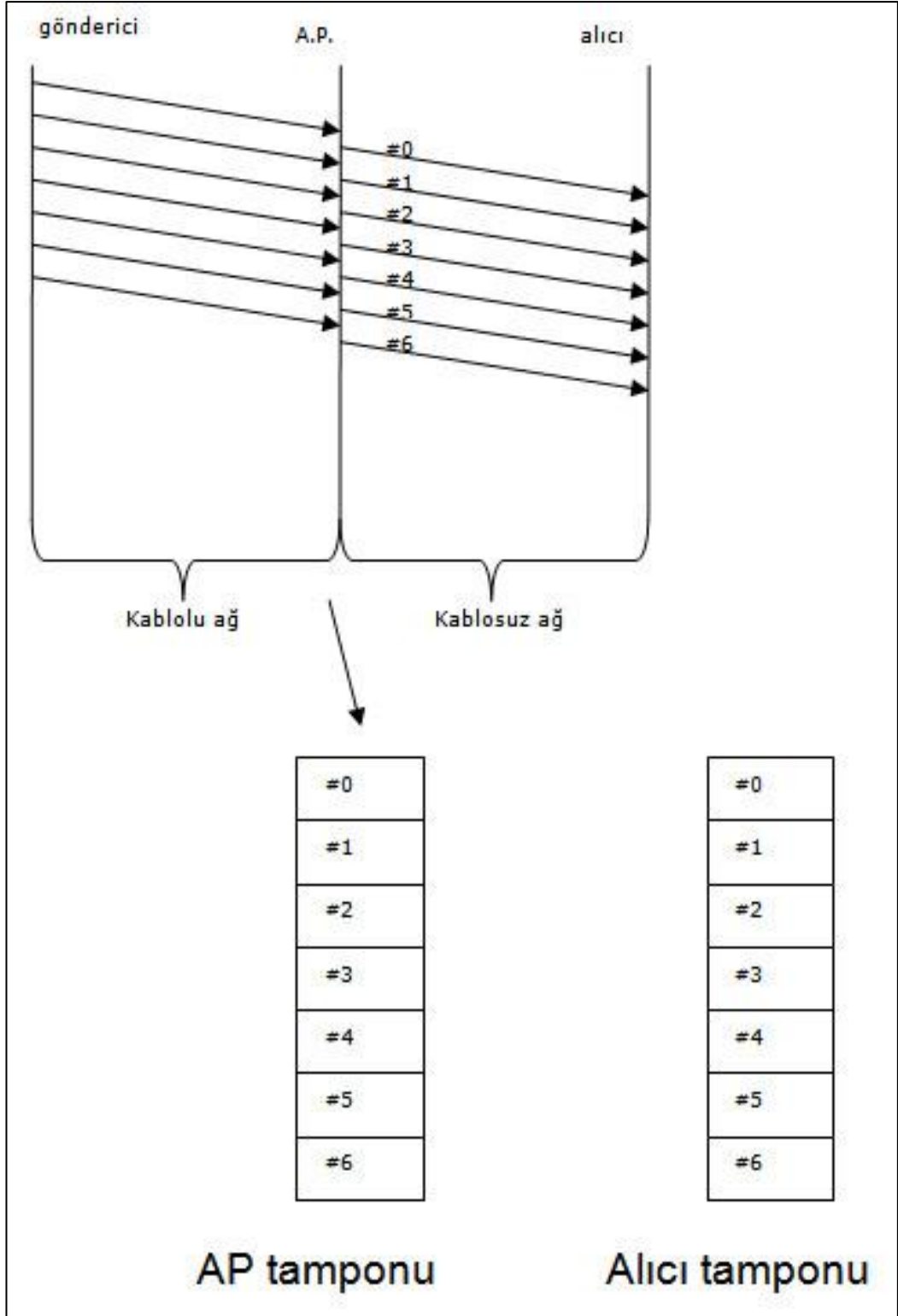




Şekil 3.10. RTT ve sabit zaman için zaman diagramı- hatalı ulaşımda tekrar istek

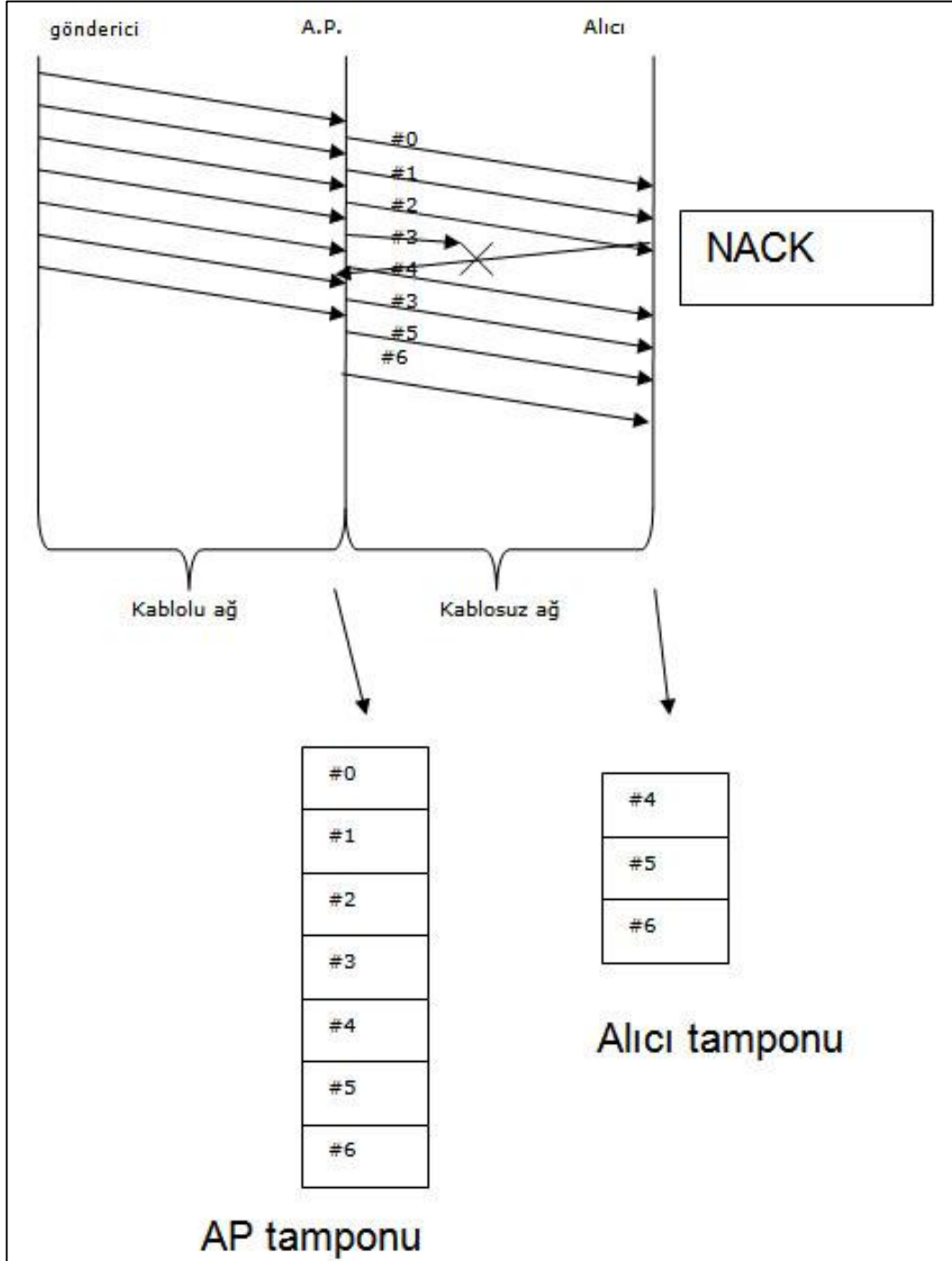
Bekleme süresi için RTT veya sabit bir değer kullanıldığında, gönderici belirlenen zaman kadar paketlerin kopyalarını tamponda tutar. Zaman aşımı olduğunda kopyalar tampondan çıkartılır. Alıcıda ulaşmayan bir paket geldiğinde, sonraki numaralandırılmış paketler, belirlenen zaman(timeout) kadar paketleri tamponda bekletir. Tampona atılan her paket için gelmeyen paket için göndericide istekte bulunur. Gönderici erişim noktası veya kablosuz uç birim olabilir. RTT ve sabit zaman için zaman diagramı Şekil 3.10'de gösterilmiştir.

Bekleme süresi sabit olan ve RTT'li tasarım için yine zaman akış çizelgesi ve tamponların durumu Şekil 3.11, Şekil 3.12 ve Şekil 3.13'de gösterilmiştir.



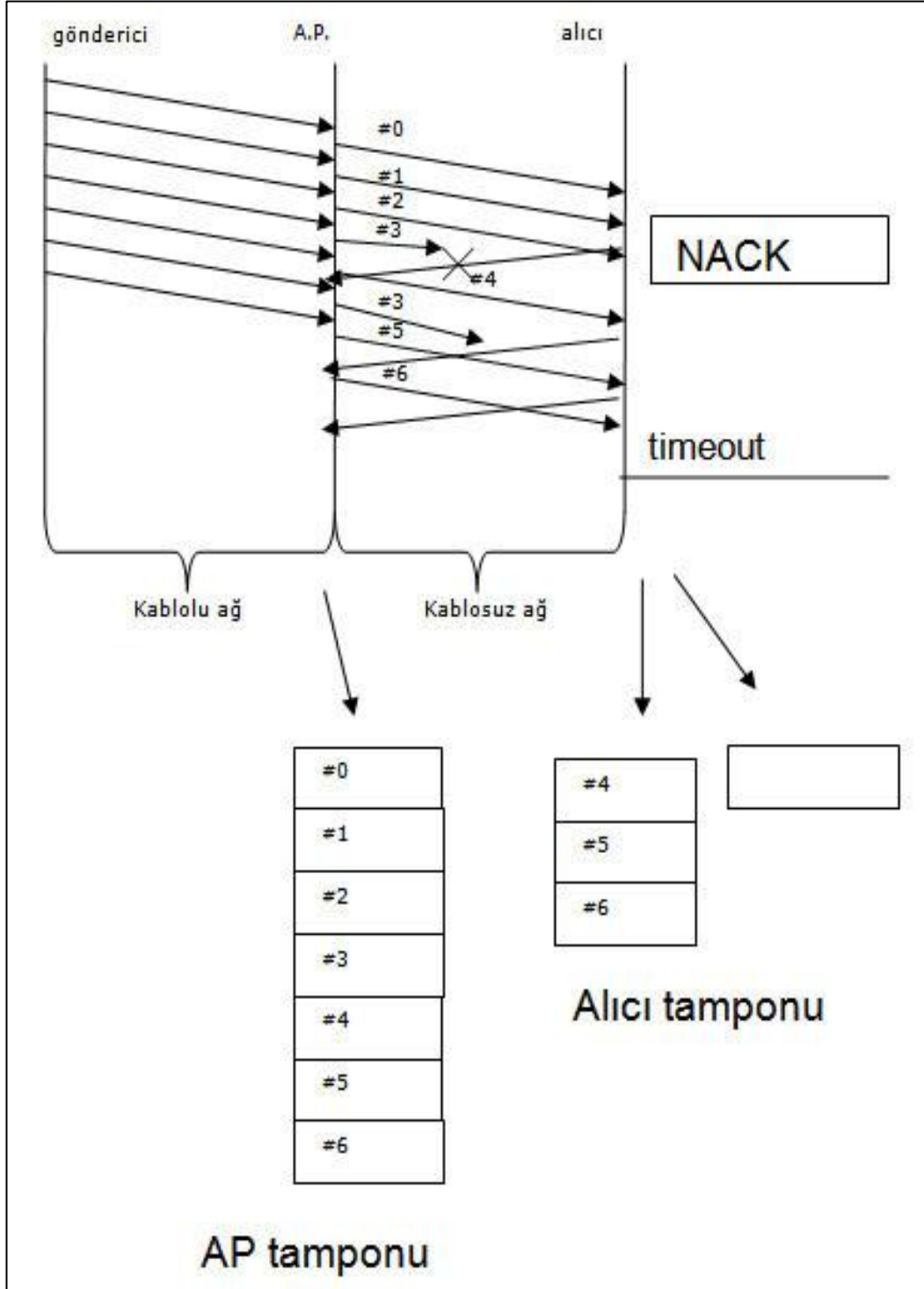
Şekil 3.11. RTT ve sabit zaman tampon görünümü

Yukarıdaki şekilde, herhangi bir paket kaybı olmadığı süreçte AP ve alıcı tamponlarında paketlerin tamponlarda tutulduğu görülmektedir. İletimde sorun yaşanmadığı için paketler ulaştığında tampondan üst katmana gönderilmektedir.



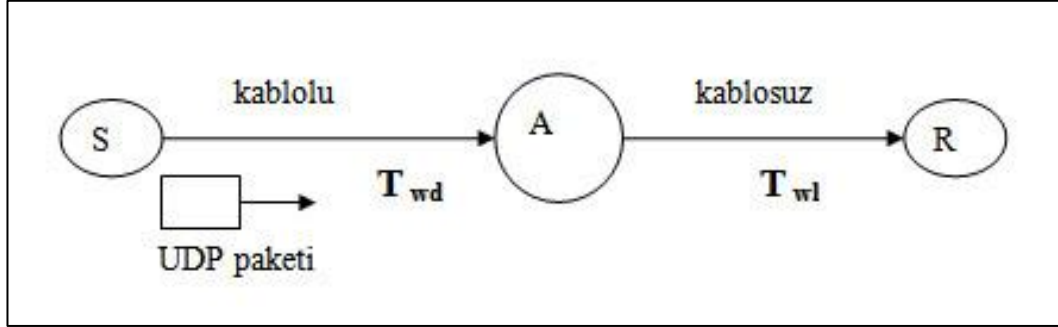
Şekil 3.12. Hatalı paket ulaştığında RTT ve sabit zaman tampon görünümü

Yukarıdaki şekilde 3 numaralı paket alıcıya ulaşmaz, 4, 5, ve 6. paketler ulaştığında ise tamponda bekletilerek istekte bulunan 3 numaralı paketin gelmesi bekletilir. Bekletilme süresi yine RTT veya sabit zamanla sınırlıdır.



Şekil 3.13. Hatalı paket ve zaman aşımı için RTT ve sabit zaman için tampon

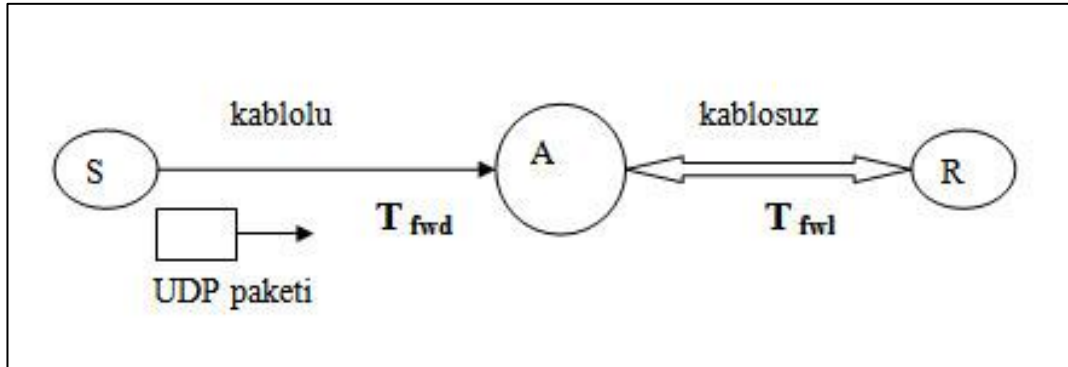
Yukarıdaki şekilde tamponlamada paketin ulaşmaz ve sınır sürenin(timeout) biterse son durumu gösterilmiştir. 3 numaralı paket timeout süreci içerisinde alıcıya ulaşmamıştır. Bu süreç içerisinde 4, 5 ve 6 numaralı paketler alıcı tamponunda bekletilir ve bekleme süreleri bittiğinde ise üst katmana gönderilirler.



Şekil 3.14. Geleneksel UDP için iletim zamanı sembolleri

UDP için bir paket için geçen süre kablolu ve kablosuz olarak iki bölüme ayrılmıştır.

$$T_{total} = T_{wd} + T_{wl} \quad (3.1)$$



Şekil 3.15. Geleneksel UDP için iletim zamanı sembolleri

F-UDP için yine aynı şekilde bir paket için alıcıya ulaşma süresi kablolu ve kablosuz olarak ikiye ayrılmış, sembolleri aşağıda verilmiştir.

$$T_{ftotal} = T_{fwd} + T_{fwl} \quad (3.2)$$

F-UDP için paket kaybını engellemek için odaklandığımız zaman kablosuz ağda geçen işlem zamanıdır( $T_{fwl}$ ).  $T_{wd}$  ve  $T_{fwd}$  eşit olduğu için gerçek zaman limitlerini geçmemek için kontrol edilecek süre  $T_{fwl}$  olacaktır. F-UDP için

kablosuz taraftaki işlem zamanı UDP'ye göre daha büyük değer olacağı görülmektedir. Timeout bekleme süresi ve paket numaralandırma ve hizmet verme süreçleri ek gecikmeye neden olacaktır.

$T_{nmb}$ =F-UDP paketler numara yerleştirme işlemi için oluşan gecikme

$T_{feed}$ = NACK kullanarak geriye doğru hata düzeltme işlemi için gecikme

$T_{buffer}$ = tampona paketi atma ve almak için gereken zaman

$$T_{fwl} = T_{nmb} + n \times T_{feed} + T_{buffer} \quad (3.3)$$

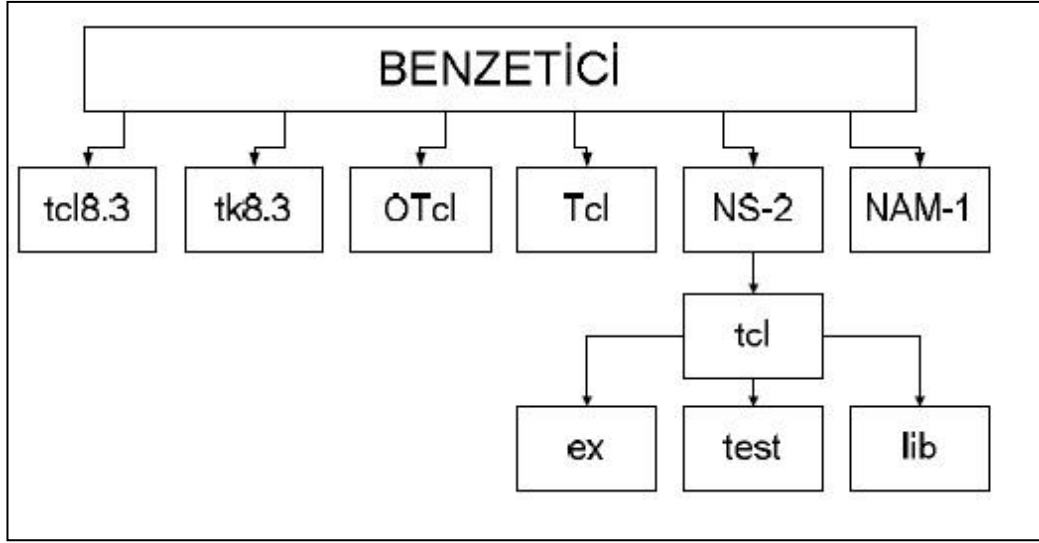
Bu denklemde n, gerçek zamanlı iletim için sınır değer(timeout) ve paket kaybı olmak üzere iki parametreye bağlıdır.

Gerçek zamanlı uygulamalar için kullanılan geleneksel UDP haberleşmesi ile öneride bulunan F-UDP tasarımın karşılaştırılması amacıyla NS-2 simulatörü kurulmuş örnekler yapılmış ve ağ tasarımı kurulmuştur.

### **3.2. Simülasyon**

NS2 programı bir ağ ortamı oluşturmak ve oluşturulan ağ ortamının bağlantı analizlerini yapmak ve bunu sanal ortamda görebilmeyi sağlar. NS2 birçok kablolu ve kablosuz ağ benzetimlerini desteklemektedir. NS2, Unix, Linux, SunOS ve Solaris gibi Unix türevleri olan platformlarda geliştirilmektedir. C++ programlama diliyle yazılmış olup komut ve konfigürasyon arayüzü olarak OTcl betik dilini kullanır.

NS2 önemli bileşenleri ağ animasyon ortamı (Network Animation Media-NAM) ve araç komut dilidir (Tool Command Language-TCL). NS2 ile yazılan bir betiğin animasyonunu görmemiz için NAM programı kullanılır. TCL ise OTcl betiklerinin derlendiği bir derleme programıdır (Şekil 3.16).



Şekil 3.16. NS simülör dosya sistemi

Benzetim ile ilgili yama ve eklentiler sadece benzeticinin çekirdeği olan NS2 kısmına yapılır. Tcl dizininin altındaki lib dizini derlenen TCL kodlarının saklandığı yerdir. Test dizini NS2 kurulduktan sonra kurulumun doğru olup olmadığını testetmek için gerekli olan bazı test betiklerinin bulunduğu yerdir. ex dizini içinde ise NS2 örnek betikler vardır.

EvalVid gerçek veya benzetim yapılmış ağlarda video iletimi kalitesinin değerlendirilmesi için geliştirilmiş bir araçtır.

Tepe İşaret Gürültü Oranı (Peak Signal to Noise ratio- PSNR): PSNR video iletim kalitesi başarımlı ölçütüdür. Kaynak görüntü ( $S$ ) ve alınan görüntünün ( $D$ ) parlaklık (luminance) değerleri ( $Y$ ) arasındaki farka dayanır ve aşağıdaki eşitlik ile verilir.

$$PSNR(n)db = 20 \left( \frac{V_{peak}}{\sqrt{\frac{1}{N_{col}N_{row}} \sum_{i=0}^{N_{col}} \sum_{j=0}^{N_{row}} [Y_s(n,i,j) - Y_d(n,i,j)]^2}} \right) \quad (3.4)$$

Burada  $V_{peak}=2^k-1$  dir ve  $k$ , video görüntüsündeki her pikselin parlaklık değerinin kaç bitle temsil edildiğidir. PSNR yeniden alıcıda inşa edilen video dosyası ile orijinal video dosyası arasındaki farkı ölçer.  $N$  ve  $M$ , kaynak ve alınan görüntülerin yatay ve dikey piksel sayılarıdır.

NS2-EvalVid: NS2 ve EvalVid'in birlikte çalışması Şekil 3.14'de görülmektedir. EvalVid'in çalışma yapısı ise aşağıdaki şekilde tanımlanmıştır. Video dosyası VE tarafından sıkıştırılmış videoya dönüştürülür. VS sıkıştırılmış videoyu okur ve UDP paketler olarak ağ üzerinden gönderir. Aynı zamanda VS, ET video değerlendirmesi için video izleme dosyasını oluşturur. Son olarak, video çerçeveleri video çözücü tarafından YUV formatına çözülüp, kullanıcıya sunulur. Her bir video paketinin tipi ve zaman-pulu, göndericiye ve alıcı izleme dosyasına video kalite değerlendirmesi için kayıt edilir. İzleme dosyalarının karşılaştırılmasıyla ET çerçeve kaybı ve gecikmesi raporlarını oluşturur. PSNR programı her bir çerçeve için PSNR değeri hesaplar.

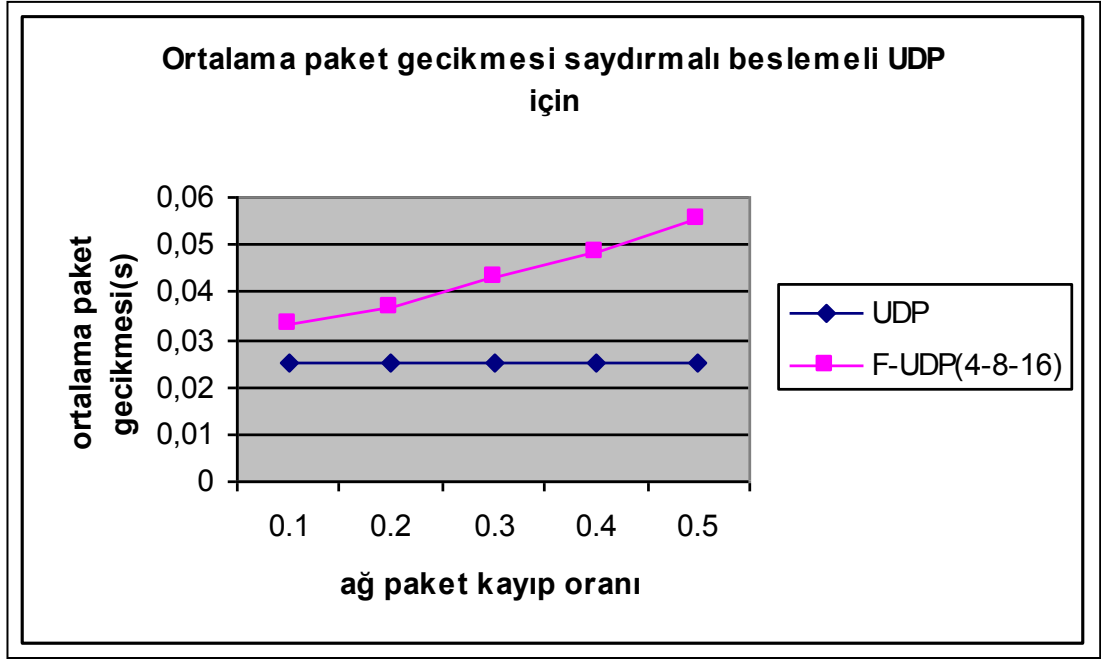
### **3.3. Simülasyon Sonuçları**

Önerilen F-UDP için, paketlerin tamponda bekletilmesi ve geri bildirim için 2 farklı yöntem ve bu farklı yöntemler içerisinde bekleme sürelerine göre paket kayıp oranları ve paket gecikmeleri ile ilgili simülasyon ölçümleri yapılmıştır. Gecikmeler saniye cinsinden hesaplanmıştır.

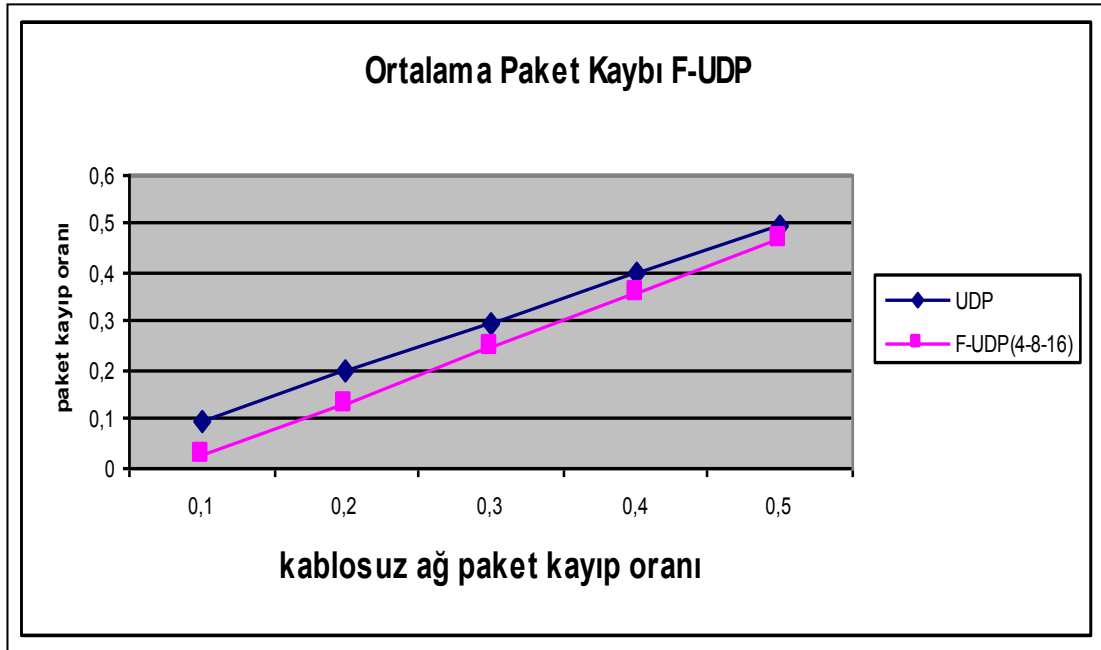
Benzetimi yapmak için 1 adet kablosuz haberleşme yapmayan video sunucusu, 1 kablosuz erişim noktası, 1 adet kablosuz video alıcı yerleştirilmiştir. Video sunucusu ile kablosuz erişim noktası arası erişim hızı 10 Mbps, kablosuz ağ erişim hızı yine 10 Mbps ve kablolu ağ gecikmesi 10 milisaniye olarak tasarlanmıştır.

Bu ölçümlerden birincisinde, paket saydırmaya göre tamponlama için ortalama paket kayıp oranı ve paket gecikmesi (latency) aşağıda grafikte verilmiştir.





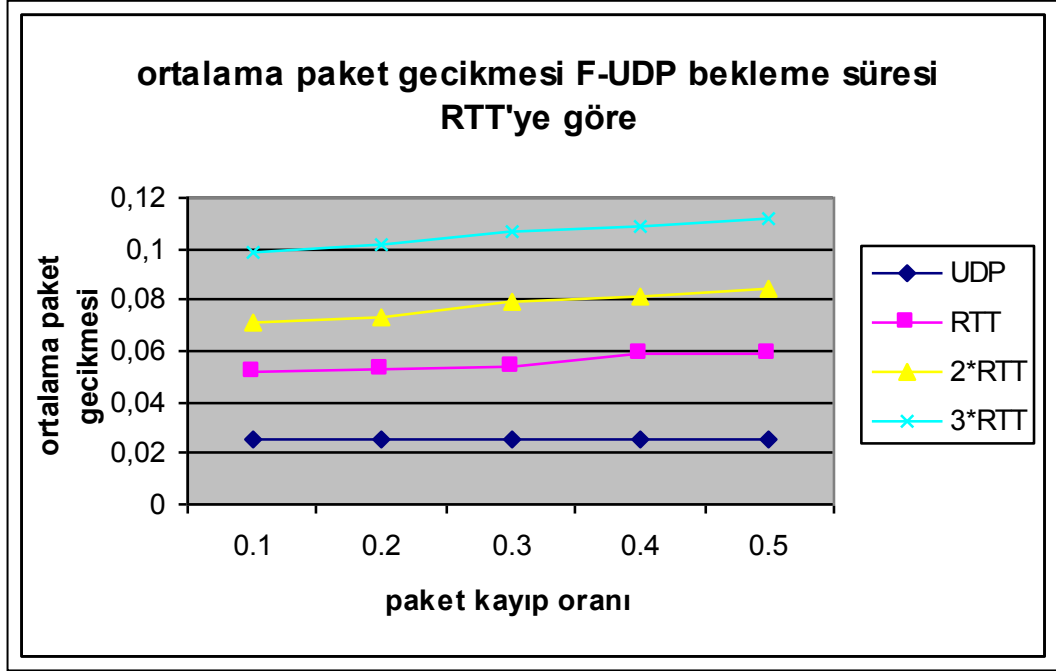
Şekil 3.17. F-UDP paket saydırma ile tamponlama için paket gecikmesi



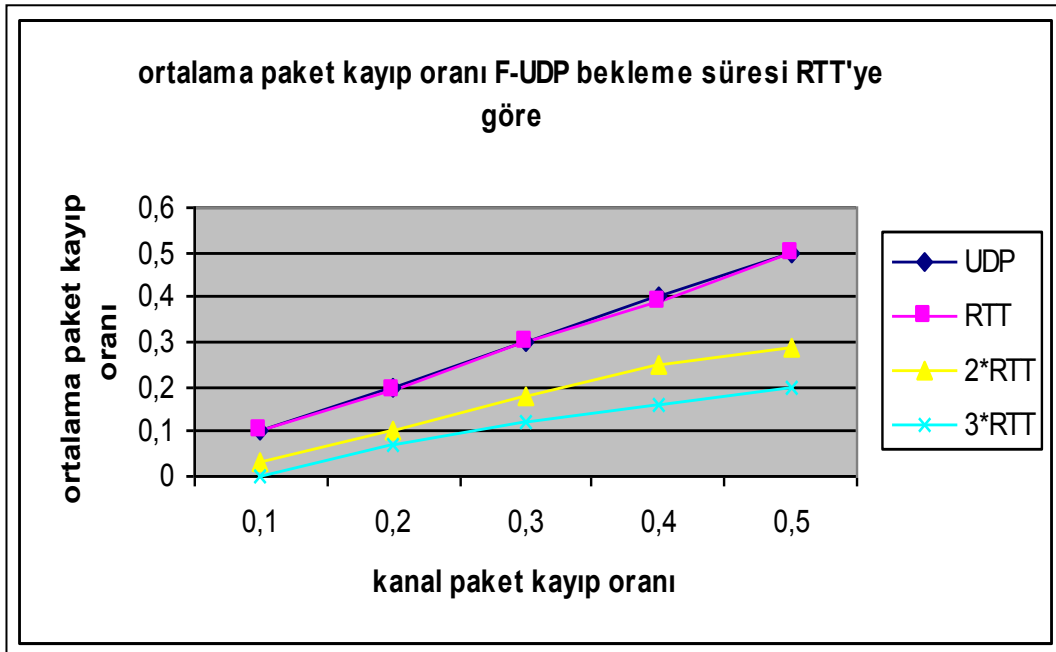
Şekil 3.18. F-UDP paket saydırma ile tamponlama için paket kaybı

Bekleme süresi RTT alındığında, paket gecikmesi ve ortalama kayıp için grafik aşağıdaki gibidir. RTT hesaplaması için ilk önce herhangi anlık zamanda TCP için RTT kullanılması düşünülmüştür fakat ihtimaller dahilinde TCP bağlantısı olmadığında herhangi bir hesaplama olamayacağı göz önünde bulundurularak IP katmanında “ping” isteğinde bulunularak RTT hesaplaması yapılmış olup, zaman

hesaplanmıştır. Fakat zamanın anlık değeri değişebileceği için ölçüm belli periyotlarla tekrarlanması gerektir. Bu sebeple bu periyod yine son RTT olarak belirlenmiştir. Her son RTT zaman aralığında tekrar ping komutu işletilerek RTT hesaplanmıştır. Bu yöntem haliyle daha fazla gecikmeye neden olmuştur.

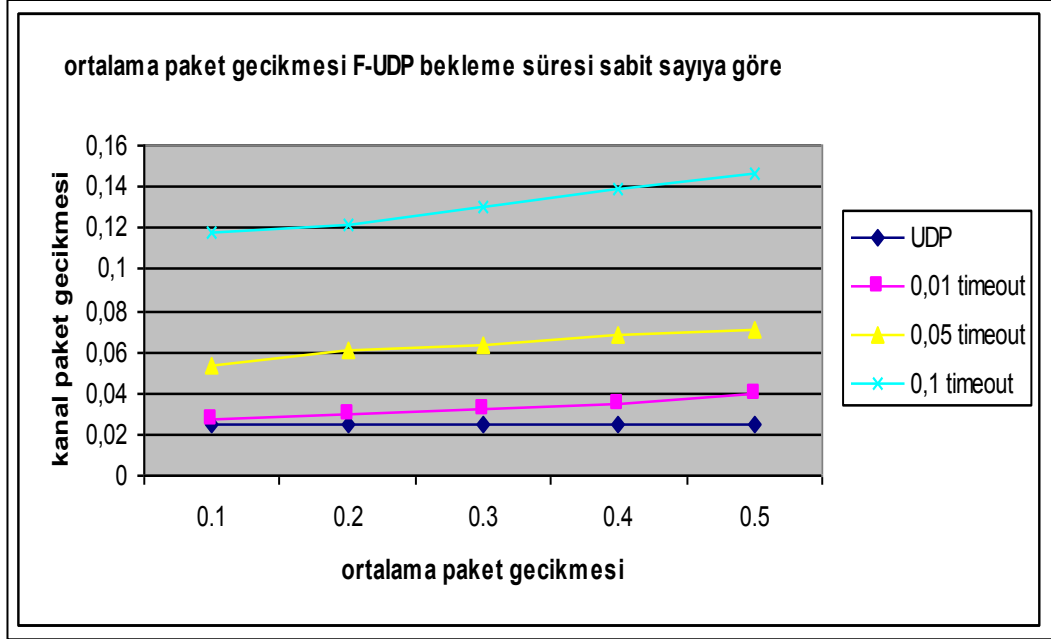


Şekil 3.19. F-UDP için tampon bekleme süresi RTT için paket gecikmesi

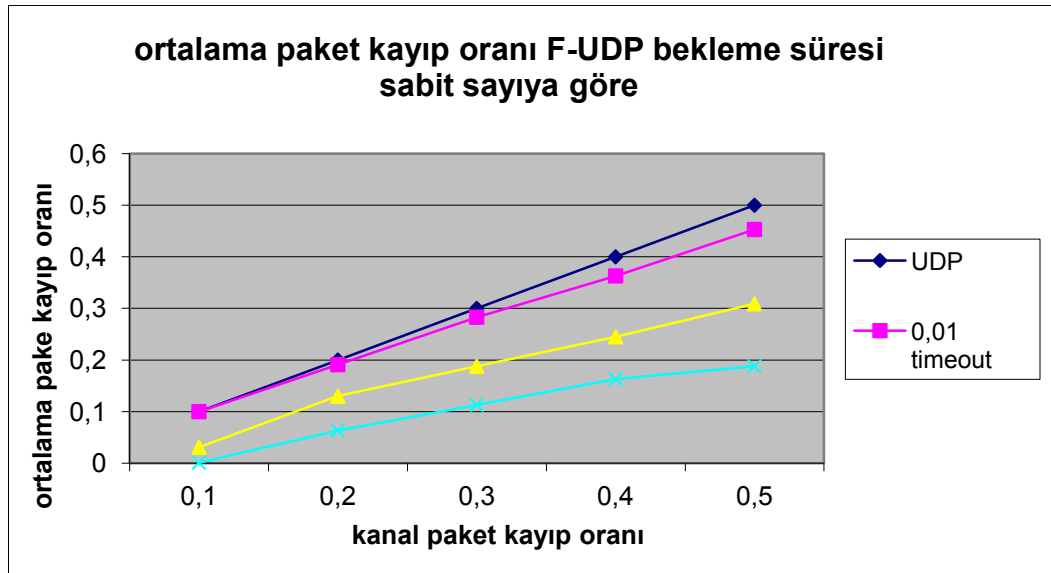


Şekil 3.20. F-UDP için tampon bekleme süresi RTT paket kayıp oranı

Son tampon bekleme mekanizması olarak, süre sabit bir değer seçilmiştir. Bu süre seçilirken gerçek zamanlı uygulama için sınır bekleme süresi, uygulama katmanı ve iletim katmanı için işlem zamanları dikkate alınarak değerler seçilmiştir. Seçilen değerlere göre gecikme ve paket kayıp oranları aşağıda grafikte verilmiştir.



Şekil 3.21. F-UDP için tampon bekleme süresi sabit değeri için paket gecikmesi



Şekil 3.22. F-UDP için tampon bekleme süresi sabit değeri için paket kayıp oranı

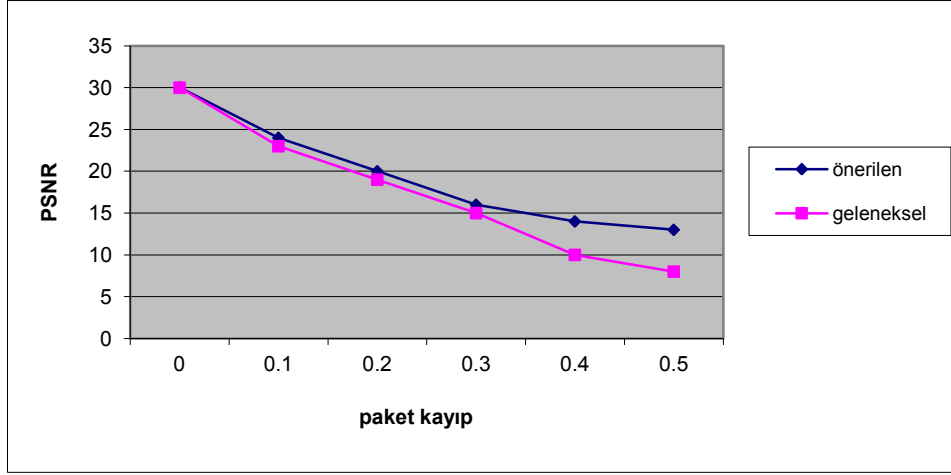
Grafiklerin sonuçlarına bakıldığında bekleme süresi bakımından en iyi sonuç sabit 10 milisaniye timeout süresiyle alınmıştır. Paket kayıp oranı en düşük olan değerler yine 100 milisaniye sabit bekleme ve  $3*RTT$  bekleme süresiyle alınmıştır. Gerçek zaman etkileşimli ses ve video kriteri için gecikmenin göreceli olarak ve paket kayıp oranının bu göreceli duruma göre az olduğu en iyi değer 50 milisaniye sabit bekleme süresi olduğu görülmüştür. Kanal paket kayıp oranının 0.5 olduğu değerde bile paket kaybının %30'larda ve gecikmenin 50 milisaniyelerde olduğu düşünülürse iyi bir sonuç elde edilmiştir.

Sonuçlara bakıldığında tampon için bekleme süresi artırıldığında paket kayıp oranlarının düştüğü fakat uygulama katmanına gönderilecek olan paketlerin bekletilme olasılığının yüksek olması nedeniyle, gerçek zamanlı uygulamalar için istenmeyen bir durum ortaya çıkmaktadır. Gerçek zaman etkileşimli uygulamalarda üst iyi bir başarı için üst sınır 150 milisaniye olması göz önünde bulundurulduğunda, kablolu ağ tarafındaki işlem ve yayılım gecikmeleri, kablosuz ağa girişteki tamponlama, sıra numarası koyma işlemleri göz önünde bulundurulduğunda tamponda bekleme süresinin daha aşağı değerler alınması gerekir. Diğer taraftan timeout'un küçük bir değer seçilmesi de paket kayıp oranını neredeyse kanalın başarımı ile aynı olmasına ve UDP için geri besleme yönteminin, gelenekse UDP yönteminden farklı sonuç üretememesine ve paket kayıp oranının aynı kalmasına neden olmaktadır.

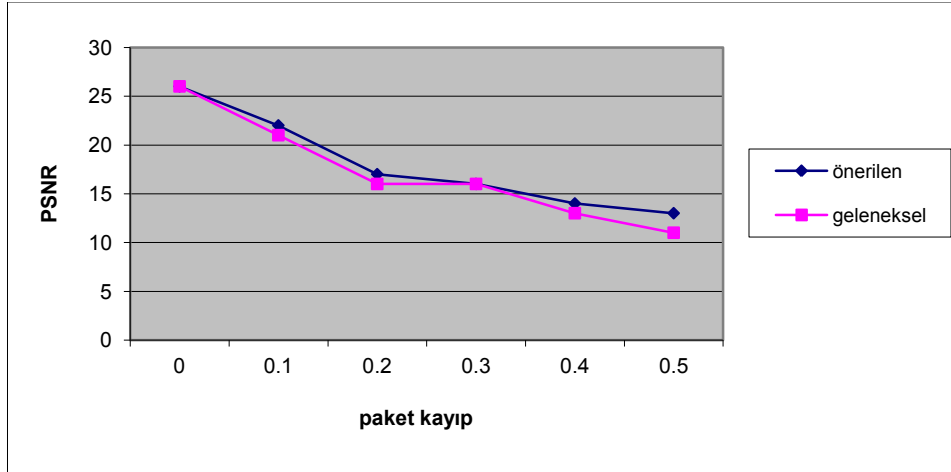
Kablosuz ağlardaki paket kayıp oranının yüksek oluşu nedeniyle gerçek zamanlı uygulamalardaki UDP protokolü kullanımı nedeniyle uygulamalarda problemler yaşanmaktadır. F-UDP ile gerçek zamanlı uygulamalar için 150 milisaniye gibi kabul edilebilir gecikme süresini alıcıyı beslemek için kullanımı hata oranını düşürmek için uygulanabilecek yöntemlerden birisi olduğu simülasyon sonuçlarında görülmektedir.

PSNR Test uygulama-I: Benzetimi yapmak için 1 adet kablosuz haberleşme yapmayan video sunucusu, 1 kablosuz erişim noktası, 1 adet kablosuz video alıcı yerleştirilmiştir. Video sunucusu ile kablosuz erişim noktası arası erişim hızı 10 Mbps, kablosuz ağ erişim hızı yine 10 Mbps ve kablolu ağ gecikmesi 10 milisaniye olarak tasarlanmıştır.

PSNR Test Uygulama II: I. Test uygulamadan farklı olarak kablosuz ağı trafik amaçlı kablosuz bilgisayar eklenmiştir. Bilgisayarda 6 Mbps FTP trafiği mevcuttur.



Şekil 3.23. Test 1 için PSNR değerleri



Şekil 3.24. Test 2 için PSNR değerleri

Test1 ve Test 2 incelendiğinde önerilen çalışmanın geleneksel UDP'den daha iyi sonuçlar verdiği fakat trafik arttığında AP'den kaynaklanan gecikme ile paketlerin geleneksel UDP gecikmesine doğru arttığı görülmüştür.

## 5. SONUÇLAR VE ÖNERİLER

İnternet tabanlı gerçek zamanlı uygulamalar için genellikle, UDP ulaşım katmanı protokolü olarak kullanılır. İletim katmanının protokolleri olan TCP veya UDP’de alıcıyı gönderilen paketlerin kayıp oranı, kablosuz yerel alan uygulamalarında önemli bir sorundur. UDP’de sorunu çözmek için önerilmiş olan uygulama katmanında ileri hata düzeltme yöntemleri bulunmaktadır. TCP için de yine bağlantı bölümlene yöntemleri, akış kontrol, geriye dönük hata düzeltme yöntemleri kullanılarak paket kaybı yok edilmiş ve bekleme süreleri azaltılmıştır. Önerilen ağ tasarımında, gerçek zamanlı uygulamalar için çok yaygın olarak kullanılan UDP protokolünün, kablosuz ağlardaki paket kayıpları ve bozulması durumunda geriye doğru hata düzeltme yapılması öngörülmüştür. Geleneksel UDP içinde kayıp olan paketlerin tekrar istenmesi, protokol dizaynı nedeniyle mümkün olamamaktadır. Bu nedenle, protokolün işleyişi ve paket başlığına eklenen bilgilerle UDP protokolünün kablosuz TCP metodlarından bağlantı bölme benzetimi sağlanarak güvenilirliği sağlanmaya çalışılmış ve zamanda paket bekleme süresi azaltılması amaçlanmıştır. UDP için paket iletimi ortamı kablolu ve kablosuz olarak ikiye ayrılmış, kablolu tarafta geleneksel UDP çalışmış, kablosuz tarafta geriye dönük hata düzeltme yöntemi ile paket kayıp oranı düşürülmeye çalışılmıştır. Protokol tasarlanırken, paket kayıp oranının düşürülmesi yanında gerçek zamanlı uygulamalar için kullanılması dikkate alınarak bekleme süresi de geleneksel UDP ile yakın değerlerde olması göz önünde bulundurulmuştur.

Gerçek zaman uygulamalar için paketin ağ içerisindeki dolaşım sırasında geçen süre önemlidir. Gerek göndericideki, gerek ağ içerisinde geçtiği ağ cihazları üzerinde gerek de alıcı içerisindeki işlem zamanı gecikmeye neden olmaktadır. Bu gecikmenin sınırları önceki çalışmalarda yayınlanmıştır. 150 milisaniyenin altında gecikme gerçek zamanlı uygulamalar için problem oluşturmamaktadır. 150 milisaniye ile 400 milisaniye arasındaki gecikmeler ise iletişimi bozmamak ile birlikte kullanıcılar tarafından hissedilebilir gecikmeler olarak tanımlanmaktadır.

150 milisaniye önerilen F-UDP çalışması için kritik parametre olmuştur. Bu parametre gerçek zamanlı uygulamalar için kablosuz ağlardaki paket kaybını azaltmak için kullanılabilirliği dikkate alınarak geriye doğru hata düzeltme mekanizması oluşturmak F-UDP için temel çözüm yolu olmuştur.

Simülasyonlar sonuçlarına bakıldığında gecikme ve paket kaybının azalması F-UDP için ters orantılı bir denkleme dönüşmüştür. Paketlerin tamponda bekletilme süresinin tespiti için yapılan uygulamalarda, paket saydırma ve RTT hesaplanması için gereken ek süre gecikmeye neden olmuş, her ne kadar paket kayıp oranları birbirine yakın olsa da en az gecikme sabit olarak belirlenen tamponda bekletme süresi (timeout) olmuştur.

## KAYNAKLAR

Ayanoğlu E., Paul S., Laporta T. F., Sabnani K. K., Gitlin R. D., AIRMAIL A Link-Layer Protocol for Wireless Networks, *ACM/Baltzer Wireless Networks Journal*, 1995, **1**, 47–60.

Bakre A., Badrinath B. R., I-TCP : Indirect TCP for mobile hosts, *The 15th International Conference on Distributed Computing Systems*, Vancouver, Canada, July 1995.

Balakrishnan H., Padmanabhan V., Seshan S., Katz R., A Comparison of Mechanism for Improving TCP Performance Over Wireless Links, *IEEE/ACM Transactions on Networking*, 1997, **5**, 756-769.

Bolot J. C., Turetti T, A rate control scheme for packet video in Internet, *IEEE Infocom*, 1994, **1**, 1216-1223.

Bolot J. C., Vega-Garcia A., Control Mechanisms for Packet Audio in The Internet, *IEEE Infocom*, 1996, 232-239.

Basset S. A., Schulzrinne H., An analysis of the Skype peer-to-peer Internet Telephony Protocol, *IEEE Infocom*, 2006, 1-11.

Floyd S., Kohler E., Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control, RFC, 4341, 2006.

Hassan M., Jain R., *High Performance TCP/IP Networking: Concepts, Issues and Solutions, 1th ed.*, Pearson Prentice Hall, New Jersey, 2004.

Klaue J., Rathke B., Wolisz A., EvalVid – A Framework for Video Transmission and Quality Evaluation, *The 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, Urbana, Illinois, September 2003.

Kurose F., Ross K., *Computer Networking.*, Pearson Education, New Jersey, 2005.

Lam P. P. , Liew S. C., UDP-Liter: an improved UDP protocol for real-time Multimedia applications over wireless links, *Wireless Communication Systems 1st International Symposium*, Mauritius, 20-22 December 2004.

Larzon L., Degermark M., Pink S., UDP Lite for Real Time Multimedia Applications, *HP Labs*, Bristol, United Kingdom, Tech. Rep. HPL-IRI-1999-001, April 1999.



Leung K. C., Li V. O. K., Transmission Control Protocol (TCP) in Wireless Networks: Issues, Approaches, and Challenges, *IEEE Communications Surveys & Tutorials*, 2006, **8**, 64-79.

Mitzel D., Overview of 2000 IAB Wireless Internetworking Wksp., Request For Comments, RFC, 3002, 2000.

Mogul J., Deering S., Path MTU Discovery, RFC, 1191, 1990

Mondal S. A., Luqman F. B., Improving TCP performance over wired-wireless Networks, *Computer Networks, Elsevier*, 2007, **51**, 3799-3811.

Parsa C., Aceves G. L., Improving TCP Performance over Wireless Networks at the Link Layer, *IEEE Mobil Network and Applications*, 1999, **5**, 57-71.

Partridge C., Hinden R., Version 2 of the Reliable Data Protocol (RDP), RFC, 1151, 1990.

Pentikousis K., Tcp in Wired-Cum-Wireless Environments, *IEEE Communications Surveys*, 2000, **3(4)**, 2-14.

Postel J. B., Transmission Control Protocol, RFC, 793, 1981.

Ramakrishnan K., Floyd S., Black D., The Addition of Explicit Congestion Notification (ECN) to IP, RFC, 3168, 2001.

Saltzer H., Reed D. P., Clark D., End-to-end Arguments in System Design, *ACM Trans. Computing Systems*, 1984, **2(4)**, 277-288.

Schulzrinne H., Casner S., Frederick, R., Jacobson V., RTP: A Transport Protocol for Real-Time Applications, STD 64, RFC, 3550, 2003.

Spring N., Wetherall D., Ely D., Robust Explicit Congestion Notification (ECN) Signaling with Nonces, RFC, 3540, 2003.

Szigeti T., End-to-End QoS Network Design, *Cisco Press*, Indianapolis, 2004.

Thompson K., Miller G. J., Wilder R., Wide-Area Internet Traffic Patterns and Characteristics, *IEEE Network*, November 1997, **11(6)**, 10-23.

URL -1: [http://www.ciscosystems.com/en/US/products/hw/routers/ps272/products\\_configuration\\_guide\\_chapter09186a008033-a48d.html](http://www.ciscosystems.com/en/US/products/hw/routers/ps272/products_configuration_guide_chapter09186a008033-a48d.html), Configuring QoS in a Wireless Environment, , Cisco Systems Corp. (Ziyaret tarihi: 10.1.2010)

Velayos H., Karlsson G., Techniques to Reduce IEEE 802.11b MAC Layer Handover Time, *IEEE ICC*, Paris, France, 20-24 Haziran 2004.

Wei W., Wang B., Zhang C., Kurose J., Towley D., Classification of Access Network Types: Ethernet, Wireless LAN, ADSL, Cable Modem or Dialap?, *IEEE Infocom*, Miami, USA, 13-17 March 2005.

Wei W., Zhang C., Kurose J., Towsey D., Inference and Evaluation of Split-Connection Approaches in Cellular Data Networks, *Active and Passive Measurement Workshop*, Adelaide, Australia, March 2006.

Yavatkar R., Bhagawat N., “Improving End-to-End performance of TCP over Mobile internetworks”, *IEEE Workshop on Mobile systems*, California, USA, 8-9 December 1994.

Zheng H., Boyce J., *Streaming Video over Wireless Networks, Wireless Internet Handbook, Technologies, Standards and Applications*., CRC press, 105-126, 2003.

Zhen-ning K., Performance Analysis of IEEE 802.11e Contention-Based Channel Access, *IEEE Journal On Selected Areas In Communications*, 200, **22(10)**, 2095-2106 .

## **KİŞİSEL YAYINLAR VE ESERLER**

Uluslararası Bilimsel Toplantılarda Sunulan ve Bildiri Kitabında Basılan Bildiriler

- [1] **Gök O.**, Demiray H. E., F-UDP: A New Approach for Real Time Multimedia Access on Wired to Wireless LANs, *ICECCO'12*, Ankara, Türkiye, 1-3 Kasım 2012.

Ulusal Bilimsel Toplantılarda Sunulan ve Bildiri Kitaplarında Basılan Bildiri ve Posterler

- [1] **Gök O.**, Demiray H. E., Kablolu – Kablosuz Yerel Alan Ağlarında Gerçek Zamanlı Uygulamalar İçin UDP Benzetimi, *Elektrik Elektronik Günleri 2011*, Ankara, 29 Eylül-1 Ekim 2011.

## **ÖZGEÇMİŞ**

1978 yılında Samsun, Çarşamba'da doğdu. İlkokulu Amasya Suluova Zübeyde Hanım İlkokulunda, orta ve lise öğrenimini Amasya Anadolu Lisesi'nde tamamladı. 1997 yılında girdiği Kocaeli Üniversitesi Bilgisayar Mühendisliği lisans eğitimini 2001 yılında tamamladı. 2001-2004 yılları arasında Kocaeli Üniversitesi, Fen Bilimleri Bilgisayar Müh. Ana Bilim Dalı'nda yüksek öğrenimini tamamladı.