

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(DOKTORA TEZİ)

**MPEG-1 VİDEOLARININ İNTERNET ORTAMINDA
İLETİLMESİ VE GÖSTERİLMESİ ÜZERİNE BİR
ÇALIŞMA**

96484

Aylin KANTARCI

Bilgisayar Mühendisliği Anabilim Dalı

Bilim Dalı Kodu : 619.01.00

Sunuş Tarihi: 7.6.2000

Tez Danışmanı : Prof. Dr. Turhan Tunalı

Bornova, İZMİR

III

Sayın Aylin KANTARCI tarafından DOKTORA TEZİ olarak sunulan "MPEG-1 Videolarının İnternet Ortamında İletilmesi ve Gösterilmesi Üzerine Bir Çalışma" adlı bu çalışma, "Lisansüstü Eğitim ve Öğretim Yönetmeliği" nin 24 üncü madde (c) (d) bentleri ve Enstitü yönergesinin ilgili hükümleri dikkate alınarak tarafımızdan değerlendirilmiş olup yapılan sözlü savunma sınavında aday oybirliği..... ile başarılı bulunmuştur. Bu nedenle Aylin KANTARCI 'nın sunduğu metnin doktora tezi olarak kabulüne oybirliği..... ile karar verilmiştir.

7./6../2000

Jüri Başkanı ; Prof. Dr. Turhan Tunalı

Raportör ; Prof. Dr. Turhan Tunalı

Üye ;

Üye ;

Üye ;

Prof. Dr. Sinan Yılmaz

Yrd. Doç. Dr. Haldun Sarnel

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 8/6/2000... gün ve24/17..... sayılı kararı ile onaylanmıştır.

Enstitü Sekreteri

Dr. Süleyman Boruzanlı

Enstitü Müdürü

Prof. Dr. Alaettin Taysun

ÖZET

**MPEG-1 VİDEOLARININ İNTERNET ORTAMINDA
İLETİLMESİ VE GÖSTERİLMESİ ÜZERİNE BİR ÇALIŞMA**

KANTARCI, Aylin

Doktora Tezi, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Turhan Tunalı

Mayıs 2000, 164 sayfa

Bu çalışmada, MPEG-1 videolarının Internet üzerinden iletilmesi işlemini yerine getiren bir sistem geliştirilmiştir. İletişim protokolü olarak Lucent Technologies'den sağlanan RTP kütüphanesi kullanılmıştır. Sistem, RTCP protokolü ile alınan istatistiklere göre iletişim ortamındaki değişimlere dinamik olarak uyum sağlayabilmektedir. Bant genişliği yönetimi, video büyüklüğü ve paket sayısı istatistiklerini dikkate alarak gerçekleştirilmektedir. Sunum sürekliliğinin sağlanması için tampon saha doluluğunu gözönüne alan bir akış kontrol mekanizması geliştirilmiştir.

Sun iş istasyonları üzerinde çalışan sistem, Solaris işletim sisteminin çok iş parçacıklı ve gerçek zamanlı programlama kütüphanelerini kullanarak oluşturulmuştur. Sistem Gigabit Ethernet omurgası üzerinde denenmiştir. Başarım sonuçları, sistemin kampüs yerel ağları üzerinde başarılı bir şekilde çalışabileceğini göstermektedir.

Anahtar Kelimeler: MPEG-1 sıkıştırma, Real-Time Transport Protokolü, akışlandırma, Internet, çok iş parçacıklı programlama, POSIX4

VII

ABSTRACT

A STUDY ON THE TRANSMISSION AND PRESENTATION OF MPEG-1 VIDEOS OVER THE INTERNET ENVIRONMENT

KANTARCI, Aylin

PhD Thesis in Computer Engineering

Supervisor: Prof. Dr. Turhan Tunalı

May 2000, pages

In this study, a software only video on demand system to transmit MPEG-1 videos over the Internet has been developed. RTP protocol by Lucent Technologies has been used as the transmission protocol. System dynamically reacts to network conditions with the usage of RTCP protocol. Bandwidth management has been performed according to video size statistics and number of packets to be transmitted. A flow control mechanism has been developed to provide the continuity during presentation.

The system has been developed on Sun ULTRA workstations by using POSIX 4. real-time extensions and multithread programming facilities of Solaris operating system. The system has been tested on our Gigabit ethernet Campus LAN. Performance results indicate that the system can successfully be used on Campus LANs.

Keywords: MPEG-1 compression, Real-time Transport Protocol, Streaming, Internet, multithread programming, POSIX 4.

TEŞEKKÜR

Tez çalışması sırasında değerli fikirlerinden ve sağladığı olanaklardan yararlandığım, çalışma boyunca sabır ve hoşgörüsü ile beni destekleyen danışmanım Sayın Prof. Dr. Turhan Tunalı'ya teşekkürlerimi sunarım.

Başta annem olmak üzere, doğduğum günden bu yana maddi ve manevi her türlü desteği, sevgi, sabır, fedakarlık ve hoşgörüsüyle bana sunan ve ilgilerini hiçbir zaman esirgemeyen tüm aileme teşekkürü borç bilirim.

Ayrıca, RTP protokolünün kullanımı konusunda destek sağlayan Lucent Technologies mühendislerinden Sayın Reha Civanlar ve Jonathan Roxenberg'e, video verilerini bilgisayar ortamına aktarma konusunda göstermiş oldukları yardımdan dolayı Ege Üniversitesi Tıp Fakültesi Odyovizyel Merkez Yetkililerine ve bugüne kadar göstermiş olduğum başarıda emeği geçen tüm öğretmenlerime teşekkürlerimi sunarım.

İÇİNDEKİLERSayfa

ÖZET..... V

ABSTRACT..... VII

TEŞEKKÜR..... IX

İÇİNDEKİLER..... XI

ŞEKİLLER DİZİNİ.....XV

ÇİZELGELER DİZİNİ.....XIX

SÖZLÜK.....XXI

1. GİRİŞ..... 1

2. ÇOKLUORTAM SİSTEMLERİNE GİRİŞ..... 5

2.1 Çokluortam Verilerinin Özellikleri..... 5

2.2 Veri Sıkıştırma..... 7

İÇİNDEKİLER (devam)

Sayfa

2.2.1 MPEG: Moving Pictures Experts Group	9
2.3 Servis Kalitesi	13
2.4 Kaynak Yönetimi	15
2.5 Çokluortam Sistem Mimarileri	18
2.6 Çokluortam Uygulamaları İçin Sistem Desteği	21
2.6.1 İşletim sistemi desteği.....	21
2.6.2 İletişim desteği.....	28
2.7 Çokluortam Verilerinin İnternet Üzerinden İletimi.....	34
2.7.1 RTP (Real-time Transport Protocol)	38
2.7.2 RTCP (Real-time Control Protocol).....	39
2.8 Örnek Sistemler.....	40
2.8.1 Sistem mimarisi	40
2.8.2 İletişim mekanizmaları.....	43
2.8.3 Servis kalitesi kontrolü	44
2.8.4 Hata kurtarma.....	45
2.8.5 İşletim sistemi desteği.....	46

İÇİNDEKİLER (devam)

Sayfa

3	GELİŞTİRİLEN SİSTEM	48
3.1	Sistem Mimarisi.....	49
3.2	Servisçi.....	51
3.2.1	Alicılardan gelen bağlantı isteklerinin kabulü.....	52
3.2.2	Video verilerinin akışlara dönüştürülmesi.....	53
3.2.3	RTP paketlerinin iş dağıtımının gerçekleştirilmesi.....	56
3.2.4	Bant genişliği yönetimi	58
3.2.5	İletişimle ilgili istatistiklerin toplanması.....	62
3.2.6	Servis kalitesi kontrolü.....	64
3.3	İstemci.....	67
3.3.1	İstemci Mimarisi	69
3.3.2	Tampon saha kullanımı	71
3.3.3	Akış kontrolü.....	73
3.4	Kontrol Mesajları.....	81
3.5	İş Dağıtımı.....	82

İÇİNDEKİLER (devam)

Sayfa

4. BAŞARIM SONUÇLARI	83
4.1 Altyapı	83
4.2 Video Veritabanı	83
4.3 Sistem Geliştirilmesi Sırasında Kullanılan Araçlar	88
4.4 Sistem Yüğü	95
4.4.1 Servisçi Yüğü	955
4.4.2 İstemci Yüğü	97
4.5 Bant Genişliği Gereksinimi	100
4.6 Tampon Saha Yönetimi	1155
4.7 RTCP İstatistikleri	121
4.7.1 Kayıp İstatistikleri	122
4.7.2 Kayma ('jitter')	1244
5. SONUÇ VE TARTIŞMA	129
KAYNAKÇA	132
ÖZGEÇMİŞ	140

ŞEKİLLER DİZİNİ

Sekil

Sayfa

2.1. Bir MPEG1 video dizisinin yapısı.....	11
2.2. Bir çokluortam sistem mimarisi	19
2.3. İnternet Protokolü katmanları.....	32
2.4. BSD UNIX IP gerçekleştirimi.....	34
2.5. Gerçek zamanlı iletişim için IETF tarafından önerilen protocol hiyerarşisi.....	38
2.6 (a)Berkeley Continuous Media Player (b).Oregon OGI mimarileri	42
3.1. Sistem mimarisi	50
3.2. Servisçiye ait nesne tabanlı mimari	51
3.3. Ağ arayüzünden çıkan bir paketin yapısı.....	56
3.4. Paketlerin gönderiş zamanlarının belirlenmesi.....	58

ŞEKİLLER DİZİNİ (devam)

Sekil

Sayfa

3.5. Paketlere ayırmada (a) çerçeve tabanlı yaklaşım ve (b) GOP tabanlı yaklaşımların karşılaştırılması.....	61
3.6. Servis kalitesi kontrolü	66
3.7. İstemci yazılımı mimarisi.....	68
3.8. Birlikte işletime izin vermeyen bir istemci mimarisi	69
3.9. Alıcı istasyona ait akış diyagramının 'pipeline' modeline eşlenmesi.....	70
3.10. Tampon saha kullanımı.....	72
3.11. Alıcıda kullanılan tampon sahalar	76
3.12. Akış kontrolü işlemi için tampon sahada ayrılan bölgeler.....	78
3.13. Kullanılan akış kontrol yöntemi.....	80
4.1 Ağ topolojisi.....	84
4.2. Bir MPEG videoya ait istatistiklerin bulunduğu dosyanın görüntüsü.....	88

ŞEKİLLER DİZİNİ (devam)

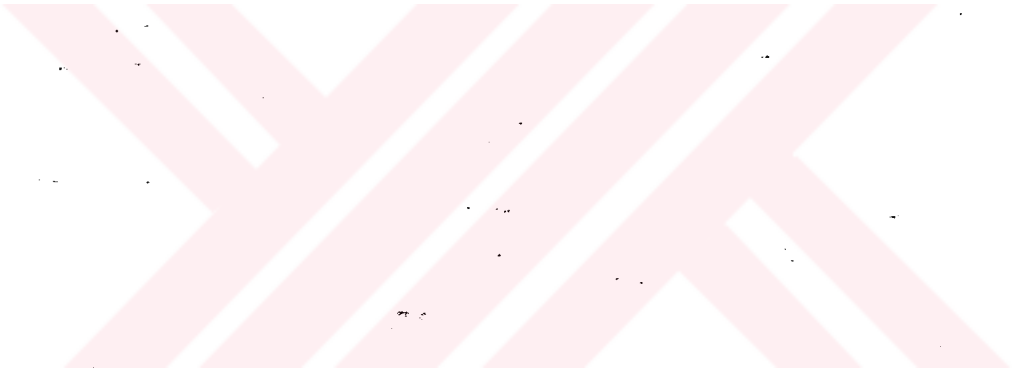
Sekil

Sayfa

4.3. Servisçi modülüne ait 1. seviye akış diyagramı.....	89
4.4. İstemci yazılımına ait 1. seviye akış diyagramı.....	90
4.5. Servisçi modülü 2. seviye akış diyagramı.....	91
4.6. İstemci modülü 2. seviye akış diyagramı.....	92
4.7. Farklı uyum seviyelerinde ve sıkıştırma bit hızlarında ihtiyaç duyulan bant genişliği miktarı	101
4.8. Farklı uyum seviyelerinde ve sıkıştırma bit hızlarında kullanılan paket sayısı.....	102
4.9. INTERVAL_FOR_EACH_GOP yöntemine göre 100, 200,500, 1000 ve 1500 Kbps için beklenen ve gözlenen paketler arası aralık.....	105
4.10. INTERVAL_FOR_VIDEO yöntemine göre 100, 200,500, 1000 ve 1500 Kbps için beklenen ve gözlenen paketler arası aralık.....	111
4.11. Sıkıştırma bit hızının 100, 200, 500, 1000 ve 1500 Kbps olduğu durumda tampon sahada bulunan çerçeve sayısı	117

ŞEKİLLER DİZİNİ (devam)**Sekil****Sayfa**

- 4.12. 1000 Kbps için ön tamponlama süresinin
1 ve 2 dak. olduğu durumlarda
tampon saha doluluğu116
- 4.13. 100, 200, 500, 1000 ve 1500 Kbps'lik
sıkıştırma hızları için kayma değerleri..... 128



XIX

ÇİZELGELER DİZİNİ

<u>Cizelge</u>	<u>Sayfa</u>
2.1. Farklı MPEG versiyonlarının karşılaştırılması.....	13
2.2. Aygıt katmanında yer alan elemanlar	20
3.1. IBBPBBPBBPBBPBB GOP dizisi için hız uyum seviyeleri	67
3.2. Kullanılan uyum seviyeleri için geçerli oynatış hızları.....	74
4.1. Video veritabanını oluşturan videolara ait özellikler	86
4.2. Servisçi tarafında paketleme ve RTP kütüphanesi harcamaları.....	96
4.3. Milisaniye cinsinden XIL kütüphanesinde harcanan süre	99
4.4. Farklı uyum seviyeleri için hesaplanan tahmini paket sayıları ve gönderilen gerçek paket sayıları	103
4.5. Kayıp istatistikleri	124

SÖZLÜK

açıcı:	decoder
açma:	decompression, decode
akış kontrol:	flow control
botuhattı:	pipeline
çok iş parçacıklı:	multi threaded
çokluortam:	multimedia
çözünürlük:	resolution
en iyi çaba:	best effort
esnek garantiler:	soft guarantees
gerçek zaman:	real-time
hız kontrol:	rate control
istemci:	client
iş parçacığı:	thread
izleme:	monitoring
katı garantiler:	hard guarantees
kayma:	jitter
kronometre süresinin dolması:	timer expiration
renklendirme:	dithering
servis kalitesi:	quality of service, QoS
servis kalitesi eşlemesi:	QoS mapping
servisçi:	server
sıkıştırma:	compression
sıkıştırma bit hızı:	encoding bit rate
süreçler arası iletişim:	interprocess communication
tampon saha:	buffer
uçtan uca gecikme:	end to end delay
uyum:	adaptation
veri patlaması:	burstiness

1. GİRİŞ

Bilgisayar teknolojilerindeki hızlı ilerleme, ses ve video konferansları, istek üzerine video, etkileşimli TV gibi dağıtık çokluortam uygulamalarının önem kazanmasına neden olmuştur.

Günümüz bilgisayar ağları, metin verilerini başarıyla taşıyabilmektedir. Ancak, aynı başarıyı çokluortam verilerini taşımada gösterememektedir. Video, ses gibi çokluortam verileri, aşağıdaki nedenlerden dolayı günümüz ağları üzerinde uygulama geliştirmeyi güçleştirmektedir:

- i. Çokluortam verileri, metin verilere göre çok daha yüksek bir bant genişliğine ihtiyaç duyarlar. Örneğin, 25 sn.lik 320x240 çözünürlüğünde bir QuickTime videosu, 2-3 MB'lık yer kaplar. Bu miktar, 1000 ekranlık metin verisinin kapladığı yere eşittir.
- ii. Çokluortam uygulamaları gerçek zamanlı olarak ağ üzerinde iletilmelidir. Ses ve video, örnekleme hızına uyumlu bir hızda, sunum sürekliliğini tehlikeye sokmayacak bir şekilde oynatılmalıdır. Gecikmeler, sunum sırasında duraklamalara ve sürekliliğin bozulmasına yol açar. Tıkanıklığın gerçek zamanlı trafik üzerinde daha olumsuz bir etkisi vardır. Tıkanıklık; gecikmelere yol açmanın yanı sıra paketlerin kaybolmasına da neden olur.
- iii. Çokluortam verilerinin iletimi sırasında sık sık 'veri patlamaları' (herhangi bir anda ağ trafiğinin ya da işlemci yükünün ani artışı) yaşanır. Sadece bant genişliğinin arttırımı

bu problemi çözmez. Alıcıdaki tampon saha miktarı sınırlıdır. Veri akışını düzgünleştirecek önlemlerin alınmaması önemli problemlere yol açabilir. Veri geliş hızının yüksek oluşu, tampon sahada taşmalara ve bazı veri paketlerinin kaybolmasına yol açar. Verilerin çok düşük bir hızda gelişi ise tampon sahada boşalmalara ve uygulamada duraklamalara neden olur.

Yukarıdaki nedenlerden dolayı ilk çokluortam uygulamaları ATM (Asynchronous Transfer Mode), adanmış hat ve kablolar gibi özel donanımlar üzerinde gerçekleştirilmiştir. Adanmış hat ve kablolar, özel donanım ve yazılım gerektirdiğinden pratik değildir. ATM; yüksek bant genişliği olanakları, bağlantı tabanlı oluşu ve değişik servis kalitelerini destekleme özelliklerinden dolayı çokluortam verileri için en iyi çözüm olarak görünmesine rağmen gerektirdiği harcamalar nedeniyle yeterince yaygınlaşamamıştır.

Diğer yandan Internet, dünyadaki tüm ağları bir araya getirecek şekilde hızla büyümekte ve bir çok dağıtık uygulama için bir platform oluşturmaktadır. Internet, aynı zamanda değişik çokluortam verilerini tek başına entegre edebilmekte; farklı ağlar arasında donanım ve arayüz gereksinimlerini ortadan kaldırmaktadır. Günümüzde, IP (Internet Protocol) ve Ethernet yerel ağlarda, ATM ise geniş alan ağlarında tercih edilmektedir.

Paylaşımlı bir datagram ağı olması nedeniyle, Internet gerçek zamanlı trafik için uygun değildir. Çokluortam uygulamalarını Internet üzerinden çalıştırabilmek için bazı iyileştirmelere gidilmesi gerekmektedir. Bu iyileştirmelerin en önemlilerinden biri, ses ve video

verilerinin gerçek zamanlı iletimini sağlayacak şekilde yeni nakil protokollerinin geliştirilme gerekliliğidir.

Uygulamaların sağlıklı bir şekilde işletimi, ağ kaynaklarının başarılı bir şekilde yönetiminin yanı sıra uç istasyonların başarımına da büyük ölçüde bağlıdır. Uç istasyonlarda kullanılan işletim sistemlerinin özelliklerinin de uygulamalar üzerinde büyük etkisi vardır.

Bu tez kapsamında Internet ortamı için bir istek üzerine video sistemi geliştirilmiştir. Benzer çalışmalar (Kee- Yin et al., 1998; Chen 1995; Chang et al, 1995, Cen et al., 1995; Patel and Rowe, 1997 genellikle sadece video verilerinin iletimi üzerine odaklanmıştır. Geliştirdiğimiz sistem ise veri iletişiminin yanı sıra, uygulamaların işletim sistemi tarafından gerçek zamanlı işletimine de ağırlık vermektedir.

Geliştirilen sistemin sahip olduğu ana özellikler şu şekilde listelenebilir:

1. Sistem sadece yazılım tabanlıdır. Herhangi bir özel donanım gerektirmemektedir.
2. Gerçek zamanlı iletişim için RTP protokolü kullanılmaktadır.
3. Gerçek zamanlı işletim için POSIX4 kütüphanesi ve POSIX iş parçacıkları kullanılmaktadır.
4. Verilerin düzgün şekilde iletimi ve sunumun sürekliliğinin sağlanması amacıyla uygun bant genişliği yönetimi ve akış kontrol mekanizmaları geliştirilmiştir

5. Sistem ağıdaki deęişimlere uyum gösteren bir yapıya sahiptir.

Bu tezin dięer bölümleri şöyle düzenlenmiştir: İkinci bölümde çokluortam verilerinin özellikleri, çokluortam uygulamaları için gereken iletişim ve işletim sistemi özellikleri ve benzer çalışmalar hakkında bilgi verilmektedir. Üçüncü bölümde gerçekleştirdiğimiz sistem mimarisi tanıtılmaktadır. Dördüncü bölümde sistemimizle ilgili başarımlar sunulmaktadır. Beşinci bölümde ise bu sonuçlar değerlendirilmiş ve bu araştırmanın devamı olarak incelenebilecek konular tartışılmıştır.



2. ÇOKLUORTAM SİSTEMLERİNE GİRİŞ

Çokluortam sistemleri birbirinden bağımsız, ayrık ve sürekli veri ortamlarının bilgisayar kontrollü üretim, düzenleme, sunum, saklama ve iletişimi işlemlerini yerine getirirler. Metin ve grafik verileri ayrık verilere, ses ve video verileri sürekli verilere örnek olarak verilebilir. Çokluortam uygulamalarının büyük bir kısmı dağıtık çokluortam kaynaklarına erişimi, bu kaynaklardan elde edilen verilerin yüksek hızlı ağlarla iletilerek entegre edilmesini ve anlamlı bir şekilde kullanıcıya sunulmasını gerektirir. Bu bölümde çokluortam verilerinin özellikleri, servis kalitesi, kaynak yönetimi, işletim ve iletişim sistemleri desteği gibi konularla ilgili bilgi verilecektir.

2.1 Çokluortam Verilerinin Özellikleri

1. Gerçek zamanlılık:

Çokluortam uygulamalarının başarımının en önemli göstergelerinden biri sunum süresince sürekliliğin sağlanmasıdır. Süreklilik, verilerin kullanıcıya belirli zaman sınırları içinde iletilmesiyle sağlanır. Çokluortam sistemleri pek çok yönleriyle geleneksel gerçek zamanlı sistemlerden ayrılırlar. Gerçek zamanlı bir işlemin tamamlanması için gereken zamana *kritik bitiş zamanı* adı verilir. Geleneksel gerçek zamanlı sistemlerde bir işin kritik bitiş zamanı geldiği halde tamamlanmaması ölümcül sonuçlara yol açar. Bu tür uygulamalar *katı gerçek zamanlı* (hard real-time) uygulamalar olarak adlandırılırlar. Gerçek zamanlı uygulamalar içinde kaçırılan bitiş zamanları konusunda daha esnek kısıtlara sahip bir uygulama grubu daha vardır. Çokluortam uygulamaları *esnek gerçek zamanlı* uygulamalar olarak da adlandırılan

bu gruba girer (Tindell and Hunson , 1995). Kaçırılan bitiş zamanları ölümcül hatalara yol açmazlar, sunum sırasında gecikme ve duraklamalara neden olurlar. Bu gecikmeler belirli bir dereceye kadar esneklik gösterilebilen gecikmelerdir. Örneğin, çoğu durumda geciken bir video çerçevesinin atlanarak bir önceki çerçevenin ekranda gösterilmesi kullanıcının dikkatini çekmez (Steinmetz and Nahrstedt, 1995).

2. Periyodiklik:

Çokluortam uygulamaları periyodik süreçlerle temsil edilirler. Örneğin, bir video oynatıcısının hızı 25 çerçeve/sn ise her $1 \text{ sn}/25=40$ milisaniyede bir çerçeveye ait video bilgisinin işlenerek ekranda gösterilmesi gerekmektedir. Kullanılan sıkıştırma algoritmasına göre gelişlerarası zaman sabit olabileceği gibi farklılık da gösterebilir. Genel olarak periyodiklik, sistemin tahminlenebilirliğini arttırdığı için uygulamaların performansı açısından olumlu bir özelliktir (Steinmetz and Nahrstedt , 1995).

3. Yüksek bant genişliği gereksinimi:

Çokluortam uygulamaları küçük bir zaman diliminde büyük miktarda veri işlenmesini gerektiren uygulamalardır. Bu durum, özellikle video verileri için daha belirgindir. Uygulamalar sırasında çeşitli nedenlerden dolayı işletim ve iletişim sistemleri kaynaklarının yetersiz kaldığı durumlarda yaşanan darboğaz, uygulamaların performansını olumsuz yönde etkiler. Konuyla ilgili önemli bir nokta, kullanılan sıkıştırma tekniğine bağlı olarak birim zamanda farklı miktarda verinin işlenmesi gereğidir. Uygulama geliştirmeyi karmaşıklaştıran bu özellik “veri patlaması” (burstiness) olarak adlandırılır (Campbell , 1995).

4. Kayıplara karşı dayanıklılık:

İletişim sırasında çokluortam verilerinin bir bölümünün kaybolması ya da gecikmesine bir dereceye kadar esneklik gösterilebilir. Örneğin, bir video çerçevesine ait bilgilerin bir kısmı kaybolmuşsa o çerçeve yerine ondan bir önceki çerçeve oynatılarak süreklilik sağlanabilir. Değişik veri türleri kayıp ve gecikmelere karşı farklı derecede dayanıklılık gösterir. Örneğin, kulak kayıplara karşı gözden daha hassastır. Bu yüzden, ses verilerindeki kayıplar video verilerine oranla daha çabuk hissedilir (Steinmetz and Nahrstedt, 1995).

2.2 Veri Sıkıştırma

Günümüzün depolama, işlemci ve iletişim ortamları yüksek bellek ihtiyacı olan video, ses ve grafik bilgilerini desteklemede yetersiz kalmaktadır. Örneğin, PAL standartlarına göre 640x480 çözünürlüğünde bir video çerçevesi her piksel için 24 bitlik bellek sahasının kullanımını gerektirir. Bu durumda, bir video çerçevesi $640 \times 480 \times 3 = 921.000$ "byte"lık sahaya ihtiyaç duyar. Bu, 25 çerçeve/sn. lik hıza sahip bir video uygulamasının her saniye için $25 \times 921.000 = 23.025.000$ byte = 22,485 Kilobyte'lık bitlik bellek kullanımı demektir. Sıkıştırılmamış verilerin işlenmesi yüzlerce gigabyte disk sahası ve yüzlerce megabyte'lık tampon saha kullanımı ile mümkün olabilir. İletişim kaynakları da gerekli bant genişliğini desteklemede yetersizdir. Kısaca, hızlı ilerleme ve gelişmelere rağmen günümüz teknolojisi, sıkıştırılmamış verilerin işlenmesi, saklanması ve iletişimi için gerekli kaynakları sağlamaktan uzaktır. Bu nedenle çokluortam verileri sıkıştırılarak işleme sokulurlar. Veri sıkıştırma teknikleri entropi, kaynak ve hibrid teknikler olmak üzere üç sınıfa ayrılır. Kaynak kodlama teknikleri verilerin içeriğine bakarak sıkıştırma yaparlar. En yüksek sıkıştırma bu teknikle sağlanır. Entropi

kodlama, veriler arasındaki ilişkiyi dikkate almaz, tüm verileri basit bir bit dizisi olarak görür. Pek çok sistemde veriler ilk önce kaynak, daha sonra da entropi kodlamaya tabi tutularak sıkıştırma oranı en yükseğe çıkarılır. Bu tür kodlama, hibrid kodlama olarak adlandırılır.

Sıkıştırılmış veriler, kullanıcıya sunulmadan önce orjinal haline getirilmek zorundadır. Bu işleme *açma* (decompression) adı verilir. Açma işleminden sonra elde edilen veriler orjinal verilerle aynı ise kullanılan teknik kayıpsız bir sıkıştırma tekniğidir. Eğer orjinal verilerde kayıplar varsa kullanılan teknik kayıplı bir sıkıştırma tekniğidir. Entropi kodlama kayıpsız, özellikle çokluortam uygulamalarında kullanılan birçok kaynak kodlama yöntemleri kayıplı sıkıştırma teknikleri sınıfına girmektedir.

Sıkıştırma ve açma işlemleri zaman alıcı işlemlerdir. Uygulamanın cinsine göre sıkıştırma ve açma işlemleri simetrik ya da asimetrik özellikler gösterebilirler. Örneğin, video konferansı uygulamalarında sıkıştırma ve açma işlemleri yaklaşık aynı miktarda zaman harcanmasını gerektirir. İstek üzerine video (video on demand) gibi asimetrik uygulamalarda veriler daha önceden sıkıştırılarak saklandığı için sıkıştırma işlemi daha az külfetlidir. Bu nedenle video konferans sistemlerinde sıkıştırma işlemi için mutlaka donanım desteğine ihtiyaç vardır. İstek üzerine video uygulamalarında daha ucuz oldukları için yazılım tabanlı sıkıştırıcılar da kullanılabilir. Açma işlemi için de gerçek zamanlı açma işlemi gerçekleştirebilecek donanımlar kullanılabilir. Donanım çözümleri pahalı çözümlerdir. Çok sayıda müşterinin bulunduğu sistemlerde en mükemmel sonucu üretmelerine karşılık donanım tabanlı açıcıların kullanılması mümkün değildir. Açma işlemi, daha fazla işlemci zamanı harcanması sebep olmasına rağmen yazılım

kullanarak gerçekleřtirmek daha mantıklı bir yaklařımdır (Steinmetz and Nahrstedt , 1995).

Her veri tipi ve uygulama türü için farklı sıkıřtırma teknikleri mevcuttur. Görüntü verileri JPEG; video verileri MPEG, MJPEG, DVI, H. 261, H. 263; ses verileri MPEG gibi tekniklerle sıkıřtırılabilirler. Bu tez MPEG formatında saklanmış verilerin iletimini dikkate aldıđı için sadece MPEG tekniđi üzerinde durulacaktır.

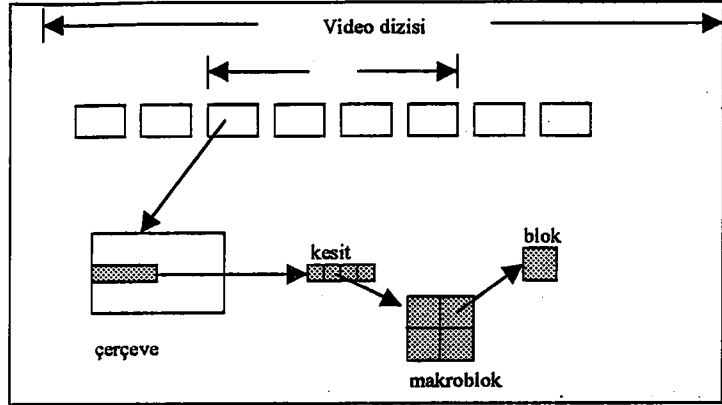
2.2.1 MPEG: Moving Pictures Experts Group

MPEG ISO/IEC JYC1/SC 29/WG11 tarafından geliştirilmiş, hareketli görüntü ve ses verilerini sıkıřtırmak için kullanılan bir tekniktir. JPEG ve H.261 standartları MPEG standartlarının temelini oluřturmaktadır. 1993'de International Standard Organization (ISO) tarafından bir standart olarak kabul edilen MPEG-1 standardı ses ve video verilerinin CD üzerine uygun bir řekilde yerleřtirilmesi amacıyla yola çıkmıřtır. MPEG-1 1.5 Mbps hızla veri erişimine olanak sađlayarak VHS kalitesinde video uygulamalarını mümkün kılar. MPEG-1 mimarisinin daha yüksek bit hızını desteklemeye uygun oluřunun ortaya çıkması, MPEG-2 standardı üzerinde alıřmaların bařlamasına neden olmuřtur (ISO/IEC 13818-2, 1998). MPEG-2, 4-6 Mb/sn'ye varan bit hızı, daha yüksek çözünürlük ve HDTV kalitesinde görüntü sađladığından broadcast servisleri, kablolu TV servisleri, etkileřimli TV servisleri için tercih edilir (Orzessek et al., 1998).

Sayısal video uygulamalarında RGB ve YUV/YCrCb olmak üzere iki tür renk uzayı kullanılır. RGB renk uzayı, bir pikselin rengini kırmızı, yeřil ve mavi olmak üzere üç gruba ayırarak belirler. RGB uzayı TV

uygulamaları için uygun değildir. YUV uzayı, rengi üç elemana ayırmak yerine; rengi Y isimli parlaklık (“intensity”, “luminance”) ve U ve V isimli iki renk (“chrominance”) elemanına böler. Y elemanı siyah ve beyaz video bilgisini içerir. Video sinyalinin sadece Y elemanı kullanılarak siyah-beyaz görüntüler elde etmek mümkündür. Renkli görüntü için video sinyalinin hem Y hem de U, V elemanları kullanılır. YUV renk uzayı gözün rengi algılayış biçimine daha uygun olduğu için TV uygulamalarında tercih edilir. MPEG YUV renk uzayını kullanan bir sıkıştırma tekniğidir (Orzessek et al., 1998).

MPEG standartlarında en küçük kodlama ünitesi, 8x8 lik bloklardır. Parlaklık değerlerini tutan 4 blok ve renk bilgisini tutan, sayısı örnekleme işlemine göre değişen birkaç blok biraraya gelerek bir makrobloku oluşturur. Makrobloklar bir araya gelerek kesitleri (slice) oluşturur. Kesitler rastgele erişimi mümkün kılan ve MPEG standardının hataya dayanırlıklılığını arttıran kodlama üniteleridir. Her kesit, içerdiği makroblokların ekran üzerindeki yeri hakkında bilgi taşır. Eğer iletişim sırasında bir kesitte kayıp ya da bozulma varsa ekran üzerinde o kesite ait yer siyah bırakılıp ya da bir önceki çerçeveye ait aynı kesit gösterilip, bir sonraki kesite geçilerek uygulamanın sürekliliği sağlanır. Şekil 2.1’de gösterildiği gibi, kesitler biraraya gelerek resimleri, resimler bir araya gelerek GOPları (Group of Pictures), GOPlar biraraya gelerek bir video dizisini oluşturur (Orzessek et al., 1998).



Şekil 2.1. Bir MPEG1 video dizisinin yapısı

MPEG çerçeveleri üçe ayrılır: I (intracoded) çerçeveler, diğer çerçevelerden bağımsız olarak kodlanırlar ve sıkıştırma oranı en düşük olan çerçevelerdir. P çerçeveler (predictive-coded picture) I çerçeveleri kullanarak elde edilirler. B çerçeveler (bidirectionally coded pictures) I ve P çerçeveleri kullanılarak kodlanırlar. En yüksek sıkıştırma oranı B tipi çerçevelerde elde edilir. Genel olarak P çerçeveleri I çerçevelerinin %30-50'i, B çerçeveleri P çerçevelerinin %50'si büyüklüğündedir. I, B, P çerçevelerinin farklı büyüklüklerde olması birim zamanda işlenmesi gereken MPEG veri miktarını değişken kılarak veri patlaması ('burstiness') durumunu açığa çıkarmaktadır.

Belli sayıda resmin biraraya getirilmesi ile ortaya çıkan GOPlar rastgele erişim gerektiren fonksiyonlarda kullanılan düzenleme üniteleridir. En sık kullanılan GOP, IBBPBBPBBPBBPBB dizisidir. Bir GOP'a ait B ve P resimleri sadece o GOPa ait I ve P resimlerini kullanarak açılabilirler.

MPEG-2 standardı, MPEG-1 standardından farklı olarak ölçeklenebilir bir standarttır. Bir MPEG bit dizisi iki ya da daha fazla sayıda alt diziye ayrılabilir. Ölçeklendirme yolu ile bu diziler arasından önem derecelerine göre seçim yapılarak uygulamaların esnekliği sağlanır. Örneğin, ATM uygulamalarında veriler iki kanal arasında bölüştürülebilir. En önemli dizi yüksek öncelikli kanalla iletilirken, diğer diziler düşük öncelikli kanalla taşınabilirler. Ağ trafiğinin yoğun olduğu durumda sadece yüksek öncelikli kanal aktif hale getirilir; trafiğin az olduğu durumda her iki kanal üzerinden de iletişim gerçekleştirilebilir (Orzessek et al., 1998).

1999 yılında standart haline gelen MPEG-4, düşük hızlı iletişim ortamlarına yönelik, nesne tabanlı paradigmaya bağlı olarak tasarlanmış bir standarttır. Bir MPEG çerçevesi ses ve video gibi gerçek kaynaklı verilerin yanısıra CAD tasarımı ya da bir çizgi film nesnesi gibi suni kaynaklı nesneleri de içerebilir. Arkaplandan bağımsız olarak kodlanan nesneler gruplandırarak bileşke nesneleri oluştururlar. Kullanıcılar etkileşimli olarak bir çerçeveye ait nesnelerin yerini değiştirebilir, çerçeveye yeni bir nesneyi ekleyebilir ya da bir nesneyi çerçeveden çıkarabilirler. MPEG-4, önceki MPEG standartlarından farklı olarak video çerçevesi gibi diktörtgensel nesnelerin yanısıra gelişigüzel şekillere sahip nesneleri de kullanabilir. MPEG-4 ölçeklenebilir bir çokluortam içeriği sunar. Sıkıştırma sırasında ön planda yer alan nesneler, yüksek bit hızıyla sıkıştırılırken, arkaplandaki düşük öncelikli nesneler düşük bit hızıyla sıkıştırılırlar. İletişim sırasında sabit arkaplan bir kez, hareket halindeki önplan görüntüleri sürekli gönderilerek ya da düşük öneme sahip nesneler hiç gönderilmeyerek ağın yükü azaltılır (Koenen, 1998).

Diğer bir yeni MPEG standardı olan MPEG7, sıkıştırma amaçlı değildir. MPEG-7, video indeksleme, içeriğe göre arama gibi erişim

amaçlı uygulamalar için tasarlanmış ve çokluortam verilerinin tanımlanmasına yönelik bir standarttır (MPEG7 Req. Group, 1997).

Çizelge 2.1'de MPEG standardının değişik versiyonları çeşitli özellikleriyle karşılaştırılmışlardır:

Çizelge 2.1 Farklı MPEG versiyonlarının karşılaştırılması

Versiyon	Hedef	En yüksek çözünürlük	En yüksek çerçeve hızı	En yüksek sıkıştırma bit hızı
MPEG-1	CD uygulamaları	352x288	30 fps	1.5 Mbps
MPEG-2	Yüksek kaliteli TV uygulamaları	720x576	30 fps	4-6 Mbps
MPEG-4	İnternet üzerinde telekonferans-telefon uygulamaları	352x288	10 fps	4.8-6.4 Kbps
MPEG-7	İndeksleme	-	-	-

2.3 Servis Kalitesi

Çokluortam uygulamalarının başarımı, uygulama sırasında kullanılacak tüm kaynakların uygulamaya yeterli miktarda tahsis edilmesine bağlıdır. Bu nedenle uygulamaların kaynak ihtiyaçları *servis*

kalitesi parametreleri adı verilen parametreler cinsinden belirtilir. Servis kalitesi deyimi ilk başta iletişim ortamlarının performansının değerlendirilmesi ve iyileştirilmesi için ortaya atılmıştır. Çokluortam uygulamalarının ortaya çıkışıyla, işletim sistemi kaynakları ve girdi/çıkı aygıtları gibi uç noktalarda bulunan kaynakların da yeterli miktarda tahsisinin garantilenmesinin gerekliliği anlaşılmıştır. Sonuç olarak bu kavramın kapsamı, kaynaktan alıcıdaki sunuma kadar çokluortam verilerinin izlediği yol üzerinde bulunan tüm kaynakları içine alacak şekilde genişletilmiştir (Narhstedt, 1995).

Her kaynak için farklı servis kalitesi parametreleri bulunmaktadır. Bu parametrelerin doğru olarak belirlenebilmesi işletim sistemleri ve ağ teknolojileri konusunda çok alt düzeyde bilgi sahibi olmayı gerektirmektedir. Uç kullanıcıların teknik kavramlar hakkında bilgi sahibi olmaları mümkün olmadığı için servis kalitesi parametreleri katmanlar halinde tanımlanmıştır. En üst katmanda bulunan *Uygulama Servis Kalitesi* parametreleri, veri kalitesi ve veriler arasındaki ilişkilerle ilgilidir. Örneğin, saniyede gösterilecek çerçeve sayısı, çerçeve büyüklüğü gibi bilgiler uç kullanıcıların sağlayacağı uygulama servis kalitesi parametreleridir. *Sistem Servis Kalitesi* parametreleri bellek miktarı, periyot, işletim süresi, kritik bitiş zamanı gibi sistem kaynakları ile ilgili parametrelerdir. *Ağ Servis Kalitesi* parametreleri, birim zamanda iletilecek bilgi miktarı (traffic throughput), kayıp paketlerin oranı, uçtan uca gecikme (end-to end-delay), gecikme varyansı sınırları (jitter) gibi ağ üzerindeki yük ve ağ performansı ile ilgili parametrelerdir. *Aygıt Servis Kalitesi* parametreleri diske erişim zamanı, ağ kartının birim zamanda işleyebileceği veri miktarı gibi aygıtlarla ilgili performans ve zamanlama bilgilerini içerirler. Kullanıcıların uygulama katmanında tanımladığı servis kalitesi parametreleri bir takım işlemlerden geçirilerek alt seviyelerde kullanılacak olan daha karmaşık servis parametrelerine

dönüştürülürler. Bu işleme servis kalitesi eşleştirmesi (QoS mapping) adı verilir (Narhstedt, 1995; Fisher and Keller, 1995).

2.4 Kaynak Yönetimi

Servis kalitesi parametreleriyle belirtilen kalitenin sağlanması başarılı bir kaynak yönetimi ile sağlanır. Geleneksel kaynak yönetimi stratejilerinin temel amacı, kaynakları her uygulamaya eşit şekilde paylaşmaktır. Yüksek veri hacmi ve gerçek zaman kısıtlarına sahip olan çokluortam uygulamaları için böyle bir yaklaşımı kullanmak bazı uygulamalara gereğinden çok ya da az kaynak tahsisi yapılmasına yol açacağından uygulamaların performansı olumsuz yönde etkilenecektir. Bu yüzden çokluortam uygulamaları için yeni kaynak yönetimi yaklaşımları geliştirilmiştir. Bu yaklaşımlar garantiledikleri performans seviyelerine göre aşağıdaki kategorilerde incelenebilirler (Kantarıcı et al., 1996).

“En iyi çaba” (best effort) garantilere dayalı yaklaşımlar: UNIX gibi geleneksel işletim sistemleri ve Internet, uygulamanın kalitesini mevcut kaynaklara göre belirler. Bu tür yaklaşımlarda kaynak yönetimi yetersiz düzeydedir, ya da hiç yoktur.

Katı garantilere dayalı yaklaşımlar: Bu tür yaklaşımlarda kullanıcı, gereksinim duyduğu servis kalitesini belirler ve gereken kaynaklar kullanıcının belirlediği miktarda uygulamaya tahsis edilir. Kaynakların yetersiz kalması durumunda kullanıcının isteği reddedilir. Bu yaklaşım gereksinim duyulan en yüksek miktarda kaynağın uygulama için ayrılmasını gerektirdiğinden kaynak israfına neden olan bir yaklaşımdır.

Esnek garantilere dayalı yaklaşımlar: Sıkıştırılmış verilerin kullanıldığı durumlarda uygulama sırasında birim zamanda kullanılan kaynak miktarı değişken bir özellik gösterir. Bu yüzden garantiler istatistiksel bir şekilde ya da bir değer aralığında belirtilir. 'Paketlerin %85'i 120 milisaniyeden daha az bir gecikmeyle karşı tarafa iletilecektir.' ifadesi bu tür bir garantiye örnektir. Kaynaklar gereksinim duyulan ortalama miktarda tahsis edilir. Bu yüzden katı garantili yaklaşımlardan daha yüksek bir oranda kaynak kullanımı ve aynı anda desteklenen akış sayısında artış sağlanır. Ancak, bu yaklaşım ağda ve işletim sistemi kaynaklarında gecici veya kalıcı yüklenmelere yol açabileceğinden uygulama performansı sürekli olarak izlenmelidir.

Genel olarak kaynak yönetimi işlemleri şu adımlardan oluşur (Nahrstedt and Steinmetz, 1995):

- i. Servis kalitesinin belirlenmesi ve alt seviyelerdeki servis kalitesi parametrelerine dönüştürülmesi:

Kullanıcılar grafiksel bir arayüz ile uygulama seviyesi servis kalitesi parametrelerini sisteme verir. Bu parametreler, bazı fonksiyonlardan geçirilerek ağ ve aygıt seviyesindeki daha karmaşık parametrelere dönüştürülürler.

- ii. Servis kalitesi parametrelerinin dağıtılması ve kabul kontrol:

Servis kalitesi parametreleri kaynak ve alıcı arasında bulunan her düğüme gönderilir. Her düğümde kaynakların yeni bir uygulamayı kabul edip edemeyeceği belirlenir. Kabul kontrol servisi, kullanılan paylaşımlı kaynaklardan arta kalan kapasitenin yeni işin gereksinimleri için yeterli olup olmayacağını belirler. Bunun için yeni iş bazı testlerden geçirilir.

Eğer sisteme yeni bir iş katılmasıyla, çalışan diğer uygulamaların performansı düşmüyorsa ve kaynaklar yeterliyse bu iş sisteme alınır. Kabul kontrol testleri uçtan uca tüm kaynakları dikkate alarak tasarlanmalıdır.

iii. Rezervasyon ve tahsis:

Yeni bir iş, kabul kontrol testlerinden başarıyla geçtikten sonra, o iş için gereken kaynakların tahsisi yapılır. Bu şekilde, istenen servis kalitesinin sağlanacağı bir ortam yaratılmış olunur. Rezervasyon, kaynakların en kötü ya da ortalama kullanım durumuna göre yapılır. Eğer en kötü kullanım durumları dikkate alınıyorsa servis kalitesi kesin olarak garantilenir. Ancak, uygulamaların hepsi her an en kötü koşullara göre kullanılamayacağı için kaynaklar diğer yaklaşımlara göre daha az oranda kullanılır. Ortalama iş yükü kullanılarak yapılan rezervasyonlarda daha çok işe kaynak verilir. Ancak, sürekli olarak sistemde ortalama iş yükünün aşılıp aşılmadığı kontrol edilmelidir.

iv. İş dağıtımı:

Kaynak tahsisi gerçekleştirilen uygulamalar uygun CPU ve ağ iş dağıtımı mekanizmaları kullanılarak aktif hale getirilir.

v. İzleme ve uyum:

Bağlantı kurma aşamasında belirlenen servis kalitesi değerleri, işletim sırasında yetersiz kalabilir. Kaynak yönetimi mimarisi, bu durumların nedenlerine uygun çözümlere sahip olmalıdır. Uygulamalar, işleme başlamadan önce, gerçek servis kalitesi değerleri bilinmeyeceği için kabul kontrol testlerinde tahmini değerler kullanılır. İşletim sırasında belli aralıklarla uygulamaların çalışması izlenir ve gerçek servis kalitesi gereksinimi tahmini değerlerle karşılaştırılır. Gerçek değerler tahmini

değerlerden farklı ise, geri besleme yolu ile uygulamanın kaynak ihtiyaçlarını gözden geçirmesi istenir. Servis kalitesi değerleri ile ilgili değişiklik gereksinimi kullanıcıdan da gelebilir. Bu durumda, kabul kontrol, kaynak rezervasyonu ve kaynak tahsisi işlemleri yeni değerlere göre tekrar yapılır. İstek, herhangi bir uygulamanın beklenmeyen olumsuz davranışı yüzünden işletim sistemi kaynaklarının yetersiz kalması nedeniyle işletim sisteminden de gelebilir. Bu durumda, bu işin bağlantı kurma aşamasında kabul edilen değerlere uyması sağlanır, ya da kaynakların bir kısmı, performansı düşürme pahasına, diğer uygulamalardan geri alınarak bu işe verilir. Eğer değişiklik isteği ağdaki aşırı yüklenmeden geliyorsa, yük dengeleme, geri besleme yoluyla kaynak hızını değiştirme (rate control) veya ölçeklendirme (scaling) teknikleriyle yeni duruma uyum sağlanır.

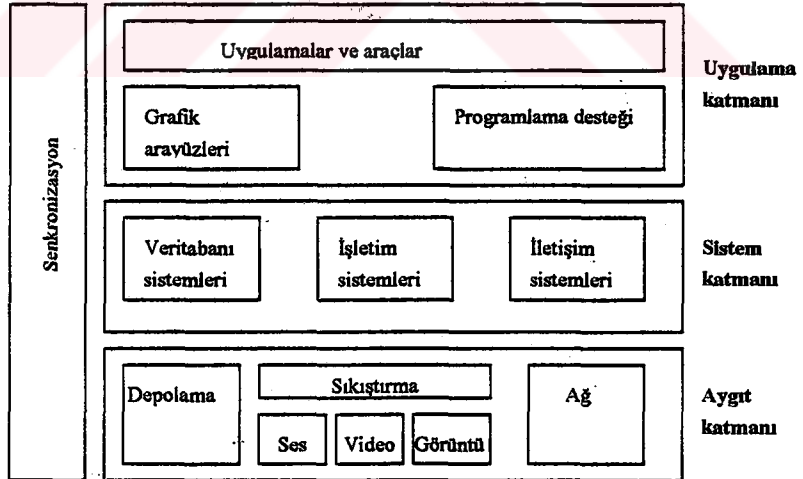
Kaynak yönetimi konusu ile ilgili literatüre geçmiş çalışmalar bulunmaktadır. Bunlar arasından Pennsylvania Üniversitesi'nde geliştirilen OMEGA mimarisi (Nahrstedt, 1995), Berkeley Üniversitesi'nde geliştirilen TENET mimarisi (Campbell, 1995), Heidelberg European Networking proje grubu tarafından geliştirilen HEITS mimarisi (Campbell, 1995) ve Lancaster Üniversitesi'nde geliştirilen QoS-A mimarisi (Campbell, 1995) kendilerinden sonra gelen kaynak yönetimi çalışmalarına öncelik etmişlerdir.

2.5 Çokluortam Sistem Mimarileri

Yüksek veri hızı ve işletim gücü gereksinimleri, ilk çokluortam sistemlerinin donanım tabanlı olmasına yol açmıştır. Örneğin, Cambridge Üniversitesi tarafından geliştirilen ve dünyadaki ilk çokluortam sistemi olarak kabul edilen Pandora projesi ve Lancaster Üniversitesi tarafından

geliştirilen MNI projesi çokluortam işlemlerini Pandora kutusu adı verilen ana işlemciden bağımsız özel bir aygıtı yönlendirir (Burkow, 1994). Cambridge Üniversitesi tarafından geliştirilen Pegasus dağıtık çokluortam sistemi de büyük ölçüde donanım tabanlıdır. Pegasus mimarisi, video verilerini ağa yerleştirilmiş ve otomatik olarak ATM paketleri üreten ATM kameralardan alır ve DAN (Desk Top Area Network) adlı paketlerin bellekte yerleşeceği yeri belirleyen arayüzlere iletir. DAN arayüzleri verileri işletim sistemi tarafından işleminden geçirilmeden doğrudan video RAM'e gönderir (Leslie et al., 1992).

Donanım tabanlı yaklaşımlar sisteme hız kazandırmalarına rağmen pahalı ve veri üzerinde işlem yapmaya olanak sağlamadıkları için esnek olmayan yaklaşımlardır. Bu nedenle işlemci hızının artmasıyla birlikte yazılım tabanlı yaklaşımlara da yönelinmiştir. Sonuç olarak, tıpkı OSI (Open Systems Interconnection) mimarisinde olduğu gibi herbiri bir alt katmana servis sağlayan katmanlı bir çokluortam sistem mimarisi ortaya çıkmıştır. Şekil 2.2'de gösterilen bu mimari temel olarak üç katmandan oluşur (Steinmetz and Nahrstedt, 1995).



Şekil 2.2. Bir çokluortam sistem mimarisi

En üst seviyede yer alan Uygulama Katmanı, çokluortam programlama kütüphaneleri, işletim ve iletişim sistemi arayüzler ve grafik arayüzlerinden oluşur.

Sistem Katmanı işlemci, bellek, aygıt ve iletişim kaynaklarının yönetiminden sorumlu işletim sistemi, çokluortam verilerinin yönetim ve erişim işlemlerinden sorumlu veritabanı sistemi ve çokluortam verilerinin belli zaman kısıtları içerisinde iletiminden sorumlu bir iletişim sistemi olmak üzere üç elemandan oluşur.

En alt seviyede yer alan Aygıt Katmanı, kayıt aygıtları, depolama aygıtları, iletişim ortamı, bilgisayar sistemleri ve sunum aygıtlarından oluşur. Çizelge 2.2’de bu aygıtlara örnekler verilmektedir.

Çizelge 2.2. Aygıt katmanında yer alan elemanlar

Aygıt	Örnekler
Yakalama aygıtları	Video kamera, mikrofon, klavye, fare vb.
Depolama aygıtları	Disk, CDROM, DVD
Ağ	Ethernet (10-100 Mbps) Token Ring (100 Mbps) FDDI (100 Mbps) ATM (maksimum 1 Gbps)
Bilgisayar sistemleri	Pentium işlemciler, iş istasyonları, MPEG, DSP donanımları vb.
Sunum Aygıtları	HDTV, SVGA Hires monitor, hoparlör vb.

Bu mimaride farklı veri türlerine ait zaman ilişkilerini düzenlemekten sorumlu eşzamanlama modülü, tüm katmanlarla ilişkili olduğu için tek bir katmanla sınırlı kalmamaktadır.

2.6 Çokluortam Uygulamaları İçin Sistem Desteği

2.6.1 İşletim sistemi desteği

2.6.1.1 İşletim sistemi seçimi

Geleneksel işletim sistemleri, uygulamaların zaman kısıtlarını dikkate almayarak kaynakları tüm uygulamalara eşit miktarda dağıtarak sistem verimini arttırmayı hedefler. Bu yaklaşım, çokluortam verilerinin doğasına aykırıdır. Çokluortam verileri, gerçek zaman kısıtları olan verilerdir. Bu yüzden çokluortam uygulamaları ilk önce Yartos, ARTS gibi geleneksel katı gerçek zamanlı işletim sistemleri üzerinde geliştirilmiştir. Çokluortam uygulamalarının esnek ve dinamik yapısı nedeniyle, bu tür işletim sistemlerinin de çokluortam uygulamaları için elverişli olmadığı anlaşılmıştır. Günümüzde eğilim, gerçek zaman olanakları bulunan genel amaçlı işletim sistemleri üzerinde çokluortam uygulamaları geliştirmektir. Çokluortam verilerinin esnek gerçek zamanlılık özelliği ile uyum içinde bulunan bu yaklaşımın avantajları şöyle özetlenebilir (Adelberg 1994; Kantarcı and Tunalı, 1998a; Kantarcı and Tunalı, 1998b):

1. Genel amaçlı işletim sistemleri üzerinde yazılım geliştirme, daha geniş bir programcı kitlesine hitap ettiği için katı gerçek zamanlı sistemler üzerinde yazılım geliştirmekten daha kolaydır

2. Genel amaçlı işletim sistemleri, gerçek zamanlı işletim sistemlerinden daha yaygın ve ucuzdur.
3. Genel amaçlı işletim sistemleri üzerinde geliştirilmiş uygulamaların taşınabilirliği daha yüksektir.
4. Katı gerçek zamanlı işletim sistemleri, sadece gerçek zamanlı uygulamaları desteklerken genel amaçlı işletim sistemleri hem gerçek zamanlı, hem de gerçek zamanlı olmayan uygulamaları desteklerler.

Genel amaçlı işletim sistemlerine gerçek zamanlılık özelliği kazandırmak üzere POSIX 4 standartları oluşturulmuştur. Genel amaçlı işletim sistemleri içinde, bu konuda günümüzde verilebilecek en iyi örnek, yüksek işletim gücüne sahip Sun Ultra Sparc iş istasyonları üzerinde çalışan POSIX 4 desteğine sahip Solaris işletim sistemidir. Günümüzde pek çok UNIX türevi işletim sisteminin gerçek zamanlı programlama desteği bulunmaktadır (Gallmeister, 1995).

2.6.1.2 Mikroçekirdek mimari

Çokluortam uygulamaları için işletim sisteminin tüm fonksiyonlarını kapsayan monolitik çekirdekler yerine sistemin sadece en alt düzeyde fonksiyonlarını içeren mikroçekirdekler tercih edilmelidir (Colouris, 1994). Monolitik çekirdeklerin fazla yer kaplama, sisteme bir fonksiyon ekleme durumunda çekirdeği tekrar derleme gerekliliği gibi sorunlar vardır. Mikroçekirdek yaklaşımında iş dağıtımı, bellek yönetimi, kesinti yönetimi gibi temel fonksiyonları içeren küçük bir çekirdek bulunur. Dosya yönetimi, senkronizasyon, terminal yönetimi işlemleri ayrı servisçilerce yerine getirilir. Mikroçekirdekler, monolitik

çekirdeklere oranla daha esnek, daha güçlü, genişleyebilir, taşınabilir, tekrar kullanılabilir, hataya dayanıklı işletim sistemlerinin oluşturulmasını sağladıkları için daha avantajlıdır (Mullender, 1993).

2.6.1.3 İş dağıtımı

Çoklu ortam işletim sistemleri işlere öncelik vererek zaman kısıtlarının sağlanmasını amaçlarlar. Önceliklerin belirlenmesi ve uygun iş dağıtımı algoritmalarının seçilmesi karar verilmesi gereken önemli konulardır. İşlerin gerçek öncelikleri sınırsız sayıdaki doğal sayılarla ifade edilirler. Günümüz işletim sistemleri, sınırlı sayıda öncelik seviyesine sahiptir. Dolayısıyla gerçek önceliklerin uygun bir şekilde işletim sistemine ait sınırlı sayıda önceliklerle eşleştirilmesi gerekmektedir. Öncelik belirleme işlemi, işlerin doğal özelliklerinin işletim sistemi öncelikleri cinsinden ifadesidir. Öncelik belirleme işleminin sağlıklı gerçekleştirilmesi, işlerin gerçek zaman kısıtlarının sağlanmasında önemli rol oynar (Katcher and Strosnider, 1993; Katcher et al, 1995).

Öncelikler, işlerin periyodu ve bir periyod içindeki işletim süresi gibi parametrelere bakılarak belirlenir. Çoklu ortam uygulamaları; bir anda kaç kişinin sisteme katılacağı, sistemden ayrılacağı tahmin edilemeyen ve işletim sırasında değişken miktarda kaynak ihtiyacı olan uygulamalardır. Bu yüzden, işlere öncelik atama önemli bir işlemdir (Fisher and Keller, 1995).

Gerçek zaman desteği veren işletim sistemleri statik öncelikler kullanırlar. Diğer bir deyişle, iş sisteme katıldığında ona bir öncelik değeri verilir ve bu öncelik işletim sırasında değiştirilmez. Teorik

yaklaşımlara göre önceliklerin işletim sırasında değiştirilebildiği dinamik yaklaşımlar daha avantajlıdır. İşletim sistemlerinin de bu amaçla kullandığı komutlar bulunmasına rağmen, “yeni süreç yükleme” (context switching) işleminin getirdiği yük ve bir seferde sadece tek bir işin önceliğinin değiştirilebilmesi gibi nedenlerden ötürü statik yaklaşımlar tercih edilmektedir (Jensen, 1996).

İş dağıtımı algoritmaları, işlerin zaman kısıtlarını çığnemeyecek şekilde bir iş dağıtımı planı bulmak zorundadır. Genelde, çoklu ortam uygulamaları için RM (Rate Monotonic) ve EDF (Earliest Deadline First) olmak üzere iki temel algoritmadan biri tercih edilmektedir. RM, periyodu en küçük olan işlere en yüksek, EDF kritik bitiş zamanı en erken işlere en yüksek önceliği verir. İki algoritma da kesintili algoritmalar. Kesintiler, ‘kontekst’ değiştirme işlerine yol açarlar. ‘Kontekst’ değiştirme, tahmin edilebilirliği olumsuz etkilediği için en aza indirilmeye çalışılan bir işlemdir. Çalışmalar, EDF’nin RM’den daha az sayıda ‘kontekst’ değişimine neden olduğunu, buna karşılık RM’in daha kararlı ve gerçekleştiriminin daha kolay olduğunu göstermektedir (Pingali, 1994). RM’in ve EDF’nin avantajlarını birleştiren bazı yaklaşımlar da geliştirilmiştir (Kantarıcı and Tunalı, 1998a; Kantarıcı and Tunalı 1998b).

2.6.1.4 Bellek ve tampon saha yönetimi

Geleneksel gerçek zamanlı sistemler, sayfalama hataları nedeniyle meydana gelebilecek gecikmeleri ortadan kaldırmak için ‘kilitleme’ mekanizması kullanarak verilerin sürekli bellekte kalmasını sağlarlar. Yüksek veri hacmi nedeniyle bu yaklaşımı çokluortam uygulamaları için kullanmak sakıncalıdır (Kantarıcı and Tunalı, 1997).

Sürekli veri işlemleri sırasında karşımıza çıkan başka bir problem, veri kopyalama işleminin neden olduğu zaman kayıplarıdır. Kopyalama fonksiyonları yerine “pointer” atamalarının kullanılması, ‘copy on write’ tekniği ile ortak verilerin modüller arasında ya da süreçler arasında paylaşımı, ağdan gelen verilerin işletim sistemi tamponlarına yönlendirilmeden doğrudan uygulamaya gönderilmesi kopyalama işlemini azaltır (Colouris, 1994).

Çokluortam uygulamaları, bir ‘pipeline’ düzeninde birbirini izleyen modüllerden oluşur. Verilerin kaynaktan alınması, iletişime hazır hale getirilmesi, ağa gönderilmesi, ağdan alınması, açılması ve kullanıcıya sunulması işlemleri birbirini izleyen modüller tarafından gerçekleştirilir. Her bir modül, bir önceki modülün çıktılarını girdi olarak alır, işler ve çıktı olarak bir sonraki modüle aktarır. Modüllerin hızlarının ve kullandıkları veri miktarının farklı oluşu, modüller arasında tampon saha kullanım gereksinimi ortaya çıkarır. Tampon saha kullanımının başka bir amacı da geliş zamanları arasındaki varyasyonun azaltılarak sistemin daha kararlı bir hale getirilmesidir. Örneğin, 30 çerçeve/saniye hızda çalışan bir video oynatıcısı her çerçeveyi 33 milisaniye aralıklarla ekrana vermelidir. Verilerin erken ya da geç gelmesi ardışık çerçevelerin ekrana verilmesi zamanları arasındaki farkı değişken kılacaktır. Dolayısıyla bazı çerçeveler çok uzun bir süre ekranda kalarak duraklamalara neden olacaktır. Bu nedenle, gösterime başlamadan önce, bir miktar verinin tampon sahalarda tutulması ve zamanı gelince oynatılması gerekmektedir. Tampon miktarının belirlenmesi de önemli bir konudur. Sabit miktarda tampon sahası tahsisi yapmak yerine verilerin özellikleri, uygulama sırasında gözlenen gecikmelerin varyansı dikkate alınarak esnek tampon yönetimi mekanizmaları kullanmak daha avantajlıdır. Tampon saha ve bellek yönetimi, süreçler arası iletişim, paylaşılan bellek

ve senkronizasyon kontrolü gibi konularda yeterli olanakların bulunmasını gerektirir (Chatterjee et al., 1995).

Veri kopyalama sorunu, tampon saha işlemlerinde de karşımıza çıkar. Genel olarak, her bir çokluortam 'pipeline' safhası, verilerin önüne özel bir başlık yerleştirir. Bu durum, safhalar arasında sürekli veri kopyalamayı gerektirir. Bu kopyalamayı azaltmak için iki teknikten biri kullanılabilir (Nahrstedt, 1995). Offset kontrolü adlı ilk teknikte tampon saha için tek bir bellek "segment"i tahsis edilir. Bellekte, tampon sahaya yerleştirilecek tüm veriler ve başlıkları kapsayacak genişlikte yer ayrılır. Her modül, kendi başlık sahasına karşılık gelen sahaya başlığını yerleştirir. Bu yöntemin problemi, tüm başlık sahalarını kapsayacak büyüklükte yer ayırma zorunluluğudur. Bu da tüm başlıkların büyüklüğünü bilmeyi gerektirir. 'Scatter/gather' yönteminde veriler ve her başlık için ayrı bir bellek segmenti kullanılır. Bu yaklaşım, kullanılan tampon sahanın büyüklüğünü dinamik olarak değiştirebilir ve farklı veri ünitelerinin kolayca birleştirilmesi ya da birbirinden ayrılmasını sağlar (Nahrstedt, 1995).

2.6.1.5 Disk ve dosya sistemleri

Uygulamaların sürekliliğini sağlama açısından, her biri yüksek bant genişliği ihtiyacına sahip birden fazla akışa ait verilerin aynı anda okunması ve ağa gönderilmesini sağlayacak uygun dosya ve disk yönetimi mekanizmaları kullanılmalıdır. Çokluortam verilerinin geleneksel veriler gibi, diske geliş güzel yerleştirilmesi erişim zamanını yükseltir. Aynı anda oynatılacak verilerin disk üzerinde birbirine yakın sahalarla yerleştirilmesi konusunda çalışmalar yapılmaktadır. Diske erişim için EDF ve SCAN-EDF türü yaklaşımlar kullanılabilir. Birden

fazla diske paralel erişim yapmayı sağlayan RAID disk mimarilerinin kullanımı ve 'disk striping' mantığında verilerin diske yerleştirilmesi hem disk erişimi için gereken süreyi düşürür, hem de aynı anda desteklenebilecek kullanıcı sayısını artırır (Keeton et al., 1995; Kunii 1995).

2.6.1.6 Programlama desteği

i. Gerçek zamanlı süreçler arası iletişim (SAİ) ve Senkronizasyon:

Çokluortam uygulamalarına ait süreçlerin birbirleriyle iletişimi için gerçek zamanlı SAİ ve senkronizasyon desteği kullanılmalıdır. Geleneksel uygulamalarda SAİ ve senkronizasyon desteği çekirdek tarafından sağlanır. Gerçek zamanlı uygulamalar için, SAİ işlemi sırasında çekirdek seviyesine inmenin getirdiği yükten kurtulmak amacıyla kullanıcı seviyesinde bulunan paylaşılan bellek, mesaj kuyrukları gibi SAİ olanakları ve mutex, semafor gibi senkronizasyon olanakları sağlayan SAİ ve senkronizasyon kütüphaneleri geliştirilmiştir (Raghavan et al., 1998).

ii. Gerçek zamanlı I/O yönetimi:

Geleneksel gerçek zamanlı sistem tasarımında katı gerçek zaman garantileri sağlayabilmek için I/O olaylarının işlem hızının mümkün olduğu kadar artırılmasına çalışılır. Çokluortam uygulamalarında ses ve video gibi farklı akışlar arasında senkronizasyon sağlanması gerektiği için verilerin doğru zamanda ağdan ya da diskten alınarak işleme konması gerekir. Bunun için sıklıkla kullanılan bir yaklaşım, verilere zaman etiketlerinin yerleştirilmesidir (Buford, 1994).

iii. Gerçek zamanlı iş parçacığı (thread) programlama:

İş parçacıkları süreç içinde belli bir aktiviteyi yerine getiren birimlerdir. Bir süreç içinde bulunan iş parçacıkları, sürecin adres sahası aracılığı ile SAI mekanizmaları kullanmadan haberleşebilirler. Birden çok CPU'nun bulunduğu durumda birden fazla iş parçacığı aynı anda çalışarak paralellik yaratılabilir. Aynı adres sahası içindeki iş parçacıkları arasında geçiş yapmak, süreçler arasında geçiş yapmaktan daha az harcama gerektirir. Ayrıca iş parçacığı yaratmak, süreç yaratmaktan daha ucuzdur. İş parçacıkları yönetimi çekirdek ve kullanıcı seviyesinde bulunan veri yapıları ve kütüphanelerle gerçekleştirilir. Kullanıcı seviyesi iş parçacıklarını kullanmak daha az külfetlidir. Ancak, girdi çıktı, kesinti ve görünür bellek vb. işlemler için çekirdek seviyesi iş parçacıkları kullanılır (Lewis and Berg, 1998).

2.6.2 İletişim desteği

Dağıtık çokluortam uygulamaları geleneksel veriler yanında çokluortam verilerinin de bilgisayarlar arasında iletişimini gerekli kılarlar. Bu durum çokluortam verilerinin yüksek bant genişliği, sınırlı 'jitter', gerçek zamanlılık, senkronizasyon gibi gereksinimlerini dikkate alarak yeni servis ve protokollerin tasarlanmasına neden olmuştur. Bu protokollerin mevcut protokollerle uyumlu olması, hatta mevcut protokollerden de yararlanması gerekmektedir.

2.6.2.1 Çokluortam uygulamaları açısından OSI katmanları

i. Alt seviye katmanları:

Alt seviye katmanları dağıtık çokluortam ile ilgili olarak iki önemli rol üstlenirler. Bunlar, verilerin bir yerden bir yere iletim hızıyla ilgili olan bantgenişliği olanakları (bandwidth availability), diğeri de bantgenişliğinin birçok kullanıcı tarafından nasıl paylaşılacağı ile ilgili olan bantgenişliği tahsis (bandwidth allocation) işleridir (Hodges and Sasnett, 1993).

Bant genişliği olanakları: Çokluortam ağları büyük miktarda veri transferi gerçekleştirmek zorundadırlar. 40 sayfalık bit metin dökümanı yaklaşık 1 Mbitlik bir bantgenişliği kullanımı gerektirir. 10 Mbpslik bir Ethernet ağından saniyede bu tip dökümanlardan 10 adet gönderilebilir. Diğer taraftan sıkıştırılmamış sayısal video verileri 640x480 lik çerçeve, 24 bit derinlik ve 30 çerçeve/sn aktarım hızı gözönüne alındığında saniyede 220 Mbitlik bantgenişliğine ihtiyaç duyarlar. 10 Mbpslik Ethernet ağı metin verilerini sorun çıkarmadan taşıyabildiği halde, video verilerinin istenen düzeyde iletimini sağlayamamaktadır. (Hodges and Sasnett, 1993).

Bantgenişliği tahsisi: Bazı veri türleri çok yüksek miktarda bantgenişliğine gereksinim duyarken, diğerleri için küçük bir miktar yeterlidir. Bantgenişliğine ihtiyaç duyulan en yüksek seviyede tahsis yapılırsa, bantgenişliğinin büyük bir kısmının hiç kullanılmama durumu ortaya çıkar. Bazı ağ mimarileri bantgenişliğine gereksinim duyulduğunda kullanılmak üzere dinamik bir şekilde tahsis ederler. Bu şekilde kaynakların verimli bir şekilde kullanılması sağlanır. ATM bu tip sistemlere iyi bir örnektir. ATM'in çokluortam uygulamalarında popüler olmasının nedenleri şu şekilde özetlenebilir:

1. ATM ağları üç farklı veri tipini desteklemektedir: Sabit Bit Hızı (Constant Bit Rate-CBR) trafik, Değişken Bit Hızı (Variable Bit Rate-VBR) trafik ve Mevcut Bit Hızı (Available Bit Rate-ABR) trafik. CBR trafik ATM ağına sürekli aynı hızda erişen trafik için kullanılan terimdir. Bu iletişim garantili bağlantı, sabit bantgenişliği ile düşük gecikme koşullarının sağlanmasını gerektirir. Sıkıştırılmamış ses ve video iletimi bu tip trafik için verilebilecek örneklerdendir. VBR ile iletişim hızı değişkenlik gösterebilir. Sıkıştırılmış video bilgileri bu tür trafikle iletilir. ABR trafik, CBR ve VBR trafikten arta kalan bant genişliğini kullanır, metin verilerinin iletimi için uygundur.
2. ATM anahtarlama bir teknolojidir. Anahtarlama bağlantısının, ortak erişimli medyaya göre üstünlükleri vardır. Anahtarlama bağlantılarda bantgenişliği paylaşılması yerine bağlantıya adanmış bantgenişliği sağlanır. Dolayısıyla toplam bantgenişliği, bağlantı yapıldıkça yükselir.
3. Hata kontrolünün sadece uç noktalarda yapılmasından dolayı ara düğümlere düşen yük azalmıştır. Bu da sistemin hızını artırır.
4. ATM, hem LAN hem WAN'larda kullanılan bir tekniktir. ATM tabanlı uygulamalar ilk önce LAN larda denenip sonra WANlara genişletilebilir.
5. 'Rate based control' ve 'credit based control' gibi akış kontrol yöntemleriyle veri kaybı sorunu büyük ölçüde çözülmüştür (Kantarci and Tunali, 1996).

ii. Üst Seviye Katmanları:

Nakil (transport) katmanı: Uçtan uca bağlantının sağlandığı katmandır. Alt katmanların kötü servis kalitesine sahip olması durumunda, nakil katmanı gerekli servis kalitesini sağlamak amacıyla kontrolü ele alır. Büyük paketler bu seviyede küçük pakelere ayrılır. Alıcı tarafta ağdan gelen paketler bu seviyede birleştirilerek orijinal veriler elde edilir.

Oturum katmanı: İstasyonlar arasında bağlantının sağlanması ve farklı akışların senkronizasyonu bu katmanda gerçekleşir.

Sunum katmanı: Format dönüşümü, veri kodlama gibi işlemleri yerine getirir. Servis kalitesi üzerinde tarafların anlaşması da bu katmanda yerine getirilir. Taraflar sıkıştırma işlemi için gerekli kararları bu katmanda verirler.

Uygulama katmanı: Elektronik ileti, dosya transferi gibi uygulamaya özgü işlemler bu katmanca yerine getirilirler. Kullanıcı istediği servis kalitesini bu katmanda verir (Kantarıcı et al., 1996).

2.6.2.2 Internet protokolü

DARPA'nın çabaları ile 1980 li yıllarda dünyadaki tüm bilgisayarları birbirine bağlayabilen Internet ağ sistemi ortaya çıkmıştır. IP protokolü OSI modelini temel almış olsa da Şekil 2.3 deki IP mimarisinde bazı OSI katmanlarının tek bir katmanda toplandığı görülmektedir (Tan, 1999).

Uygulama katmanı (SNMP, FTP vb.)
Transport katmanı (RTP, TCP, UDP)
Ağ katmanı (RSVP, ST II, ICMP, IGMP, IP)
Veri bağlantı katmanı

Şekil 2.3. Internet Protokolü katmanları

2.6.2.3 BSD (Berkeley Software Distribution) Unix IP gerçekleştirimi

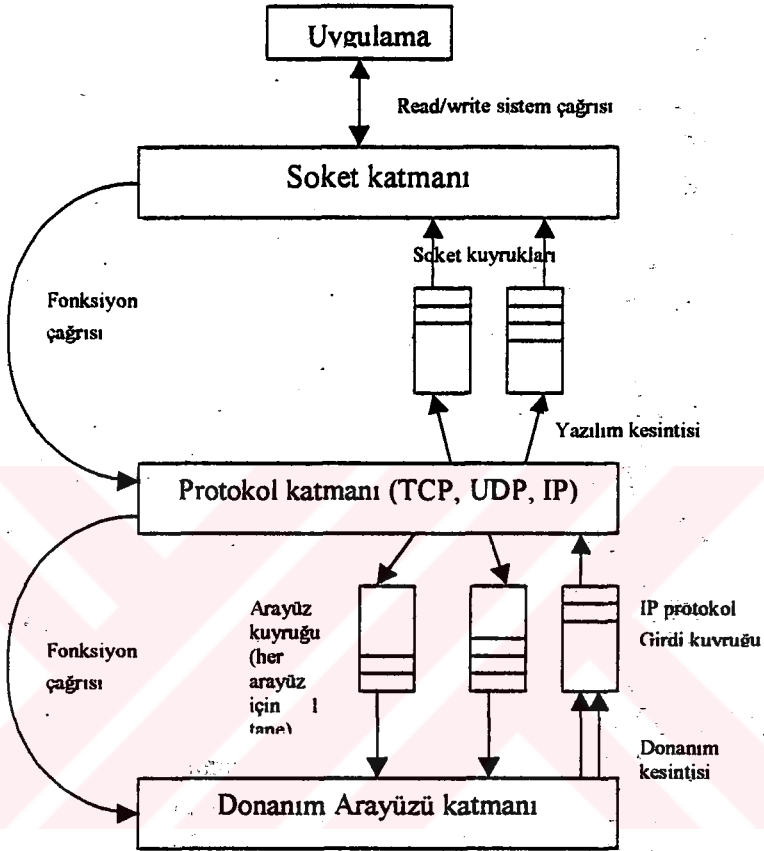
BSD Unix IP yığıtı, soket kütüphanesi ile birlikte bir çok ağ sisteminin temelini oluşturur. Uygulamalar soket arayüzü vasıtasıyla iletişim servisleri ile bağlantı kurarlar. Soket arayüz kütüphanesi TCP (Transmission Control Protocol) ve UDP (User Datagram Protocol) bağlantıları kurmaya, veri göndermeye ve almaya yönelik olanaklara sahiptir. Bir sokete gönderilen veriler mbuf adı verilen küçük bellek parçalarında saklanırlar. Mbuf'lar verilerin kopyalama işlerini en aza indiren protokol başlıkları eklenmesine elverişli şekilde tasarlanmışlardır (Stevens 1998, Stevens 1999).

Üst katmanlarda bulunan protokoller bir donanım arayüzü ile donanımı kontrol ederler. Çıktı işlemi çekirdeğe yapılan bir sistem

çağrısıyla eş zamanlı bir şekilde ya da bir kronometre süresinin dolması sonucu gerçekleşir. Girdi işlemi ise asenkron bir işlemdir. Veri gelişiyile gerçekleşen bir donanım kesintisi, gelen paketin bir mbuf'a alınmasını sağlar. IP paketleri IP katmanı kuyruğunda sıralanarak *Önce gelen önce servis alır* (first come first served, FCFS) mantığı ile işleme alınırlar. Tüm paketler tek bir IP kuyruğu kullanırlar. Şekil 2.4' de BSD UNIX'in IP protokol katmanları arasındaki iletişim gösterilmektedir.

Pek çok UNIX ağ çekirdeğinin temeli olan BSD ağ yığıtı, çokluortam sistemleri için bazı dezavantajlara sahiptir (Tan, 1999):

1. Tüm ağ trafiği için sistemde tek bir öncelik seviyesi bulunduğundan tüm paketler aynı önceliğe sahiptir. Farklı paketlere farklı öncelikler vermek mümkün değildir.
2. BSD UNIX, CPU kesintileri kullandığı için tek işlemcili sistemlerle sınırlıdır. Çok işlemcili sistemler için geliştirilen BSD versiyonları bir seferde sadece tek bir işlemcinin protokol sistemini kullanmasını sağlayacak şekilde karmaşık kilit mekanizmasına sahiptir.
3. UDP ve IP protokolleri için tek bir kuyruğun kullanılması paketlerin sadece FCFS mantığıyla işleme alınmasına yol açar.
4. Soket arayüzü, servis kalitesi ve kaynak rezervasyonu ile ilgili hiçbir fonksiyon içermez.



Şekil 2.4. BSD UNIX IP gerçekleştirimi

2.7 Çokluortam Verilerinin İnternet Üzerinden İletimi

Yeni işlem ve iletişim teknolojilerinin ortaya çıkışıyla ses/video konferansları, bilgisayar destekli eğitim, istek üzerine video, etkileşimli TV, endüstriyel kontrol ve etkileşimli oyunlar gibi uygulamalar yaygınlaşmaya başlamıştır. Bu uygulamaların ortak bir yönü, dağıtık uygulamalar olmalarıdır. Metin bilgileri taşıyan geleneksel ağlar, çokluortam verilerini taşımada yetersiz kalmaktadırlar. Çokluortam

uygulamaları, uygulama geliştirmeyi zorlaştıran yüksek bant genişliği gereksinimi, gerçek zamanlı iletişim ve işlem gereksinimi gibi özelliklere sahiptir.

Çokluortam verilerinin taşınması için mevcut alternatifler arasında ilk akla gelen, uygulamaya adanmış donanım ve yazılım desteğine sahip ortamlardır. Bu alternatiflerin ortak dezavantajları pahalı olmaları ve bir çok kullanıcı tarafından paylaşılamamalarıdır. Paketleri zamanında iletmeyi ve ihtiyaç duyulan bant genişliğini garantileyen ATM tabanlı çözümlerse hem pahalıdır, hem de yeterince yaygınlaşmamıştır.

Geleneksel paket anahtarlama ağı ve LAN teknolojileri, uygulamaların ağa sürekli olarak gereksinim duymadığı varsayımına dayanarak pahalı altyapıların paylaşımını mümkün kılarlar. Fakat çokluortam uygulamaları, ağ üzerinde sürekli büyük bir yük yarattıklarından geleneksel paket anahtarlama ve LAN ağlarını elverişsiz hale getirirler.

ATM satıcılarının, bugünün paket anahtarlama ağıyla gerçek zamanlı verilerin taşınamayacağını iddia etmelerine rağmen, mevcut teknolojilere bir bakış, uygun yazılımların kullanımıyla çokluortam verilerinin geleneksel ağlarla entegre edilebileceğini ortaya çıkarmaktadır. İnsan gözü ve kulağının gecikme ve bozulmalara karşı belli bir dereceye kadar duyarlı olması nedeniyle, çokluortam verileri mevcut paket anahtarlama ağlarda ATM kalitesine yakın bir kalitede taşınabilmektedir.

Gelecek kuşaklarda en ideal ağ ortamının tüm veri türlerini entegre edebilen ve dünyanın her yerine taşıyabilen bir ağ ortamı olacağı şüphesizdir. Sürekli genişleyen İnternet, bu tip bir ağ ortamıdır. IP

protokolüne dayanan mevcut LAN ve WAN teknolojileri, tüm ağları İnternete bağlayabilmektedir. İnternet, farklı ağları birbirine bağlayarak pahalı donanım ve arayüz kullanım ihtiyacını ortadan kaldırmaktadır. Aynı zamanda, uygulamalara her veri türünün entegre edilmesini de sağlamaktadır.

İnternet paylaşımlı bir “datagram” ağı olduğu için gerçek zamanlı trafik desteği yoktur. İnternet üzerinden çokluortam uygulamalarını çalıştırabilmek için İnternet üzerinde bazı değişikliklerin yapılması gerekmektedir. İlk olarak, mevcut altyapı çok yoğun veri miktarı ve veri trafiğini destekleyecek bant genişliğini sağlayabilmelidir. Güvenilirliği sağlamaya yönelik yöntemler çokluortam verilerinin gerçek zamanlılık özelliği ile çelişki yarattığı için güvenilir olmayan iletişim protokolleri kullanılmalıdır. Kullanılacak iletişim protokollerinin video konferansları gibi uygulamaları destekleyebilmesi için bir paketin birden çok işlemciye aynı anda gönderilmesini (“multicast”) sağlaması gerekmektedir. Kullanılacak protokollerin aynı zamanda iletişimle ilgili zamanlama ve senkronizasyon işlemleri için de olanakları bulunmalıdır.

Başka bir önemli nokta, çokluortam uygulamalarının tamamen yazılım temelli yaklaşımlar üzerine oturtulması gerekliliğidir. Örneğin, ATM ve MPEG kartlarının pahalı oluşu nedeniyle İnternet üzerinde bulunan tüm bilgisayarların bu kartlara sahip olmaları mümkün değildir. Öte yandan, Pentium CPU’lar MPEG verilerinin yazılım yoluyla sıkıştırılmasını ve açılmasını mümkün kılmaktadırlar.

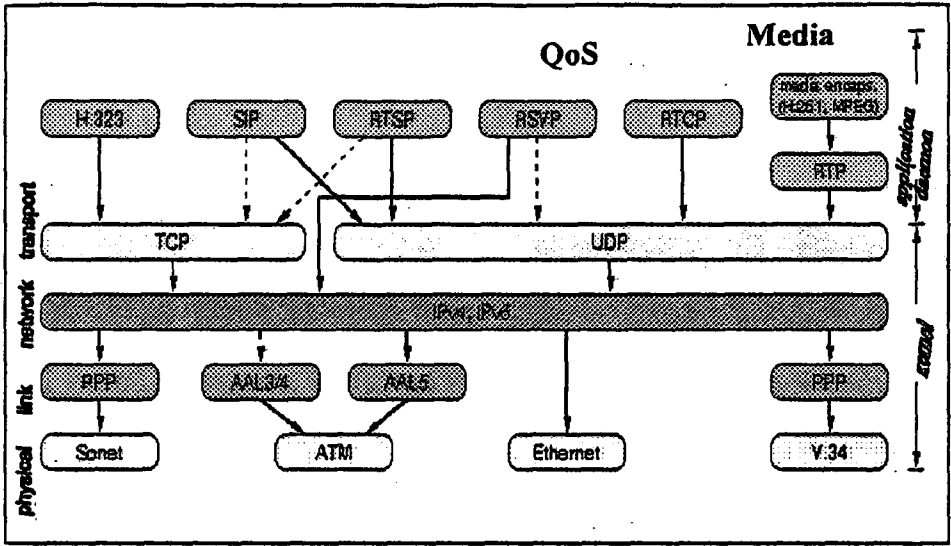
İnternet, değişik ihtiyaçlara ve özelliklere sahip her türlü trafiği destekleyebilen bir ortamdır. Örneğin, dosya transferi uygulamaları, belli miktarda verinin belli bir sürede iletimini mümkün kılarken, İnternet üzerinden telefon uygulamaları ses paketlerini 0.3 sn. içinde karşı tarafa

iletimine olanak verir. Bantgeniřliđinin yeterli olduđu durumlarda en iyi aba (best-effort servis) tm bu gereksinimleri karřılayabilir. te yandan, kaynakların kısıtlı olduđu durumlarda gerek zamanlı trafik, ađ zerinde sıklıřmalara yol aar (Chuenlei, 1998; Kantarcı and Tunalı, 1999).

WWW servisileri ve mřterileri arasında hypertext dkmanları tařımak iin kullanılan TCP tabanlı HTTP protokol ařađıdaki nedenlerden dolayı gerek zamanlı verilerin iletimi iin uygun deđildir (Cen et al. , 1995):

1. TCP'nin akıř kontrol ve pencere mekanizmalarının video ereveseleri ve ses paketleri arasındaki zaman iliřkileri zerinde olumsuz etkileri vardır.
2. TCP'nin tekrar gnderim mekanizmaları ereveseler ve akıřlar arasında zaman kaymalarına (jitter) neden olurlar. Gz ve kulađın hata ve gecikmelere karřı gsterdiđi esneklik nedeniyle gvenilir iletiřim protokollerine ihtiya yoktur.
3. TCP bađlantı tabanlı olduđu iin video konferansları gibi multicast uygulamalar iin elveriřli deđildir.

IETF (Internet Engineering Task Force), İnternet'i gerek zamanlı uygulamalar iin elveriřli hale getirmeye ynelik leklenebilir, gerek zamanlı ve multicast protokoller zerinde alıřmalar yapmaktadır. RTP (Real-time Transport Protocol), RTCP (Real-time Control Protocol), RSVP (Resource Reservation Protocol) ve RTSP (Real-time Streaming Protocol) IETF tarafından geliřtirilmiř bu protokollere rnektir (řekil 2.5). Alt blmlerde bu protokollerden sistemimizde kullandıđımız RTP ve RTCP protokolleri hakkında bilgi verilecektir.



Şekil 2.5. Gerçek zamanlı iletişim için IETF tarafından önerilen protocol hiyerarşisi

2.7.1 RTP (Real-time Transport Protocol)

1992 yılında Henning Schulzrinne tarafından RTP protokolünün ilk versiyonu geliştirilmiştir. 1995'te bir İnternet standardı olmaya aday gösterilen RTP, günümüzde standart olma yolundadır (Schulzrinne et al., 1996).

RTP, gerçek zamanlı verilerin iletimi için kullanılmasına rağmen bir iletim protokolü değildir. UDP tabanlı olarak geliştirilen RTP, hem bağlantısız hem de bağlantılı alt seviye katmanlar üzerinde çalışabilmektedir. RTP çekirdek seviyesi yerine uygulama seviyesinde geliştirilmiş bir protokoldür. RTP MPEG-1, MPEG-2, JPEG, CelB, H261 ve H263 gibi bir çok veri türünü destekleyebilmektedir. RTP, güvenilirlik için herhangi bir mekanizmaya sahip değildir.

RTP, UDP tabanlı olduğu için veri paketleri alıcıya karışık sırada gelir. Sıralama işlemi, alıcı uygulamaya bırakılmıştır. Alıcı, RTP başlıklarında bulunan sıra numaraları ve zaman etiketlerini kullanarak paketleri sıralar. Sıra numaraları aynı zamanda kayıp paketlerin belirlenmesi amacıyla da kullanılır.

RTP, “multicast” işlemler için tasarlanmasına rağmen istek üzerine video gibi “unicast” uygulamalarca da kullanılır (Schulzrinne, 1996).

RTP aynı zamanda alıcı ve göndericiler arasında çevirici ve birleştirici olarak da kullanılabilir. Çeviriciler, verileri bir veri türünden başka bir veri türüne dönüştürürler. Örneğin, video konferansı uygulamalarında bazı istasyonların yüksek, bazılarının düşük kaliteli video gösterimini destekleyebildiği durumlarda çeviriciler video kalitesini düşürmek için kullanılabilirler. Birleştiriciler bir çok akışı tek bir akış haline getirerek bant genişliğinden kazanmak için kullanılırlar. Birleştiriciler video uygulamalarından ziyade, ses uygulamaları için kullanılırlar (Thomas, 1996).

2.7.2 RTCP (Real-time Control Protocol)

RTCP, RTP ile beraber kullanılan kontrol amaçlı bir protokoldür. RTCP paketleri periyodik olarak katılımcılara gönderilirler. RTCP gönderici, alıcı, kaynak tanımlama, hoşçakal ve uygulamaya özgü paket olmak üzere beş tür rapor paketi kullanmaktadır.

Gönderici mesajlarında senkronizasyonu sağlamaya yönelik zaman etiketleri ve gönderilen paket sayısı tutulur. Alıcılar, bu bilgilere dayanarak gerçek veri hızını hesaplayabilirler. RTCP başlıkları ayrıca

paket kaybı ve ardışık paketler arasındaki zaman kayması gibi bilgileri de içerirler. Alıcılar da benzer şekilde periyodik olarak alıcı mesajları gönderirler. Alıcı raporları, gönderici raporları ile aynı sahalara sahiptir. Göndericiler, alıcı raporlarından gelen geri beslemelere dayanarak veri gönderim hızlarını ayarlarlar.

Gönderici ve alıcı raporları ağ üzerinde büyük miktarda bant genişliği tüketirler. RTCP, ağ trafiğini sabit tutacak şekilde tasarlanmıştır. Katılımcı sayısı arttıkça paket gönderim frekansı azaltılır. Bunun yanısıra rapor yaratma zamanları da rasgele olarak belirlenerek tüm katılımcıların aynı zamanda RTCP paketi göndermesi sağlanır (Schulzrinne et al., 1996).

2.8 Örnek Sistemler

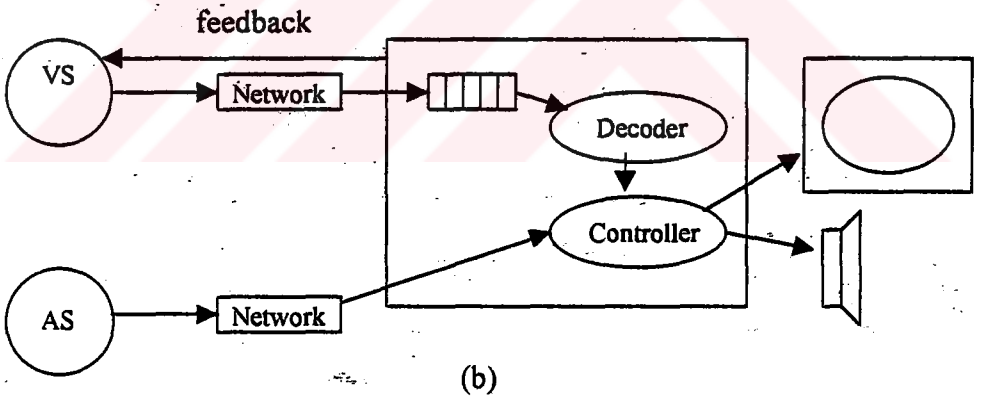
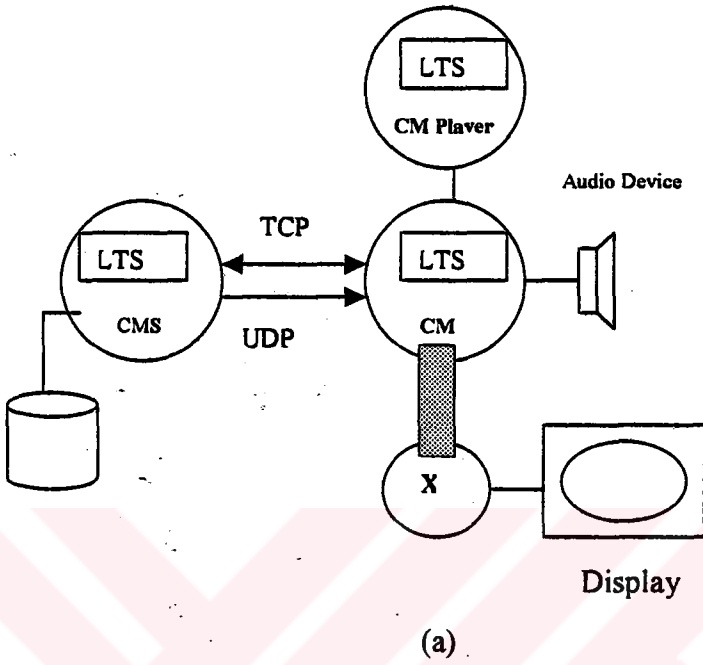
2.8.1 Sistem mimarisi

Literatürde geçen Internet üzerinden gerçek zamanlı video iletimi sistemleri (Cen et al., 1995; Rowe et al., 1994; Chen et al., 1995) süreç tabanlı olup, süreçler arası iletişim paylaşılan bellek, 'pipe' gibi mekanizmalarla sağlanır. Şekil 2.6'da Berkeley Üniversitesi'nde geliştirilen Continuous Media Player (CM) ve Oregon Üniversitesi'nde geliştirilen OGI Player mimarileri gösterilmektedir.

CM sistemi, CMT (Continuous Media Toolkit) kütüphanesi kullanılarak geliştirilmiş birbiriyle ilişki halinde bulunan yazılımlardan oluşur. Video servisçisinde bulunan CMS (Continuous Media Source) süreci ses ve video paketlerini göndermekten sorumludur. İstemcide

bulunan CMX süreci, gelen paketleri açar ve videonun doğru zamanda oynatılmasını sağlar. CMX, çerçeveleri paylaşılan belleğe yerleştirir. X Server, bu çerçeveleri paylaşılan bellekten alarak oynatır. CM Player, CMX ve CMS dağıtık LTS (Logical Time System) nesneleri ile eş zamanlanır. CMS ve CMX arasında kontrol bilgileri TCP, video ve ses bilgileri ise UDP ile taşınır.

OGI oynatıcısı bir video servisçisi (VS), bir ses servisçisi (AS), istemciler, video ve ses aygıtlarından oluşur. Video servisçisi videonun, ses servisçisi ses verilerinin iletiminden sorumludur. İstemci, bir açıcı (decoder) ve ses ve video verilerinin eş zamanlanmasından sorumlu bir denetleyiciden oluşmaktadır. Sistem tek iş parçacıklı mimariye sahip olup, servisçiler ve istemci makinalarda bulunan süreçler paylaşılan bellek, 'semaphore', 'pipe' ve sinyal mekanizmalarıyla haberleşirler. Ses verileri ve kontrol mesajları daha büyük öneme sahip olduklarından TCP protokolü ile, video verileri ise UDP protokolü ile iletilir. Alıcıda birleştirilen paketler açıcı sürece aktarılır ve Motif tabanlı bir arayüzle kullanıcıya sunulur.



Şekil 2.6 (a) Berkeley Continuous Media Player (b) Oregon OGI mimarileri

2.8.2 İletişim mekanizmaları

Mevcut sistemlerde veri iletişimi UDP tabanlı, kontrol bilgilerinin iletimi ise TCP tabanlı yaklaşımlarla sağlanır. Örneğin, Illinious Üniversitesi'nde geliştirilen Vosaic sistemi, UDP tabanlı VDP protokolünü kullanır. VDP'nin UDP'den farkı kayıp ya da geç paketlerin servisçiden tekrar istenmesidir. MBONE (Savetz et al., 1996) uygulamaları RTP protokolü kullanarak çokluortam verilerini iletirler.

UDP tabanlı mevcut protokoller, her bir çerçeveyi tek bir pakette gönderirler ya da çerçeveleri sabit büyüklükte paketlere bölerek gönderirler. Bu durum, paket ve çerçeve kaybı gibi durumlarda paketler arası bağımlılık nedeniyle kullanılabilecek maksimum veri miktarı düşmesine yol açar. RTP protokolü, büyüklüğü en büyük nakil ünitesi (maksimum transport unit-MTU) büyüklüğü ile sınırlı olacak şekilde değişken uzunlukta paketler kullanır ve farklı veri türleri için farklı veri paketleme kuralları vardır. Paketleme kuralları paketler arasındaki bağımlılığı azaltarak paket kayıplarının olması durumunda kullanılabilecek veri miktarını en yüksek düzeye ulaştıracak şekilde düzenlenmiştir.

Bant genişliği yönetim konusu üzerindeki çalışmalara da değinmek gerekir. Yukarıda tanıtılan sistemlerin hiç biri verilere ilişkin istatistikleri kullanarak akıllı bir şekilde bant genişliği yönetim işlemi gerçekleştirmez. Feng et al. (1999) saklanmış videolara ait istatistiklere dayanarak bant genişliği ihtiyaçlarındaki değişkenliği ve tampon saha kullanımını azaltacak bant genişliği yönetim sistemleri üzerinde çalışmalar yapmıştır.

2.8.3 Servis kalitesi kontrolü

Mevcut sistemlerde ağda kaybolan paketlerin yüzdesi servis kalitesinin bozulmasının en önemli göstergesi olarak kabul edilir (Herry et al., 1999). Genel olarak alıcı tarafında bir modül periyodik olarak kayıp paketler hakkında istatistikler tutarak bu istatistikleri servisçiye bildirir. Servisçi bu istatistiklere bakarak ağdaki sıkışıklık durumlarını tespit eder. Uzun süren sıkışıklık durumlarında servisçi ağdaki yükü azaltarak uyum mekanizmalarını harekete geçirir.

Vosaic oynatıcısı her 30 çerçeveyi işledikten sonra ağdaki sorunlardan kaynaklanan kayıp çerçevelerin sayısını servisçiye bildirir. CPU hızının yetersizliğinden dolayı düşürülen çerçevelerin yüzdesi sürekli kontrol edilir. Bu yüzde %5'ten küçükse servisçiden çerçeve hızını yükseltmesi, %15'ten büyükse servisçiden çerçeve hızını düşürmesi istenir.

Oregon Üniversitesi'nde geliştirilen QUASAR oynatıcısı gönderilen çerçeve hızı ve gösterilen çerçeve hızını karşılaştırır, eğer fark çok büyükse servisçiden gönderim hızını düşürmesi istenir. Farkın küçük olması durumunda, servisçinin gönderim hızını arttırması istenir.

Columbia Üniversitesi VOD sisteminde (Chang et al, 1997) göndericinin tampon sahasının doluluğu 5-10 sn. aralıklarla ölçülür. Ardışık ölçümler birbiriyle karşılaştırılır. Farkın büyük/küçük olması durumunda göndericinin çerçeve hızını düşürmesi/yükseltmesi istenir.

Mevcut duruma uyum sağlamaya yönelik önlemler arasında belirli çerçevelerin atılarak hız kontrolünün gerçekleştirilmesi (Chen et al.,

1995; Patel et al., 1997; Cen , 1995), alçak geçiren filtreleme (Yeadon et al., 1997), renk düşürme (Yeadon et al., 1997), çerçeve büyüklüğünün düşürülmesi ve transcoding (Emir et al., 1998) gelir. Li et al. (1997) hiyerarşik filtreleme tekniği ile mevcut duruma uyum sağlar. Bu yaklaşımda her MPEG çerçeve türü için farklı video akışları bulunur. Her alıcı I çerçevelerini içeren akışa abone olmak zorundadır. Ağın durumuna ya da alıcı işlemcisinin hızına göre alıcılar B ya da P akışlarına da abone olabilirler.

2.8.4 Hata kurtarma

Kayıp paketler gözden çıkarılabilir, tekrar gönderilebilir ya da FEC (Forward Error Control) mekanizmasıyla kurtarılabilir. OGI oynatıcısı kayıp paketleri dikkate almaz. CM, Cyclic UDP tekniği kullanarak MPEG videolarında I çerçevelerine karşılık gelen yüksek öncelikli verilerin tekrar gönderilmesini sağlar. Vosaic ve Columbia oynatıcıları, tekrar gönderim isteme hakkını alıcıya bırakır.

Ng Kee Yin et al. (1998) ağda kaybolan ve alıcıda düşürülen çerçevelerin sayısını dikkate alarak her alıcı için bir öncelik değeri belirler. Kayıp çerçeve sayısına göre uygulamanın hayatı boyunca bu öncelik değeri değişkenlik gösterir. Bir alıcıdaki kayıp çerçeve sayısı arttıkça o alıcının önceliği artırılır. Servisçi, alıcıları önceliklerine göre sıraya sokarak işleme alır. Düşük öncelikli alıcılara ait akışlardan çerçeveler atılarak yüksek öncelikli akışlara daha fazla bantgenişliği ayrılması sağlanır. Bu şekilde yüksek öncelikli servisçilerin servis kalitesi yükseltilmiş ve kayıp çerçeve sayısı azaltılmış olur.

2.8.5 İşletim sistemi desteği

Literatürde çokluortam uygulamalarını desteklemek amacıyla gerçekleştirilen işletim sistemleri üzerindeki araştırmalar, genellikle iş dağıtımı konularına yöneliktir.

Carnegie Mellon Üniversitesi'nde gerçekleştirilen RTMach projesi (Mercer et al., 1994) kapsamında Mach işletim sistemine gerçek zamanlı iş parçacığı, gerçek zamanlı senkronizasyon ve iş dağıtımı konularına yönelik eklemeler yapılmıştır. RTMach projesi, çokluortam uygulamalarından çok geleneksel gerçek zamanlı uygulamalar dikkate alınarak yapılmıştır. İş dağıtımı algoritması, katı gerçek zamanlı uygulamalar için geliştirilmiş klasik teknikleri kullanır (Liu and Layland, 1973).

Govindan (1992), çekirdek seviyesi ve kullanıcı seviyesi iş parçacıklarının ortaklaşa çalıştıkları hiyerarşik iş dağıtımı modelleri üzerinde çalışmıştır.

Çokluortam uygulamaları için Lancaster Üniversitesi'nde dağıtık gerçek zamanlı ve çokluortam uygulamalarını destekleyecek mikro çekirdek tabanlı Chorus işletim sistemi üzerinde bir sistem geliştirilmiştir. Bu sistem; CPU iş dağıtıcısı, sanal bellek sistemi ve işletim sistemini entegre ederek sistem çağrılarını, içerik değiştirmelerinin ve kopyalama işlemlerinin sayısını azaltmayı hedefler. İş dağıtıcısı Govindan'ın geliştirdiği hiyerarşik iş dağıtıcısını kullanır. Çekirdeklerin iş dağıtımı EDF (Earliest Deadline First) ile gerçekleştirilir. Dağıtık bir ortamda iş istasyonları arasındaki kopyalama işlemini azaltmak için çekirdekler arasında doğrudan bağlantılar kurulur.

Veriler, kullanıcı sahasına aktarılmadan doğrudan çekirdekler arasındaki kanallarla taşınırlar. Lancaster Üniversitesi'nde dağıtık sistemler grubu işletim sistemi desteğinin yanı sıra iletişim desteği üzerinde de çalışmalar yapmaktadır. Bu alanda üzerinde çalışılan konular arasında çokluortam ağ arayüzleri (Blair et al., 1993), çokluortam akışlarının senkronizasyonu (Shepherd et al., 1991) ve yüksek hızlı ağlar için geliştirilen QoS mimarileri (Campbell 1992a, Campbell 1992b) göze çarpmaktadır.

Kantarıcı ve Tunalı (1998a, 1998b) tarafından düşük sayıda öncelik seviyesi destekleyen sistemler için bir iş dağıtım mekanizması geliştirilmiştir. Bu mekanizmada işletim sisteminin sahip olduğu öncelik tablosu sınıflara bölünür Her sınıfa birden fazla doğal öncelik seviyesi atanır. Sisteme yeni bir iş katıldığında işin periyoduna bakılarak hangi öncelik sınıfına düştüğü belirlenir. Bir öncelik sınıfında bulunan işlerin önceliği de o öncelik sınıfında yer alan işlerin ortalama işletim süreleri dikkate alınarak belirlenir. Simülasyonlarla bu yaklaşımın kesinti başarımı ölçülmüş ve önerilen mekanizmanın Rate Monotonic iş dağıtım algoritmasından daha az sayıda kesintiye neden olduğu saptanmıştır.

3 GELİŞTİRİLEN SİSTEM

İstek üzerine video konusunda yapılan çalışmalar iki ana kolda ilerlemektedir. Bunlardan birincisi video verilerinin depolanması ve videolara erişim konularına yöneliktir. İkincisi, video bilgilerinin ağ üzerinden iletimi ve alıcı istasyonlarında gösterimi ile ilgilenir. Bu tez çalışması, video verilerinin iletilmesi ve alıcıda sunulması konularına odaklanmıştır. Sistemin en önemli özelliklerinden biri sadece yazılım tabanlı bir sistem oluşudur. Sistem yazılımı servisçi, istemci ve iletişim modülü olmak üzere üç parçadan oluşmaktadır.

Servisçi yazılımının temel görevleri iletişim modülüyle ilişki kurarak video paketlerini istemciye göndermek ve ağın mevcut durumunu dikkate alarak iletişimde yeni düzenlemeler yapmaktır. İstemci yazılımı, ağdan gelen paketlerin alınması, birleştirilmesi ve kullanıcıya sunulmasından sorumludur.

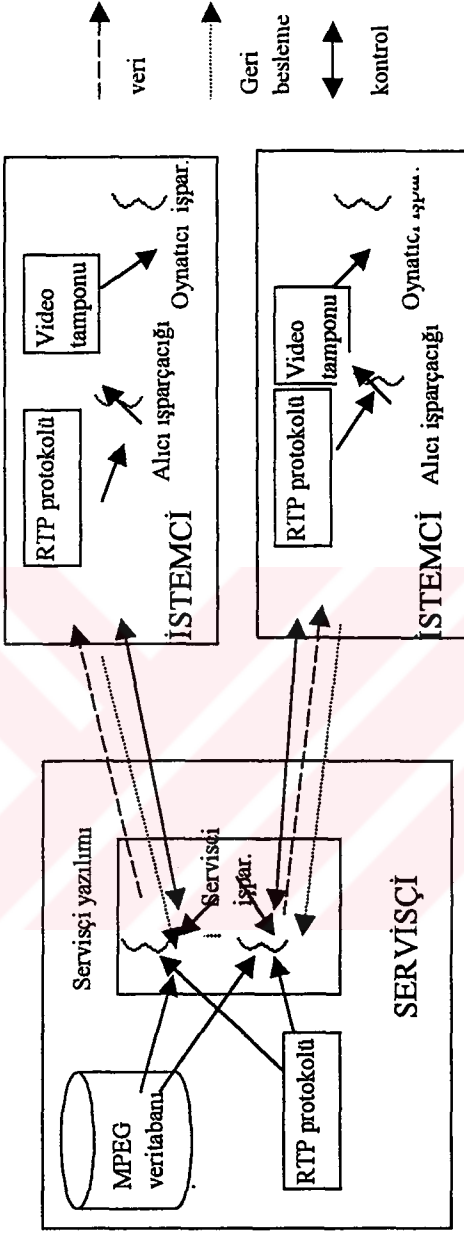
İletişim modülü, Lucent Technologies tarafından geliştirilen RTP (Real-time Transport Protocol) kütüphanesi kullanılarak oluşturulmuştur. MPEG-1 videolarının RTP paketlerine dönüştürülme işlemi tarafımızdan gerçekleştirilmiştir. Servisçiden her bir istemciye doğru tek yönlü bir RTP kanalı video verilerinin taşınmasından sorumludur. Her bir istemciden servisçiye uzanan bir RTCP kanalı, ağın durumu hakkında servisçiye düzenli olarak bilgi verir. Kontrol bilgileri ise çift yönlü bir TCP kanalı ile taşınmaktadır.

Sistemi benzerlerinden ayıran önemli özelliklerden biri POSIX 4 gerçek zamanlı programlama kütüphanesi ve POSIX iş parçacığı kütüphanesinin kullanımıyla oluşturulmuş çok iş parçacıklı ve esnek

gerçek zamanlı bir uygulama olmasıdır. Mevcut uygulamalar (Cen et al., 1995; Rowe et al., 1994; Chen et al., 1995) geleneksel işletim sistemlerinin sahip olduğu programlama olanaklarını kullanan tek iş parçacıklı uygulamalardır. Bu uygulamaların temel sorunu, program geliştirmenin zorluğu, programlama yükünün fazlalığı ve performans düşüklüğüdür. Çok iş parçacıklı mimari ile program geliştirme kolaylaşmış, program kodunun büyüklüğü azaltılmış ve uygulamanın başarımı arttırılmıştır. Geleneksel işletim sistemi olanaklarının yanı sıra POSIX 4 gerçek zamanlı uygulama geliştirme kütüphanesinin de kullanımıyla video verilerinin iletimi ve sunulması sırasında gereksinim duyulan zaman kısıtlarına uyulması sağlanmıştır. Alt bölümlerde geliştirilen sistemle ilgili detaylı bilgi verilmektedir.

3.1 Sistem Mimarisi

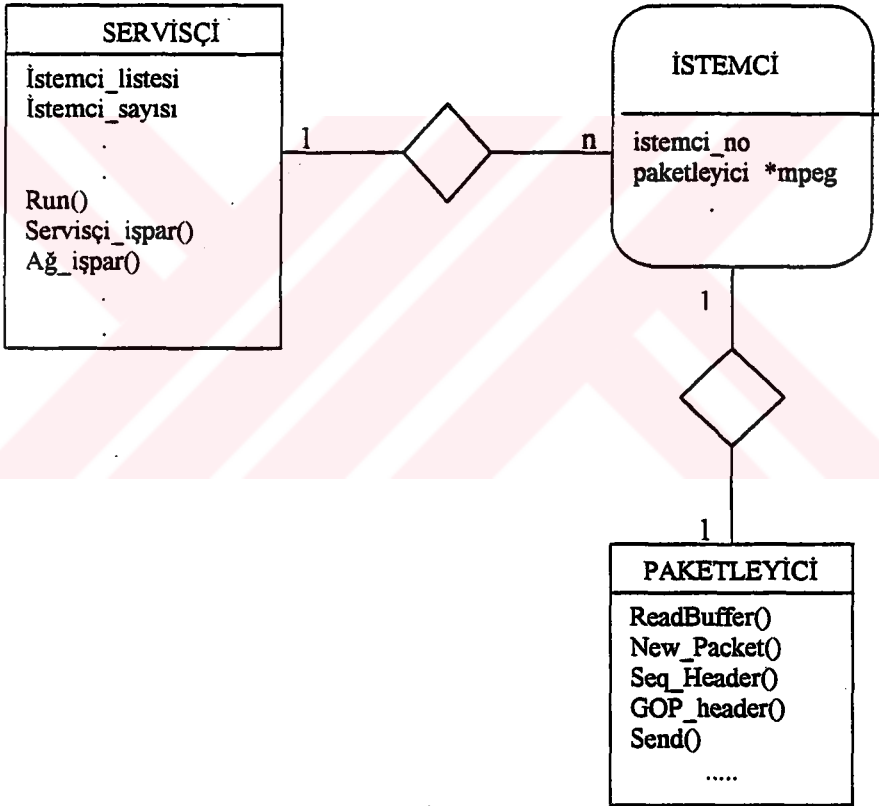
Geliştirilen sistemin mimarisi Şekil 3.1'de verilmektedir. Sistem İnternet üzerinde bir servisçi ve bir çok alıcıdan oluşan istemci-sunucu (client-server) mimariye sahiptir. Servisçi istemciden gelen video isteklerini kabul etmekte ve düzenli aralıklarla diskten okuduğu verileri UDP tabanlı RTP protokolü ile istemciye iletmektedir. İstemci gelen verileri oynatılış anına kadar tampon sahada saklar. İstemci ve sunucu arasında kontrol mesajlarının gönderimi için iki yönlü bir TCP kanalı kullanılmaktadır. Bu konfigürasyon video verilerinin taşınması için geçen süreyi düşürmekte ve kontrol bilgilerinin güvenli bir şekilde iletimine olanak sağlamaktadır. Ayrıca, istemci ağın durumunu gözetleyerek RTCP protokolü ile servisçiye bilgi vermekte ve gerektiğinde servisçiden iletişim ile ilgili düzenlemeler yapmasını istemektedir. Tampon sahada alıcı için sürekli veri bulunmasını garantilemek için bir akış kontrol mekanizması kullanılmaktadır. Alt bölümlerde servisçi ve istemci mimarileri detaylı olarak anlatılmaktadır.



Şekil 3.1. Sistem mimarisi

3.2 Servisçi

Servisçi yazılımı çok iş parçacıklı ve nesneye dayalı programlama mantığı ile oluşturulmuştur. Servisçi tarafında, her bir istemci bir iş parçacığı ile temsil edilir. Her bir iş parçacığı, ilgili istemciden gelen isteklerin kabulü, videonun akışlar (Johnson, 1996) halinde istemciye gönderilmesi ve istemciyle ilgili servis kalitesinin kontrolü işlemlerini yerine getirir. Servisçi yazılımına ait nesne diyagramı Şekil 3.2 de verilmiştir.



Şekil 3.2. Servisçiye ait nesne tabanlı mimari

Şekil 3.2'de görüldüğü gibi servisçinin kendisi bir nesnedir. Her yeni bağlantı isteği yeni bir istemciye karşılık gelir. Servisçi yazılımı her yeni istemci için bir iş parçacığı ve İSTEMCİ isimli bir yapı yaratır. Her İSTEMCİ yapısı, bu yapının ait olduğu istemciye paketleri gönderilmesini sağlayan PAKETLEYİCİ isimli bir nesneye sahiptir. RT kütüphanesi ile ilişki bu nesne sayesinde kurulmaktadır. Servisçinin nesne tabanlı olarak gerçekleştirimi servisçi yükünü azaltmış ve istemcilerin izinin tutulmasını kolaylaştırmıştır.

POSIX 4 gerçek zamanlı programlama uzantıları ve çok iş parçacıklı programlama kütüphanesi kullanılarak geliştirilen servisçi yazılımı şu fonksiyonları yerine getirmektedir:

1. Yeni bağlantıların kabulü
2. Video verilerinin akışlara dönüştürülmesi
3. Video paketlerinin iş dağıtımının gerçekleştirilmesi
4. Bant genişliği yönetimi
5. Uyum

Alt bölümlerde bu fonksiyonlar detaylı olarak açıklanmaktadır.

3.2.1 Alıcılardan gelen bağlantı isteklerinin kabulü

Alıcılar, TCP kanalı vasıtasıyla video isteklerini servisçiye bildirirler. İstenen video, servisçi makinasında mevcutsa, servisçi ve alıcı arasında RTP/RTCP kanalları kurulur ve servisçi bu videoyu isteyen alıcı için bir iş parçacığı yaratır. Bundan sonra servisçi çerçeve hızı,

çözünürlük gibi videoya ilişkin temel bilgileri TCP kanalıyla istemciye gönderir. Bu işlemden sonra servisçi veri göndermeye, istemci de veri almaya hazır hale gelmiştir.

3.2.2 Video verilerinin akışlara dönüştürülmesi

Mevcut uygulamalarda (Cen et al., 1995; Rowe et al., 1994; Chen et al. , 1995) kullanılan akışlandırma yaklaşımlarından biri her bir çerçevenin tek bir pakette istemciye gönderilmesidir. Bu yaklaşım, göndericinin bant genişliği gereksinimini artırır. Aynı anda bir çok istemciye servis vermek gerektiği durumlarda bu yaklaşım, servisçide büyük bir yük yaratır.

Bazı sistemler, çerçeveleri sabit büyüklükte paketlere bölerek bu yükü azaltırlar. MPEG video yapısı hiyerarşik bir özellik göstermektedir. MPEG çerçevelerinin içeriği dikkate alınmadan paketlere bölünmesi bu hiyerarşik yapının getirdiği avantajlardan vazgeçilmesi anlamına gelir.

RTP protokolü, bu iki yaklaşım yerine verilerin yapısını dikkate alarak, farklı veri formatlarını İnternet üzerinden iletme uygun şekilde paketleyebilmektedir. IETF, RTP protokolünün desteklediği her veri türü için ayrı paketleme kuralları belirlemiş ve bu kuralları çeşitli İnternet taslakları halinde yayınlamıştır (Schulzrinne et al., 1998).

MPEG-1 çerçeveleri çok yer tuttuğu için kullanılmakta olan LAN/WAN MTU (maksimum nakil ünitesi-maximum transportable unit) büyüklüğünden daha küçük büyüklükteki paketlere bölünmelidir. Ethernet için bu değer yaklaşık 1400 'byte'tir. İlgili taslakta verilen paketleme kuralları şöyledir (Hoffman et al., 1998):

1. MPEG Video_Sequence_Header, her zaman bir RTP paketinin başında yer almalıdır.
2. MPEG GOP_header bir Video_Sequence_Header'ini izlemeli ya da bir RTP paketinin başında yer almalıdır.
3. MPEG Picture_header, bir RTP paketinin başında yer almalı ya da bir GOP başlığını izlemelidir.

Bu kurallar paketler arasında bağımlılığı en aza indirgeyecek ve alıcıda açılıp oynatılabilen veri miktarını en yüksek düzeye çıkaracak şekilde tasarlanmıştır. MPEG verilerini RTP paketlerine dönüştürülme kuralları paketler arasındaki bağımlılığı azaltır ve alıcıdaki kullanılabilir veri miktarını en yüksek düzeye çıkarır. Bunu şöyle açıklayabiliriz: MPEG çerçeveleri kesitlerden meydana gelmiştir. Bir MPEG açıcısı normal olarak bir kesitte veri kaybı veya bozulması olduğunda diğer bir keside geçecektir. Bunu kolaylaştırmak için kesit başlangıcı ve bitişini gösteren bitlerin RTP başlıklarına yerleştirilmesi gerekmektedir. Bir kesitin başlangıcı bir paketin başına ya da bir pakete yerleştirilen tam sayıda kesitin arkasına yerleştirilmelidir. Bu sayede bir kesite ait paketlerden biri kaybolduğunda paketin alıcı tarafından taranmasına gerek kalmadan bu kesiti izleyen kesit bulunabilir.

Bir RTP başlığı şu sahalardan oluşur (Schulzrinne, 1996):

Yük türü: RTP, farklı veri türlerini desteklediği için akışların hangi veri türüne ait olduğu RTP başlığında verilmelidir. Alıcı, RTP başlığındaki bu sahayı kontrol ederek kullanılan veri türüne uygun biçimde paketleri açar ve çerçeveleri oluşturur.

M biti: Bir video çerçevesine ait son paketteki başlıkta bulunan M biti 1 olacak şekilde ayarlanır.

Zaman etiketi: MPEG resimlerinin sunum zamanını gösteren 90 KHz'lik saatten elde edilen 32 bitlik zaman etiketi bir RTP başlığının en son elemanıdır.

RTP paketlerinde, kullanılan veri türüne özgü bir başlık da bulunur. MPEG verilerine ait başlık şu bilgileri içerir:

T biti: MPEG2 videolarının kullanıldığı durumda, MPEG başlığını MPEG2 videoları için kullanılan bir başlık izler. Bu durumda T biti 1 yapılır.

TR biti: Bir çerçeveyi o GOPta bulunan diğer çerçevelerden ayırmak için kullanılan 10 bitlik bir sahadır. Bir çerçeveye ait tüm RTP paketleri için bu saha aynıdır.

N biti: Paket yükü bir çerçevenin başlangıcı ise 1 değerini alan tek bitlik bir sahadır.

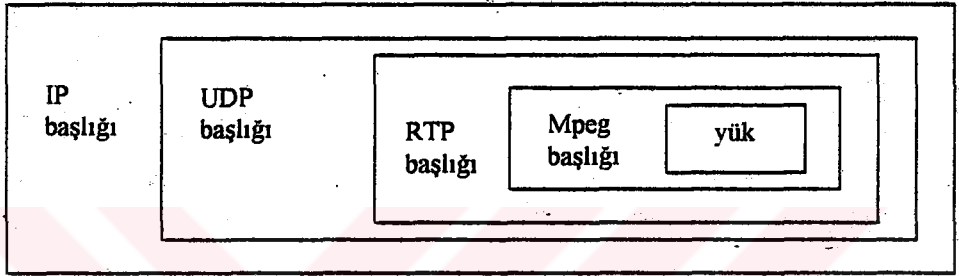
S biti: Paket yükü bir video dizisi başlığı ile başlıyorsa bir değerini alan tek bitlik bir sahadır.

BS - Beginning of slice: Yük sahasının ilk bilgisi slice_start_code ise ya da bir slice_start_code bir Video_Sequence_Header, GOP_header, Picture_header bilgilerini izliyorsa bu bit 1 yapılır.

ES - End of slice: Bir MPEG yükünün son byte'ı bir MPEG kesidinin sonu ise bu bit 1 yapılır.

P - Picture type: O paketin ait olduđu resim türünü gösteren bir sahadır (1- I, 2- B , 3-P).

Uygulama katmanında verilerin önüne sırasıyla MPEG ve RTP, daha alt katmanlarda UDP ve IP başlıkları eklenir. Ağ arayüzünden çıkan bir paketin yapısı Şekil 3.3'te verilmiştir.



Şekil 3.3. Ağ arayüzünden çıkan bir paketin yapısı

3.2.3 RTP paketlerinin iş dağıtımının gerçekleştirilmesi

Servisçi yazılımının çok iş parçacıklı olması paketlerin iş dağıtımını kolaylaştıran ve iş dağıtım yükünü en aza indirgeyen önemli bir unsurdur. Eğer tek iş parçacıklı bir mimari kullanılmakta olsa idi, iş parçacıklarının iş dağıtımı için bir zamanlayıcı kuyruğu ('timer queue') kullanılması gerekirdi. Bu durumda, bir iş dağıtıcısı modülü periyodik kesintilerle iş kuyruğundan servis zamanı gelmiş bir alıcıyı belirleyerek o alıcıya ait bir paketi ağa gönderecekti. Saat kesintileri periyodik aralıklarla zamanlayıcı kuyruğundan bir iş seçtiği için paketlerin ağa gönderiliş zamanının teorik ağa bırakılış zamanından farklı bir zaman

olması olasılığı yüksektir. Çok iş parçacıklı mimari, zamanlayıcı kuyruğu kullanma gereksinimini ortadan kaldırarak paketlerin teorik olarak ağa bırakılmaları gereken zamandan daha yakın bir zamanda ağa gönderilmesini sağlamaktadır. Bu, her iş parçacığının kendi kendini zamanlayabilmesi ile mümkün olabilmektedir. POSIX 4 kütüphanesine ait *pthread_cond_timed_wait* çağrısı kullanılarak ağa bir paket gönderildikten sonra o iş parçacığının bir sonraki paket gönderme zamanına kadar bloklanması sağlanır. Geleneksel *sleep* ve *nanosleep* çağrıları, tüm süreci blokladığı için diğer iş parçacıklarının çalışması etkilenmez. Bir iş parçacığı bloklanırken diğer bir iş parçacığı CPU'yu kullanır. *pthread_cond_timed_wait* çağrısının önemli bir özelliği, bu çağrının son kullanıldığı zamana göre değil, saatin başlangıcına göre bloklama yapmasıdır. Şekil 3.4'de paketlerin iş dağıtım algoritması verilmiştir.

Solaris işletim sisteminde gerçek zamanlı saat çözünürlüğünün 10 ms. oluşu paketlerin ağa gönderilmesi konusunda önemli bir sorun yaşatmıştır. Teorik hesaplamalara göre paketlerin ağa gönderiliş zamanları arasındaki zaman farkı 1-2 milisaniye ya da birkaç yüz mikrosaniye olabilmektedir. Saat çözünürlüğünün 10 milisaniye oluşu, paketlerin ağa en az 10 milisaniye aralıklarla bırakılmasına neden olarak alıcıda gecikmelere yolaçmıştır. /etc/system dosyasına *set hires_tick=1* komutu eklenerek saat çözünürlüğünün düşürülmesi sağlanmış ve ağdaki gecikmeler önlenmiştir.

```

/* starttime: Paket işlenmeye başlanmadan önceki zaman
Next_packet_time: Sonraki paketin işleme konacağı an
Packet_interval: Ardışık paketler arası zaman*/
while(playback_has_not_ended)
{
    gettimeofday(starttime);
    process_a_packet();
    next_packet_time=starttime+packet_interval;
    pthread_cond_timed_wait(next_packet_time);
    ...
}

```

Şekil 3.4. Paketlerin gönderiş zamanlarının belirlenmesi

3.2.4 Bant genişliği yönetimi

Servisçide bulunan veri tabanına yeni bir video katıldığı zaman bu videoya ait istatistikler bir yazılım tarafından çıkarılır ve bir dosyada saklanır. Servisçi gönderim sırasında bu istatistikleri kullanarak ardışık paketlerin gönderim aralığını belirler. En ideali, her periyotta bir çerçeve gönderilmesidir. Ancak MPEG çerçeveleri fazla yer tuttuğu için bu yaklaşım ağdaki trafiği artırır. O yüzden bir çerçeveyi paketlere bölerek ağa göndermek gerekir. Çerçeve büyüklüğünü S , paket büyüklüğünü s ile gösterecek olursak bir periyotta gönderilmesi gereken paket sayısı $n=S/s$ olarak hesaplanabilir. Böylece, r çerçeve hızı olmak üzere bir çerçeveye ait paketlerin $(1/r)/n$ saniyelik aralıklarla gönderilmesi gerekir. Bu

yaklaşım, çerçeve büyüklüklerinin birbirine yakın olduğu sıkıştırma yöntemleri için uygun sonuçlar verir. Fakat, MPEG gibi çerçeve büyüklükleri değişen sıkıştırma algoritmaları için bu yöntemin bazı sakıncaları vardır. Çerçeve büyüklüklerinin değişken olması ağda VBR (variable bit rate) türü bir trafik yaratır. Örneğin, çerçeve hızının 10 çerçeve/saniye, bir I çerçevesinin 100.000 bit ve bir B çerçevesinin 10.000 bit büyüklüğünde olduğu durumda yukarıdaki yaklaşımda her bir çerçeve 0.1 saniyede gönderileceğinden bir I çerçevesini göndermek için gereken ağ kapasitesi 1.0 Mbps olurken, bir B çerçevesini göndermek için gereken kapasite sadece 0.1 Mbps olur.

Ayrıca, büyük çerçeveler, küçük çerçevelerden daha fazla sayıda pakete bölünürler. Bu nedenle, büyük çerçeveler için paketler arası zaman aralığı daha küçüktür. Bu durum, ağda sıklık anında büyük çerçevelere ait paketlerin kaybolma olasılığını artırır. MPEG videolarında I çerçeveleri en büyük çerçevelerdir. Aynı zamanda, bir GOP'a ait tüm çerçevelerin açılabilmesi I çerçevelerine ait paketlerin tam olarak alınabilmesine bağlıdır. Bir I çerçevesine ait paketlerin birindeki kayıp, bu çerçeveye ait diğer çerçevelerin bozulmasına yol açar. I çerçeveleri en büyük çerçeveler olduğu için paketler arası gönderim aralığı diğer çerçeve türlerine göre küçük, paketlerin kaybolma olasılığı da diğer çerçevelere oranla daha büyüktür.

Bu nedenle, çerçeve tabanlı paket gönderim aralığı belirleme yaklaşımı yerine INTERVAL_FOR_EACH_GOP isimli GOP tabanlı bir yaklaşım geliştirilmiştir. Bu yaklaşımda paket aralığı bir çerçeveye ait paketler değil, bir GOP'ta bulunan çerçevelerin büyüklüklerine ait istatistikler dikkate alınarak belirlenmiştir.

S çerçeve büyüklüğü, s paket büyüklüğü, T bir GOP'un gönderileceği süre, m bir GOP'ta bulunan çerçeve sayısı olsun. Buradan i. çerçevenin bölüneceği paket sayısı yaklaşık olarak

$$n_i = \frac{S}{s} \quad (1)$$

eşitliği ile bulunabilir. Bir GOP'un bölüneceği yaklaşık paket sayısı N

$$N = \sum_{i=1}^m n_i \quad (2)$$

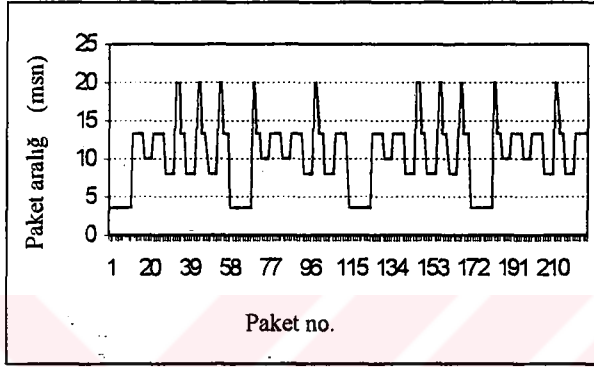
olarak hesaplanır. Paketler arasındaki gönderim aralığının yaklaşık değeri

$$t = \frac{T}{N} \quad (3)$$

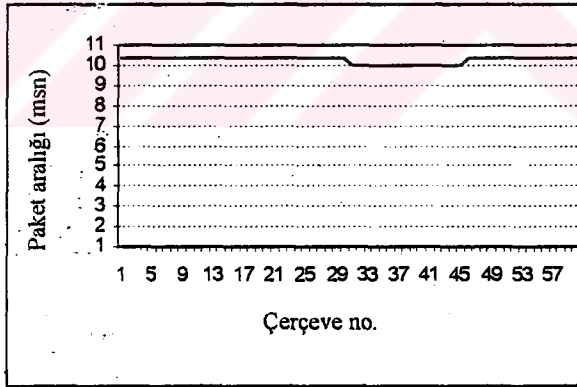
olarak belirlenir. T 30 çerçeve/sn lik çerçeve hızı için 0.5 sn.'ye karşılık gelmektedir.

Çerçeve büyüklüğüne bağlı bant genişliği yönetim mekanizmaları, MPEG çerçeveleri için atanmış bant genişliği miktarının değişkenliğini arttırarak sistemin tahminlenebilirliğini düşürür. GOP büyüklüğüne bağlı yaklaşım, bir GOP'a ait paketler arası iletim zamanını sabit hale getirerek atanmış bant genişliği miktarının değişkenliğini düşürür. Bir videoya ait GOP'ların çoğunlukla birbirine yakın büyüklükte olduğu düşünülürse, bir videoya ait paketler arası gönderim zamanının küçük bir aralıkta

değişkenlik gösterir. Şekil 3.5'te, çerçeve ve GOP tabanlı yaklaşımlara göre bir dakikalık bir MPEG-1 videosu için geçerli paket aralıkları görülmektedir. Bu şekilden de anlaşılabacağı gibi GOP tabanlı yaklaşım, paket aralıkları bakımından daha az değişkenlik göstermekte, uygulamanın tahminlenebilirliğini ve kararlılığını arttırmaktadır.



(a)



(b)

Şekil 3.5. Paketlere ayırmada (a) çerçeve tabanlı yaklaşım ve (b) GOP tabanlı yaklaşımların karşılaştırılması

Yukarıda anlatılan yaklaşımda, gönderilecek paket sayısının tahmin edilen değerleri kullanılmaktadır. Düşük sıkıştırma hızlarında tahmini ve gerçek değerlerin birbirine yakın olduğu gözlenmiştir. RTP paketlerinin büyüklüğü sabit olmadığı için ve formülde kullanılan s değeri paket büyüklüğünün üst sınırı olduğu için gerçek paket sayısının tahmini yüksektir. Sıkıştırma bit hızı yükseldikçe gerçek paket sayısı ve tahmini paket sayısı arasındaki farkın arttığı gözlenmektedir. Bu nedenle, gerçek paket sayısının kullanıldığı `INTERVAL_FOR_WHOLE_VIDEO` isimli başka bir yaklaşım geliştirilmiştir. Bu yaklaşımda, sisteme yeni bir video katıldığında paketleme algoritması kullanılarak videonun bölüneceği paket sayısının gerçek değeri bulunur. Her uyum seviyesi için bulunan paket sayısı değerleri video istatistiklerinin saklandığı dosyaya yerleştirilir. Videonun oynatış süresi gerçek paket sayısına bölünerek ardışık paketler arasındaki zaman aralığı hesaplanır.

Bu yaklaşım, sıkıştırma hızı yükseldikçe ilk yaklaşımdan daha iyi sonuçlar vermektedir. Ayrıca, ilk yaklaşımda, paketler arası gönderim aralığı GOP büyüklüğüne bağlı olduğu için sürekli değişkenlik göstermektedir. İkinci yaklaşım, iletişimin başlangıcında sabit paket aralıkları kullanmaktadır. Daha sonraki bölümlerde tanıtılacak tampon saha yönetimi işlemi sırasında gerektiğinde gönderim aralığı değiştirilebilmektedir.

3.2.5 İletişimle ilgili istatistiklerin toplanması

İletişim sırasında, servisçiye belli aralıklarla ağın durumu hakkında bilgi verilmelidir. Servisçi, istemcilerden gelen bu bilgilere bakarak iletişimle ilgili yeni düzenlemeler yapabilir.

Bu amaçla, RTP protokolünün tamamlayıcı protokolü olan RTCP protokolü kullanılır. RTCP paketleriyle toplanan istatistikler ağdaki sıkışıklığın belirlenmesinde önemli rol oynarlar (Schulzrinne, 1996).

RTCP gönderici raporları, alıcı raporları, kaynak tanımlama raporları, hoşçakal raporları ve uygulamaya özgü raporlar olmak üzere beş farklı rapor türüne sahiptir. Sistemimizde, veri akışı servisçiden alıcıya doğru olduğu için sadece alıcıdan servisçiye gönderilen alıcı raporlarının kullanılması söz konusudur. Bir alıcı raporunda şu sahalara bulunur (Schulzrinne et al., 1996) :

1. SSRC: Gönderici istasyonun numarasını belirtir.
2. Kayıp paketlerin yüzdesi
3. Toplam kayıp paket sayısı
4. Alınan son RTP paketinin sıra numarası
5. Gelişlerarası gecikmelerin varyansı (jitter)
6. Son gönderici raporunun alındığı zaman
7. Son gönderici raporunun alınması ve bu alıcı raporunun gönderilmesi arasında geçen süre

RTCP paketlerinde bulunan gelişlerarası gecikme zamanının varyansı, ağdaki kısa dönemli sıkışıklıkların göstergesidir. Varyansın artması gelecekte yaşanacak paket kayıplarının habercisi olabilir. Paket

kayıp istatistikleri ise ağdaki kalıcı sıkışıklığın, dolayısıyla iletim kalitesinin bir göstergesidir.

RTCP raporları servisçiye önemli bir geri besleme sağlamalarına rağmen, kontrolsüz olarak gönderilirlerse büyük bir bant genişliği ihtiyacı yaratarak RTP paketlerinin zamanında alıcıya ulaşmasına engel olabilirler. Bu yüzden, ardışık RTCP paketlerinin gönderim aralığı belli kurallar kullanılarak belirlenir. Bu kurallar, ağdaki RTCP trafiğinin sabit düzeyde kalmasını sağlayacak şekilde belirlenmiştir. Sisteme katılan alıcı sayısı arttıkça, RTCP raporlarının gönderim frekansı düşer. Kuralların belirlenmesinde ikinci bir ilke, her istasyonun rapor yaratma aralıklarının rasgele olarak belirlenmesidir. Tüm istasyonların aynı aralıkla rapor göndermesi, hepsinin aynı anda rapor göndermesine yol açacağından ağ yükünü arttırır. Bu durumdan kaçınmak için alıcılar, orijinal gönderim aralığının yarısı ve 1.5 katı arasında rasgele bir değeri gönderim aralığı olarak kullanır (Schulzrinne et al., 1996).

3.2.6 Servis kalitesi kontrolü

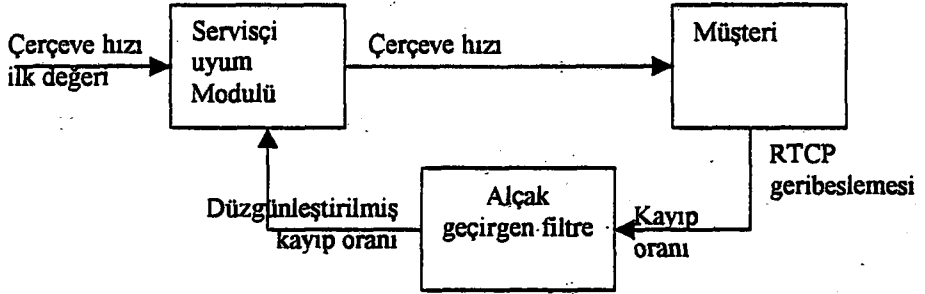
İnternet, bir en iyi çaba ortam olduğu için ağ üzerinde sürekli değişken bir yük gözlenir. Ağ yükünün aniden arttığı durumlarda iletilen paketler kaybolmaya başlar. Kayıp paketler, alıcıda önemli sorunlara yol açabilir. Bu nedenle, servisçinin değişken ortama uyum sağlamak için gerçekleştirmesi gereken işlemler mevcuttur. Servisçi, RTCP raporlarıyla aldığı paket kaybı istatistiklerine bakarak gönderilecek GOP dizisinden bazı çerçevelerin atılmasına karar verir. Çerçeve düşürme gelişi güzel değil, akıllı bir şekilde yerine getirilir. Çerçeve türleri arasındaki bağımlılık ilişkilerinin dikkate alınması ve atılacak çerçevelerin bir GOP içinden homojen olarak seçilmesi gerekir.

RTCP raporlarındaki kayıp oranı istatistikleri ağdaki tıkanmanın bir göstergesi olarak kullanılmıştır. Tıkanmalar geçici ya da kalıcı olabilirler. Geçici tıkanmaların olduğu durumlarda kayıp oranı kısa bir süre için artar, sonra eski haline gelir. Tıkanmanın uzun sürmesi durumunda ise kayıp oranı sürekli olarak artar. Tıkanıklılığın kalıcı olup olmadığı alçak geçiren (low-pass) filtre kullanılarak belirlenebilir.

III- Sistemde

$$\lambda \leftarrow (1-\alpha)\lambda + \alpha b \quad (4)$$

filtresi kullanılmıştır (Busse et al., 1995). Burada b RTCP raporlarıyla toplanan yeni kayıp oranını, λ düzgünleştirilmiş kayıp oranını göstermektedir. α , 0 ve 1 arasında değişken bir parametre olup, yeni ölçülen kayıp oranının düzgünleştirilmiş kayıp oranı üzerindeki etkisini belirlemektedir. α arttıkça yeni kayıp oranının ağırlığı artar, düştükçe eski kayıp oranlarının ağırlığı da artar. Filtrenin çıktısı iki eşik değeri ile karşılaştırılır. Büyük eşik değerinden daha büyük bir kayıp oranı ağdaki tıkanıklığın kalıcılığının göstergesidir. Bu durumda bazı çerçeveler atlanarak ağın yükü düşürülür. Kayıp oranının küçük eşik değerinden küçük olması ağın yükünün hafif olduğu anlamına gelir. Bu durumda GOP dizisinden çıkarılarak ağa gönderilmeyen çerçeveler varsa bu çerçeveler tekrar GOP dizinine eklenir. Kayıp oranı iki eşik arasında ise herhangi bir değişikliğe gidilmez. Bu iki eşik değeri arasında servis kalitesi dalgalanmasına izin vermeyecek şekilde aralık bırakılmalıdır. Servis kalitesi kontrolü Şekil 3.6'da gösterilmiştir.



Şekil 3.6. Servis kalitesi kontrolü

Çerçeve türleri arasındaki bağımlılık ilişkisine göre ilk önce B, sonra P, son olarak da I çerçeveleri düşürülmelidir. I çerçeveleri düşürülmeye başladığında video kalitesi önemli ölçüde bozulur. Ağ yükünün tekrar normale döndüğü tespit edilince çerçeveler ters sırada GOP dizisine eklenirler.

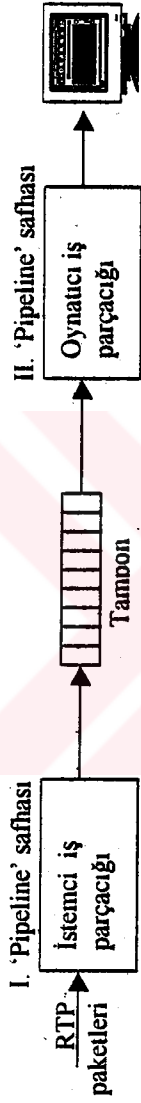
Çerçeve düşürme işlemi seviyeler halinde gerçekleştirilir. Çizelge 3.1'de IBBPBBPBBPBBPBB GOP dizisi için hız uyum seviyeleri verilmiştir. Ağda tıkanıklık yoksa akışlandırma işlemi 1. seviyede ilerler. Ağda sıkışıklığın mevcut olduğu durumda uyum seviyesi 1 artırılır ve çerçeve düşürme işlemi başlar. Ağdaki tıkanıklık ortadan kalktıkça hız uyum seviyesi 1 azaltılır ve atılan çerçeveler tekrar GOP dizisine eklenir. Önemli bir nokta, çerçevelerin farklı büyüklükte olmasından dolayı her uyum seviyesinde düşürülen veri miktarının eşit olmayışdır.

Çizelge 3.1. IBBPBBPBBPBBPBB GOP dizisi için hız uyum seviyeleri

Uyum Seviyesi	GOP dizisi	Düşürülen çerçeve türleri
1	I B B P B B P B B P B B P B B	--
2	I B x P B x P B x P B x P B x	5 B
3	I x x P x x P x x P x x P x x	10 B
4	I x x P x x P x x x x x x x x	10 B + 2 P
5	I x x x x x x x x x x x x x x	10 B + 4 P

3.3 İstemci

İstemci tarafının sorumluluğu video çerçevelerini ağdan almak, sıkıştırılmış video verilerini açmak, görüntülemek ve ağın durumunu izlemektir. Servisçi, istemci tarafından sağlanan bilgiye karşılık verir. Bu şekilde, servisçinin yükü azaltılmış ve servis verilebilecek müşterilerin sayısı artırılmış olur. Şekil 3.7'de istemci yazılımının mimarisi görülmektedir. Alt bölümlerde istemci detaylı olarak anlatılmaktadır:



Şekil 3.7. İstemci yazılımı mimaris

3.3.1 İstemci Mimarisi

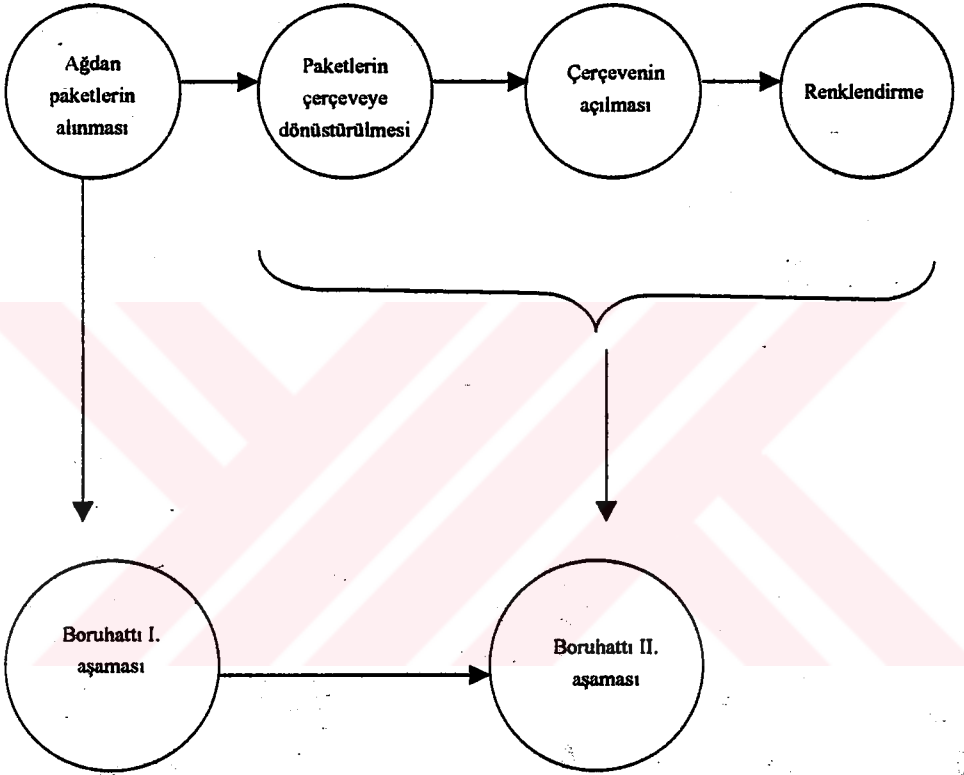
İstemci tasarımında ana amaç, uygulama içinde birlikte işletimi sağlayarak sistemin verimini ve performansını arttırmaktır. Şekil 3.8'de birlikte işleme izin vermeyen bir istemci mimarisi gösterilmektedir.

```
while (1)
{
    p=get_packet_from_network();
    if (p == last_packet_of_a_frame)
    {
        Combine_the_packets_into_a_frame()
        Display_frame()
    }
}
```

Şekil 3.8. Birlikte işleme izin vermeyen bir istemci mimarisi

Bu yapı, ağ üzerinde tam bir çerçeve gelinceye kadar paketleri alır; bir çerçeveye ait son paket geldiğinde o çerçeveye ait tüm paketleri birleştirir ve çerçeveyi açarak ekranda gösterir. Sıkıştırılmış verilerin açılması ve oynatılması zaman alıcı bir işlem olduğu için, CPU bu işle meşgulken, ağdan gelen yeni paketlerin çoğunun alıcıdaki protokol tamponlarında kaybolmasına neden olmuştur.

Bunun üzerine, istemci, kullanılan kaynakları dikkate alarak, birlikte işletim esasına dayalı bir boruhattı ('pipeline') mimari üzerinde gerçekleştirilmiştir. İstemcinin boruhattı mimariye dönüştürülmesi adımları Şekil 3.9'da verilmiştir.



Şekil 3.9. Alıcı istasyona ait akış diyagramının 'pipeline' modeline eşlenmesi

Şekil 3.9'da verilen akış diyagramı incelendiğinde- ilk adımın ağırlıklı olarak ağ, diğer üç adımın ağırlıklı olarak CPU'yu kullandığı görülmektedir. Kullanılan kaynaklar gözönüne alınarak bu akış diyagramı iki sayfaya bölünmüştür. İlk safha, ağ kaynağını kullanan

birinci adımdan oluşmakta olup, boruhattının ilk aşamasına karşılık gelmektedir. İkinci safha, CPU ağırlıklı olan diğer üç adımı içeren boruhattının ikinci aşamasını oluşturmaktadır.

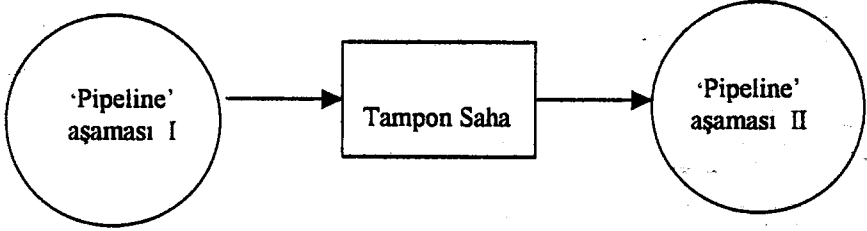
Boruhattının her bir safhası farklı bir iş parçacığı ile gerçekleştirilmiştir. İş parçacıklarından biri, ağdan gelen paketleri alır, diğeri paketleri çerçevelere birleştirir, çerçeveleri açar ve ekranda oynatır. Bu şekilde, CPU ve I/O yoğun işlemlerin birlikte işletimi sağlanarak ilk mimaride ortaya çıkan paket kayıpları en aza indirgenmiştir.

3.3.2 Tampon saha kullanımı

Tampon sahanın görevi, boruhattının iki safhasına ait farklı işletim hızları arasında uyuma sağlanmasıdır. Paketlerin ağdan gelmesi gereken zamanlar belli olmadığı için, ilk safha sürekli olarak bir döngü içinde paket bekler ve gelen paketleri tampon sahaya yerleştirir (Şekil 3.10). İkinci safha, periyodik olarak paketleri tampon sahadan alarak açma ve oynatma işlemlerini yerine getirir. Periyodiklik, POSIX.4 kütüphanesine ait gerçek zamanlı saatleri kullanarak gerçekleştirilmiştir.

RTP, UDP datagramları üzerine geliştirildiği için video paketleri alıcıya karışık sırada gelebilir. Tampon sahanın başka bir görevi, gelen paketlerin sıralanmasını sağlamaktır. Karışık sırada gelen paketlerin sıralanması için tampon saha bölümlere ayrılmış ve her bir bölüm numaralandırılmıştır. Gelen paketler, RTP paket başlığında bulunan sıra numarasına bakarak uygun bölüme yerleştirilir. Böylece paketlerin sıralanması sağlanır. Ayrıca, verilerin tampon sahadan alınışını daha

hızlı bir şekilde gerçekleştirmek için paketler RTP ve MPEG başlıkları çıkarılarak ilgili bölüme yerleştirilirler.



Şekil 3.10. Tampon saha kullanımı

İletişim başladığında alınan ilk paketler doğrudan açılıp ekranda gösterime sunulmazlar. Gelecekte oluşabilecek tıkanıklık ve 'jitter'in etkilerini hafifletmek için bir süre tampon sahada bekletilirler. Bu tampon sahadaki çerçeveler GOP'lar halinde alınarak başka bir tampon sahaya taşınırlar. Bu ikinci tampon sahada bulunan çerçevelerin açılması ve ekranda gösterilmesi için XIL kütüphanesi kullanılmaktadır. XIL kütüphanesi, çok işparçacıklı mimarileri destekleyebilecek bir görüntü ve video işleme kütüphanesidir. Videoyu en uygun hızda oynatabilmek için molekül kavramını kullanır. XIL kütüphanesine verilebilecek en küçük veri ünitesi GOP'lardır. Bu nedenle, veriler tampon sahadan GOP'lar halinde okunmaktadır. Veriler, paketler halinde tampon sahadan toplandığı için bir GOP'a ait paketler birleştirilerek XIL kütüphanesine verilirler. Okunmakta olan GOP'ta kayıp veya hatalı paketler varsa bunlar belirlenir. MPEG çerçeveleri arasında bağımlılık ilişkisi bulunduğu için çerçevelerden birindeki kayıp, onu izleyen çerçevelerin açılmasına engel olur. XIL kütüphanesi, bu durumdaki çerçeve ve ona bağlı çerçevelerin

yerine son gösterilen çerçevenin oynatılmasını sağlayan mekanizmalara sahiptir.

XIL kütüphanesi son okunan GOP'taki çerçeveleri birer birer olarak ekranda oynatır. Ardışık çerçevelerin oynatılış zamanları arasında $1/\text{çerçeve_hızı}$ saniyelik bir aralık bulunmalıdır. Örneğin, çerçeve hızı 10 çerçeve/saniye ise, her bir çerçeve 100 milisaniye aralıklarla ekranda gösterilmelidir. POSIX 4. Kütüphanesinin gerçek zamanlı saat mekanizması kullanılarak gereksinim duyulan zaman aralığı sağlanmıştır.

3.3.3 Akış Kontrolü

Oynatıcı iş parçacığı, sıkışmış verileri okuma, bu verileri çerçevelere dönüştürme ve düzenli aralıklarla kullanıcıya sunma işlemlerini yerine getirir. XIL kütüphanesinin verileri GOP'lar halinde tampon sahadan alması tampon sahanın doluluk miktarının hızlı bir şekilde düşmesine yol açar. Çerçeveler açılıp oynatılırken, ağdan gelen yeni paketlerle tampon sahanın doluluk miktarı artar. Bu nedenle tampon saha büyüklüğü değişkenlik gösterir (Şekil 3.11).

Sunumun sürekliliği için çerçevelerin tampon sahadan okunuş hızının paketlerin ağdan alınarak tampon sahaya yerleştirme hızıyla aynı olması gerekir. Tampon sahadaki çerçevelerin sayısındaki düşüş önemli bir problemin göstergesi olabilir. Problemlerin kaynağı, ağdaki sıkışıklık, servisçinin gönderim hızının düşük oluşu ya da istemcinin yavaşlığı olabilir. Tampon sahadaki çerçeve sayısının tekrar yükselmeye başlaması ise sistemdeki sorunun giderildiğine dair bir işarettir. Bu nedenlerden ötürü, bir akış kontrol mekanizması ile tampon sahanın doluluğu sürekli kontrol edilir.

Ağdaki sıkışıklık paketlerin gecikmesine ve bir süre sonra paketlerin kaybolmasına neden olur. Paketlerin gecikmesi, tampon sahaya veri ekleme hızının, tampon sahadan veri-okuma hızından küçük olmasına, dolayısıyla tampon sahada bulunan çerçeve sayısının düşmesine neden olur. Bu durum tespit edildiğinde, servisçiden uyum seviyesini yükseltmesi istenir. İstemci de yeni uyum seviyesine bağlı olarak videoyu oynatış hızını düşürür. Uyum seviyesinin artırılması, birim zamanda daha az sayıda çerçevenin ağa konmasına, dolayısıyla ağ yükünün hafiflemesine ve sıkışıklık durumunun giderilmesinde katkı sağlanmasına neden olur. Bir süre sonra tampon sahadaki çerçeve sayısı artmaya başlar. Bu durum, ağdaki sıkışıklığın hafiflediğine dair bir işarettir. Servisçiden uyum seviyesini düşürmesi istenerek eski çerçeve hızına geri dönülür. Çizelge 3.2'de farklı uyum seviyeleri için geçerli video oynatış hızları gösterilmektedir.

Çizelge 3.2. Kullanılan uyum seviyelei için geçerli oynatış hızları

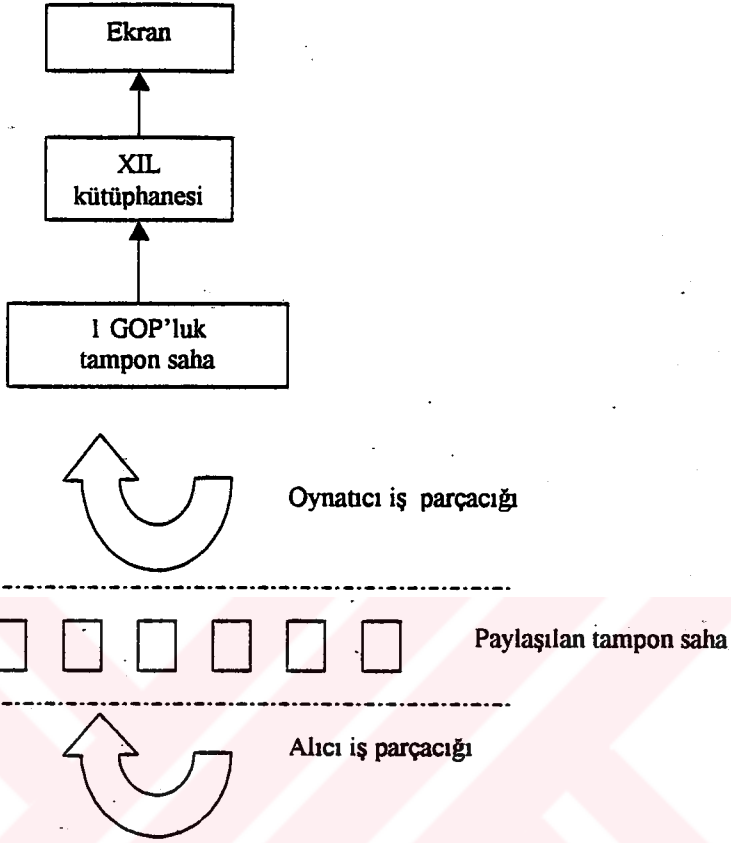
Uyum Seviyesi	Oynatış hızı
1	30
2	20
3	10
4	6
5	2

Servisçinin CPU yükü arttığında servisçi yavaşlar ve paketlerin gönderim aralığı teorik gönderim aralığından daha büyük olacak şekilde değişir. Bu durumda da tampon sahada bulunan çerçeve sayısı düşer. Bu durumda da servisçiden birim zamanda göndermesi istenen çerçeve sayısını düşürmesi istenir. İstemci de gösterim hızını düşürür. Bu şekilde

hem servisçinin yükü hafifletilir, hem de tampon sahadaki çerçeve sayısındaki düşüş engellenir.

Teorik olarak hesaplanan gönderim aralığı, en iyi aralık olmayıp en iyi değere bir yaklaşımdır. Dolayısıyla, fazla CPU yükü olmamasına rağmen servisçi yavaş çalışıyor olabilir. Tampon sahadaki çerçeve sayısındaki düşüşün asıl nedenini belirlemek için önce servisçiden gönderim aralığını azaltması istenerek servisçi hızını arttırmaya zorlanır. Buna rağmen tampon sahadaki çerçeve sayısında düşüş hala devam ediyorsa servisçinin yükünün fazla olması olasılığı var demektir ve servisçiden uyum seviyesini arttırarak birim zamanda gönderilen çerçeve sayısını azaltması istenir.

Akış kontrolü, tampon sahadaki taşmaları ve boşalmaları önler. Taşmalar çerçeve kayıplarına, boşalmalar sunum sırasında duraklamalara neden olurlar. Akış kontrol modülü, servisçiye tampon sahanın durumu hakkında bilgi verir. İstemci, servisçinin mutlak gönderim hızını belirlemez. Sadece tampon sahada tutulan çerçevelerin sayısını kontrol ederek gerektiğinde servisçiden gönderme hızını değiştirmesini ister.



Şekil 3.11. Alıcıda kullanılan tampon sahalar

Akış kontrolü için kullanılabilecek bir yöntem, tampon saha için alt ve üst sınırlar belirleyerek tampon sahadaki çerçeve sayısını bu sınırlarla karşılaştırmaktır (Hess, 1998). Çerçeve sayısının alt sınırın altına düşmesi servisçi hızı ya da ağdan kaynaklanan bir sorunun varlığına işaretler. Bu durumda servisçinin paketler arası gönderim zamanını düşürmesi, ya da problemin devam etmesi durumunda birim zamanda gönderilen çerçeve sayısını attırması istenir. Çerçeve sayısı üst sınırdan büyükse, servisçinin gönderim hızı alıcının tampon sahadan alış hızından daha büyük demektir. Bu durumda servisçiden paket gönderim

aralığını arttırması istenir. Servisçi, gönderim aralığını sabit bir değerle çarparak değiştirir. Bu yaklaşımın dezavantajı, çerçeve sayısındaki düşüş ya da artışın ancak sınır değerler dışına çıktığında fark edilmesidir.

Akış kontrolü için bizim geliştirdiğimiz daha iyi bir çözüm, bu sınır değerler aşılmadan tehlikenin farkına varmaktır. Bunun için yakın geçmişte tampon sahada bulunan çerçeve sayısı istatistiklerine bakarak düşüş ve artışları belirlemektir. Bu yaklaşımın harcayacağı CPU zamanını düşürmek için kullanılabilecek daha pratik bir yöntem, tampon sahayı bölgelere ayırarak çerçeve sayısının hangi bölgeye düştüğünü bulmak ve bu bölgeye göre gerektiğinde servisçiden paket gönderim aralığını uygun bir katsayı ile çarpmasını istemektir. Bu durumu şöyle bir örnekle açıklayabiliriz. Tampon sahada az sayıda GOP bulunması tampon sahanın boşalma olasılığını yükseltir. Bu durumda, servisçiden gönderim aralığını düşürmesi istenir. Servisçinin kullanacağı katsayı küçüldükçe veri gönderim aralığı da küçülür. Buna bağlı olarak tampon sahadaki çerçeve sayısındaki artış hızı da yükselir. O yüzden tampon sahada az veri kaldığında, tamamen boşalma olasılığını düşürmek için servisçiden gönderim aralığını küçük bir katsayı ile çarpması istenir. Ancak henüz boşalma tehlikesi yoksa ve tampon sahadaki çerçeve sayısı düşmekteyse gelecekte meydana gelebilecek boşalma olasılığını düşürmek için servisçiden gönderim aralığını daha büyük bir katsayı ile çarpması istenebilir. Tampon sahanın bölgelere ayrıldığı bu yaklaşım, akış kontrolünün daha etkin bir şekilde gerçekleştirilmesini sağlar.

Geliştirdiğimiz sistemde, tampon saha altı bölgeye ayrılmıştır: kararlı bölge, taşma bölgesi, taşma-uyarı, tehlike, tehlike-uyarı ve boşalma bölgeleri (Şekil 3.12). Tampon sahadaki veri miktarı 1.5 dakikayı geçiyorsa *taşma* durumu, 0.2 dakikadan kısa sürelik videoyu içeriyorsa *boşalma* durumu ortaya çıkar.

Taşma Bölgesi	1.5 dak
Taşma Uyarı Bölgesi	1.15 dak.
Kararlı Bölge	0.85 dak.
Tehlike Uyarı Bölgesi	0.5 dak
Tehlike Bölgesi	0.2 dak.
Boşalma Bölgesi	

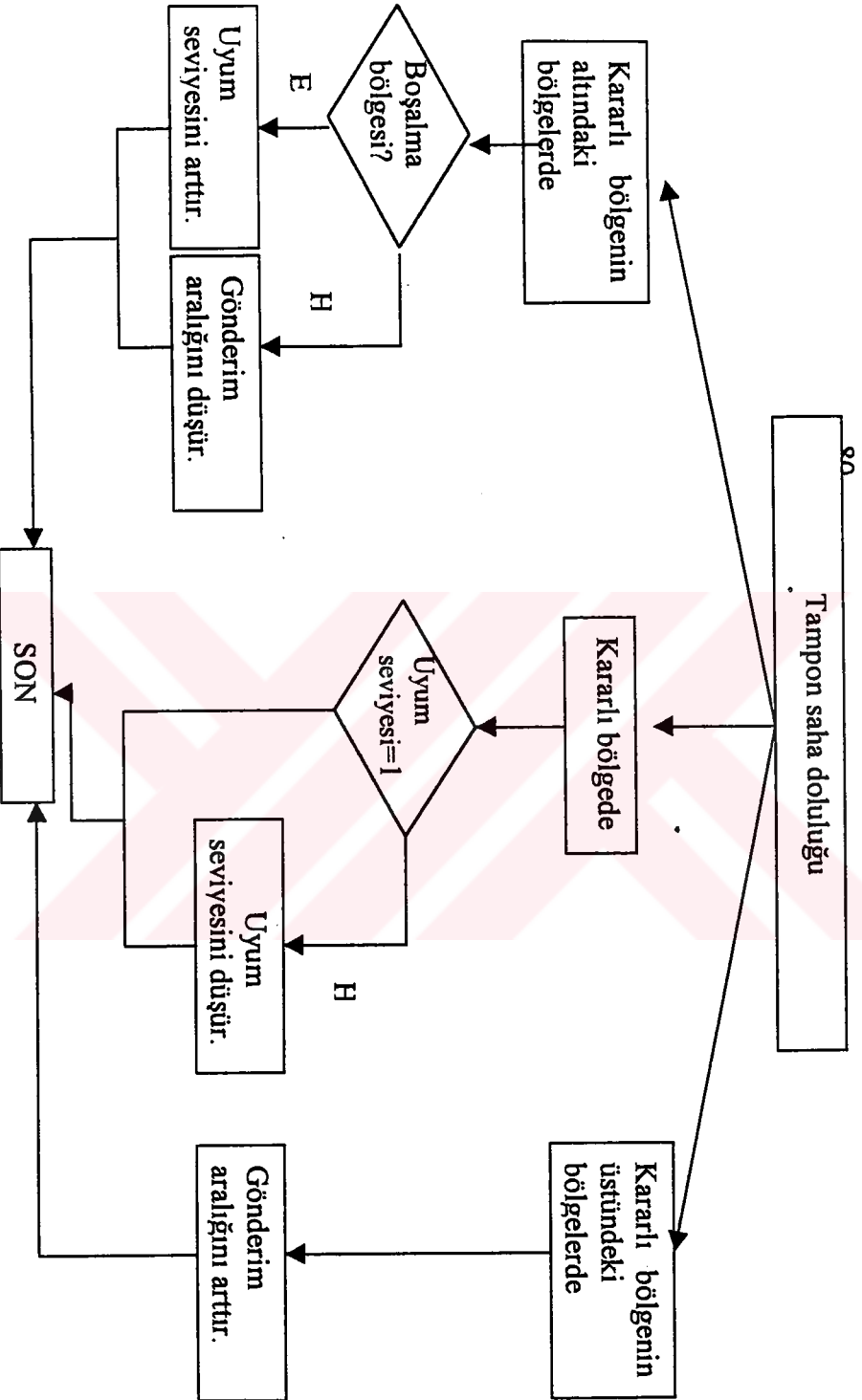
Şekil 3.12. Akış kontrolü işlemi için tampon sahada ayrılan bölgeler.

Her 10 GOP'un gösteriminden sonra tampon sahanın doluluk miktarı kontrol edilir. Boşalma bölgesine girilince uyum seviyesi yükselterek çerçeve hızının düşmesi sağlanır. Amaç, tampon sahadaki çerçeve sayısını kararlı bölgenin sınırları içinde tutmaktır. Tampon sahanın çerçeve sayısı taşma uyarı/tehlike uyarı bölgelerine düşmüşse, servisçiden gönderim aralığını yükseltmesi/düşürmesi istenir. Gönderim aralığı, teorik olarak hesaplanmış gönderim aralığının sabit katsayılarla çarpılmasıyla elde edilir. Bu katsayılar taşma, taşma uyarı, tehlike uyarı ve tehlike bölgeleri için sırasıyla 1.2, 1.1, 0.9 ve 0.8 olarak belirlenmiştir.

Gönderim hızının belli bir değerin altına düşmesi durumunda, servisçiden uyum seviyesini arttırarak birim zamanda gönderilecek çerçeve sayısını düşürmesi istenir. Katsayı belli bir değerin üstünde olursa servisçiden uyum seviyesini düşürerek birim zamanda gönderilecek çerçeve sayısını arttırması istenir.

Şekil 3.13 'de kullanılan akış kontrol mekanizmasına ait akış diyagramı görülmektedir.





Şekil 3.13. Kullanılan akış kontrol yöntemi

3.4 Kontrol Mesajları

Servisçi ve istemci arasında mesajlaşma bir kontrol kanalı üzerinden yapılır. İletişimin güvenilirliğini sağlamak için kontrol kanalı TCP protokolü üzerine geliştirilmiştir. Uygulamada kullanılan kontrol mesajları aşağıda listelenmiştir.

CMD_INIT : Bir istemci servisçiye bu mesajla bağlanır.

CMD_START : İstemci, servisçiden veri göndermesini bu mesajla ister.

CMD_STOP : İstemci, servisçiden veri gönderimini durdurması için bu mesajı kullanır.

CMD_FASTEN: Akış kontrolü sırasında tampon sahanın doluluğu düşünce servisçiden veri gönderim aralığını düşürmesi bu mesajla istenir.

CMD_SLOW: Akış kontrolü sırasında tampon sahanın doluluğu bir sınırı geçince, servisçiden veri gönderim aralığını arttırması bu mesajla istenir.

CMD_ADAPT1 : Akış kontrolü sırasında uyum seviyesini yükseltmek için kullanılır.

CMD_ADAPT2 : Akış kontrolü sırasında uyum seviyesini düşürmek için kullanılır.

3.5 İş Dağıtımı

Servisçi ve istemci yazılımlarındaki iş parçacıklarının iş dağıtımı POSIX 4 kütüphanesinin gerçek zamanlı iş dağıtımı olanaklarıyla gerçekleştirilmektedir.

POSIX standartları SCHED_RR, SCHED_FIFO ve SCHED_OTHER olmak üzere üç tür gerçek zamanlı iş dağıtım sınıfına sahiptir. Solaris işletim sistemi sadece SCHED_FIFO gibi çalışan SCHED_OTHER iş dağıtım sınıfını destekler. Sistemimize uygun çalışma prensibi SCHED_FIFO olduğu için herhangi bir sorun çıkmadan iş parçacıkları çalıştırılabilmektedir.

Gerçek zamanlı süreçlerin iş dağıtım sırasında bir öncelik değeri atanması gerekmektedir. Öncelik değerleri çerçeve hızına bakarak RM (Rate Monotonic) prensibine uygun olarak atanır. Şu an veri tabanımızda sadece 30 çerçeve/sn. lik çerçeve hızına sahip videolar bulunduğu için bu değer tüm servisçi iş parçacıkları için sabittir.

İstemci tarafında temel olarak iki iş parçacığı çalışmaktadır. Bunlardan biri, ağdan gelen paketleri alır, diğeri oynatır. Ağdan paket alma işi daha önemli bir iş olduğu için ağdan paket alan iş parçacığına daha büyük öncelik değeri verilmiştir.

4. BAŞARIM SONUÇLARI

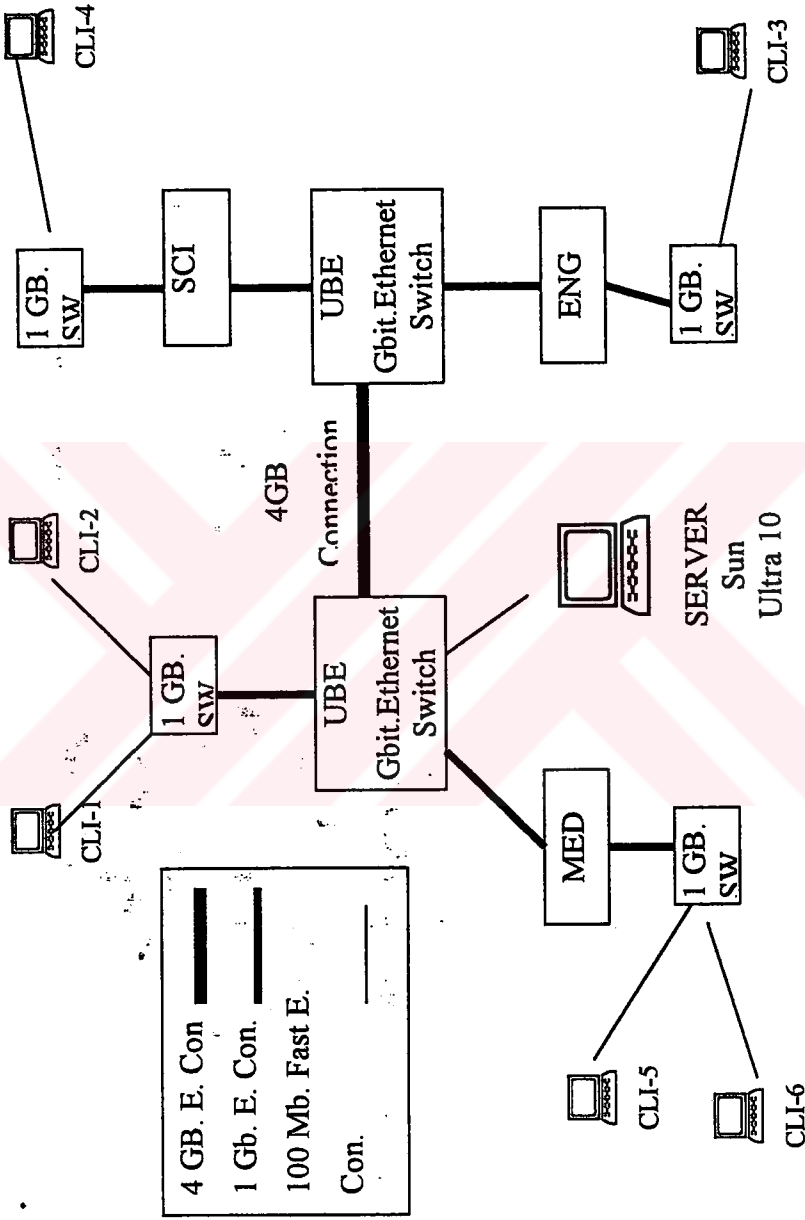
4.1 Altyapı

Sistemimiz, Gigabit Ethernet omurgası üzerine kurulu Ege Üniversitesi Kampüs Yerel Ağı'nda test edilmiştir. Üzerinde uygulamamızın denendiği kampus ağ alt yapısının parçası Şekil 4.1'de verilmektedir.

UBE ve MK anahtarları 4 Gbit/sn kapasiteli Ethernet hattıyla birbirine bağlanmıştır. Servisçi olarak doğrudan UBE anahtarına bağlı bir Sun ULTRA 10 iş istasyonu kullanılmıştır. İstemci olarak Tıp, Mühendislik, Fen Fakültesi ve UBE anahtarlarına bağlı Sun Ultra 1 ve Sun Ultra 5 iş istasyonları kullanılmıştır. İstemciler, 100 Mbitlik Fast Ethernet hatlarıyla ilgili anahtarlara bağlıdır.

4.2 Video Veritabanı

Veri tabanı olarak 'MPEG-1 video only' formatıyla sıkıştırılmış video verileri kullanılmıştır. MPEG-1 ve MPEG-2 videolarının yapısı birbirine çok benzediği için, MPEG-1 ve MPEG-2 verilerini RTP paketlerine bölme kuralları hemen hemen aynıdır. Paketleme modülü geliştirilirken MPEG-2 verilerinin de desteklenmesine dikkat edilmiştir. Ancak, MPEG-2 kablolu ve etkileşimli TV sistemlerinde daha yüksek kalitede görüntü olanağı sağlamak için geliştirildiğinden daha yüksek sıkıştırma bit hızına ("encoding bit rate") gereksinim duyar. Sıkıştırma bit hızı, hedef sıkıştırma hızının Kbps cinsinden değeridir. Örneğin, 100 Kbps'lik sıkıştırma hızından bahsedildiğinde, her bir saniyelik videonun



Şekil 4.1 Ağ topolojisi

100 Kbit yer tutacak şekilde sıkıştırıldığı anlaşılr. Kayıt sırasında sıkıştırma bit hızı yükseldikçe verilerin kalitesi artar. Bu durum, videoyu saklamak için daha büyük bir disk sahasının kullanılmasını gerektirir ve iletişim sırasında ağ yükünün artmasına neden olur. Öte yandan, günümüzün Internet ortamı ve CPU hızları, bu denli yüksek miktarda veriyi gerçek zamanda işlem den geçirme konusunda yetersizdir. İnternet uygulamalarında hedef, TV kalitesinde görüntü elde etmek değil, verilerin kabul edilebilir bir kalitede gerçek zamanlı olarak oynatılabilmesinin sağlanmasıdır.

Internet gibi düşük hızlı iletişim ortamları için kullanılabilecek başka bir MPEG versiyonu, MPEG-4 sıkıştırma algoritmasıdır. RTP protokolünün MPEG-4 verilerini paketleme kuralları bulunmasına rağmen, MPEG-4 kaydı için donanım ve yazılım olanaklarının bulunmayışı ve XIL kütüphanesinin sadece MPEG-1 verilerini destekleyebilmesi nedenlerinden ötürü, veri tabanının sadece MPEG-1 videolarından oluşmasına karar verilmiştir.

Şu anki veri tabanı, 10 dakikalık bir videonun iki farklı çözünürlük (160x120 ve 320x240) ve 5 değişik bit hızında (100 Kbps, 200 Kbps, 500 Kbps, 1000 Kbps ve 1500 Kbps) sıkıştırılmış versiyonlarından oluşan 10 videoyu içermektedir. Internet akışları için 200 Kbps ve daha düşük bit hızları yeterli olmasına rağmen sistemin davranışını daha kapsamlı bir şekilde gözleyebilmek için 500 Kbps ve daha yüksek bit hızlarında da örnekler kullanılmıştır. Çizelge 4.1'de veritabanında bulunan videolara ilişkin istatistikler sunulmaktadır. Çizelge 4.1'den video çözünürlüğünün sıkıştırılmış verilerin miktarı üzerinde hiçbir etkisinin olmadığı görülmektedir. Videoların büyüklüğü sıkıştırma sırasında kullanılan bit hızının bir fonksiyonudur. Örneğin, 1 dakikalık bir MPEG-1 videosunun

disk üzerinde kaplayacağı alan yaklaşık olarak aşağıdaki şekilde hesaplanabilir.

$$1 \text{ dakikalık MPEG-1 video (Kbytes)} = \text{Video bit hızı (Kbps)} * 60 / 8$$

Bu değere GOP, resim, kesit vb. başlıklarının eklenmesinden doğan yaklaşık %5 lik bir yükü de eklemek gerekmektedir.

Çizelge 4.1. Video veritabanını oluşturan videolara ait özellikler

Ad	Çözünürlük	Bit hızı	Süre	Büyüklik
Bel1.mpg	320x240	100 Kbps	10 dak.	7.996.090 byte
Bel2.mpg	160x120	100 Kbps	10 dak	7.609.642 byte
Bel3.mpg	320x240	200 Kbps	10 dak	15.220.471 byte
Bel4.mpg	160x120	200 Kbps	10 dak	15.221.924 byte
Bel5.mpg	320x240	500 Kbps	10 dak.	38.048.024 byte
Bel6.mpg	160x120	500 Kbps	10 dak	38.048.025 byte
Bel7.mpg	320x240	1000 Kbps	10 dak	76.096.298 byte
Bel8.mpg	160x120	1000 Kbps	10 dak	76.096.024 byte
Bel9.mpg	320x240	1500 Kbps	10 dak	114.144.024 byte
Bel10.mpg	160x120	1500 Kbps	10 dak	114.144.024 byte

Sıkıştırma sırasında kullanılan bit hızı arttıkça videonun kalitesi de artmaktadır. Videonun içeriğinin dinamikliğine bağlı olarak, belli bir sıkıştırma hızının altına düşüldüğünde görüntü kalitesindeki düşüş açıkça görülebilmektedir. Sistemin geliştirilmesi sırasında ilk deneyler ameliyat videoları ile gerçekleştirilmiştir. Ameliyat videolarının bir özelliği video içeriğinin oynatış boyunca çok az değişim göstermesi ve kullanılan renk sayısının düşük olmasıdır. Daha sonraki deneylerde içeriği daha zengin ve dinamik olan belgesel videoları kullanılmıştır. 100 Kbpslik bit hızında sıkıştırılmış ameliyat ve belgesel videoları incelendiğinde, sunum sırasında, çerçeveler arasındaki değişkenliğin düşük olması sebebiyle ameliyat videolarının daha yüksek bir görsel kaliteye sahip olduğu görülmüştür.

Veri tabanında her video ile birlikte o videoya ait istatistikleri tutan bir dosya bulunmaktadır. Sisteme yeni bir video katıldığında ilk önce bir program aracılığı ile o videoya ait istatistikler çıkarılır. İletişim sırasında, çerçeve büyüklüğü istatistikleri kullanılarak paketler arası gönderim aralığı hesaplanır. Bir istatistik dosyasının formatı Şekil 4.2 de gösterilmiştir.

```

Video büyüklüğü
0.,1,...,5. uyum seviyeleri için gönderilecek paket sayıları
Dizi, GOP, çerçeve sayıları
GOP patterni
/** Dizi bilgileri **/
Dizi 1 - başlangıç adresi, büyüklüğü, çözünürlük, çerçeve hızı, sıkıştırma bit hızı
Dizi 2 - ....
...
/** GOP bilgileri **/
GOP 1 - başlangıç adresi, büyüklüğü, çerçeve sayısı
GOP 2 - ...
...
/** Çerçeve bilgileri **/
Çerçeve 1 - tür, büyüklük
Çerçeve 2 - ...
....

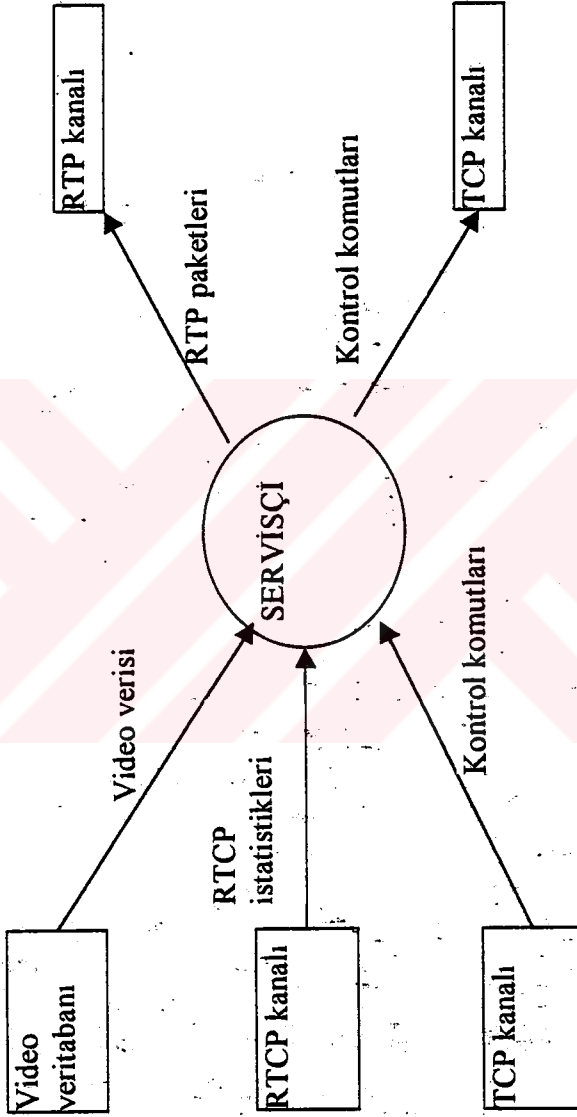
```

Şekil 4.2 Bir MPEG videoya ait istatistiklerin bulunduğu dosyanın görüntüsü

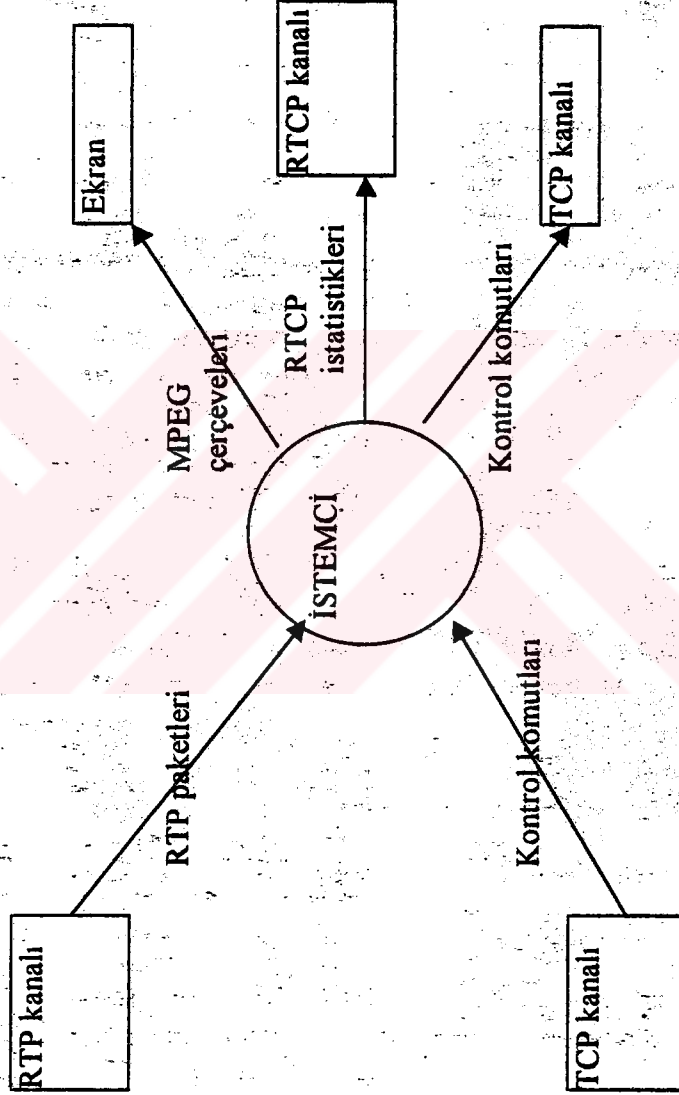
4.3 Sistem Geliştirilmesi Sırasında Kullanılan Araçlar

Servisçi yazılımını C++ dilinde çok iş parçacıklı programlama olanaklarına sahip POSIX iş parçacığı ve gerçek zamanlı programlama olanaklarına sahip POSIX 4 gerçek zamanlı programlama kütüphanesi kullanılarak geliştirilmiştir.

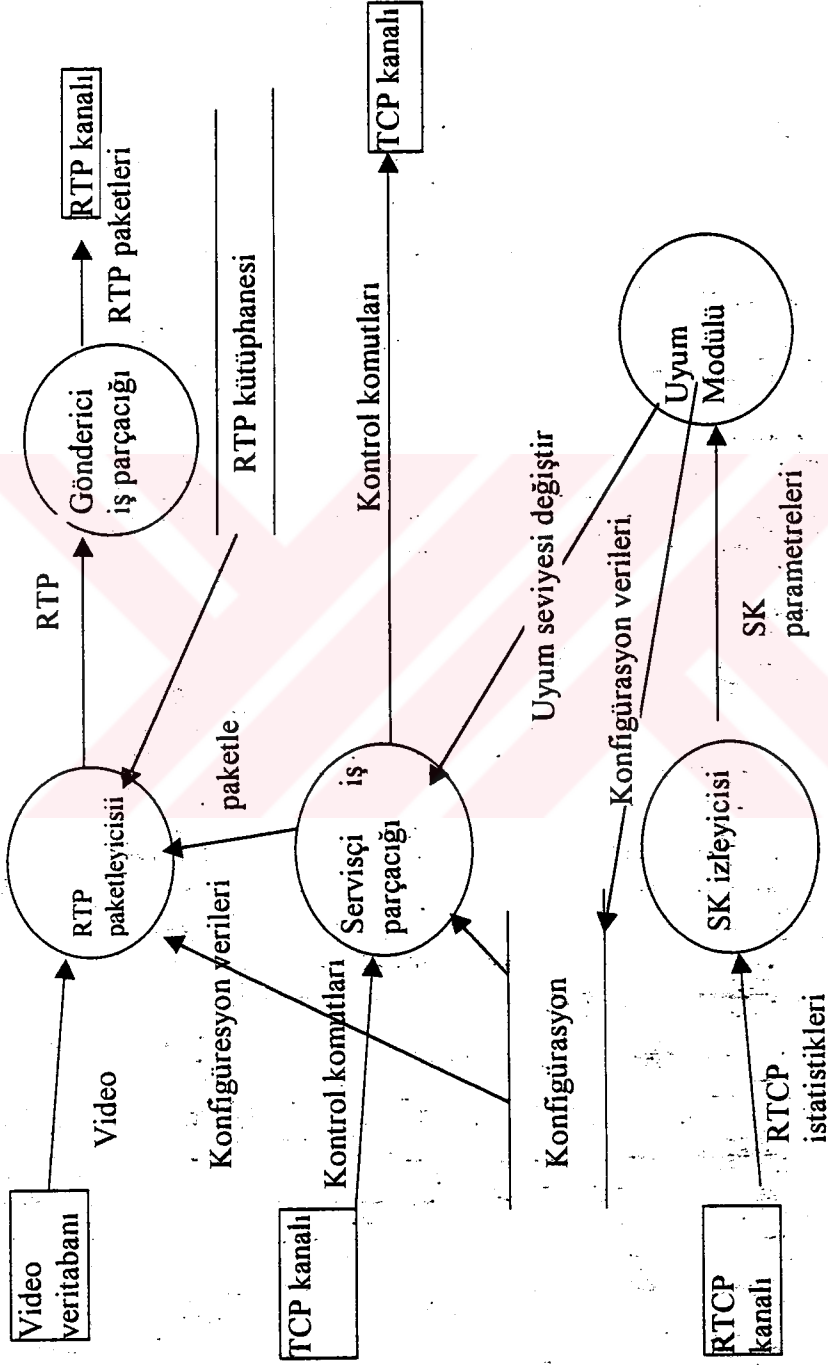
İstemci yazılımını C ++ dilinin getirdiği yükü hafifletmek amacıyla C diliyle, POSIX iş parçacığı ve gerçek zamanlı programlama kütüphaneleri ve görüntüleme amaçlı XIL kütüphanesi kullanılarak oluşturulmuştur.



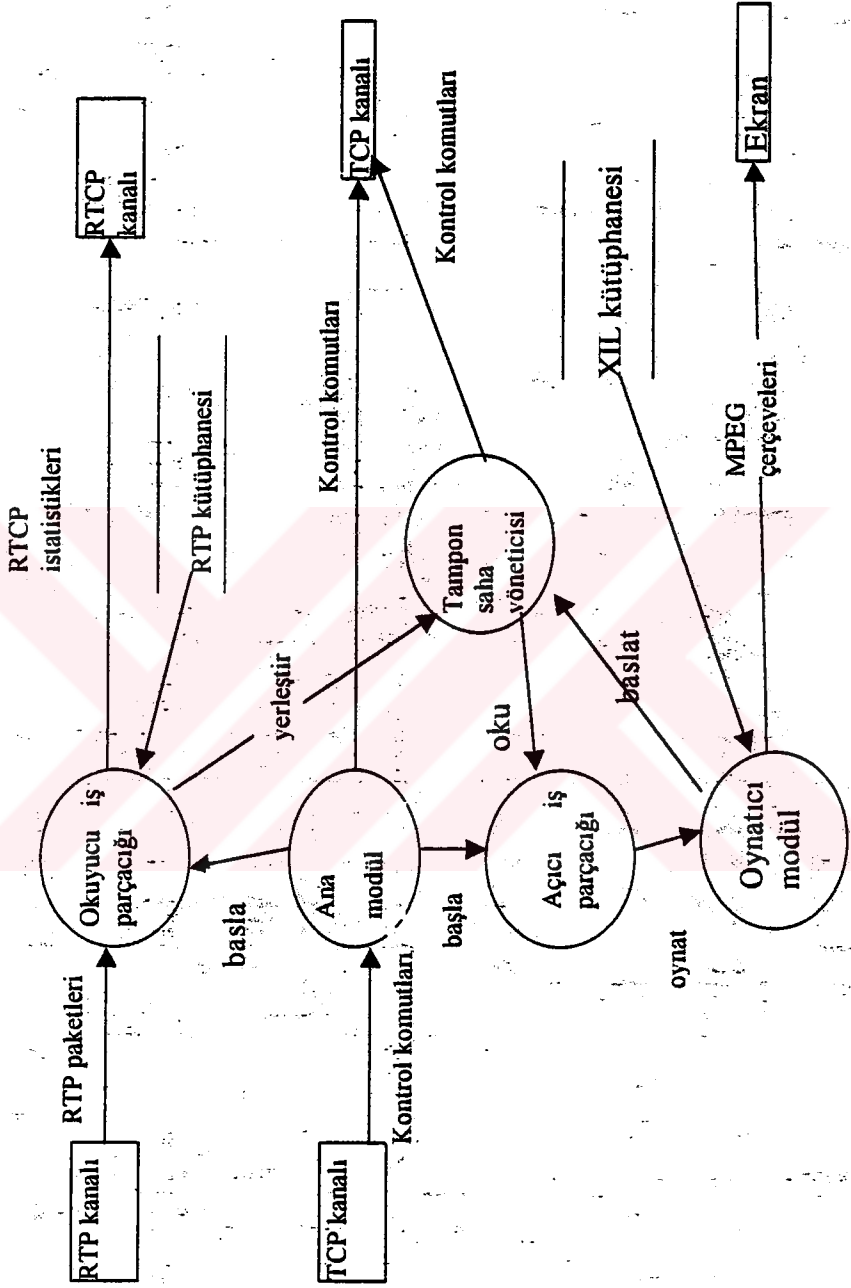
Şekil 4.3. Servisçi modülüne ait 0. seviye akış diyagramı



Şekil 4.4 İstemci yazılımına ait 0. seviye akış diyagramı



Şekil 4.5 Servisçi modülü 1. seviye akış diyagramı



Şekil 4.6 İstemci modülü 1. seviye akış diyagramı

Ölçeklendirme ('scaling'): Açma sonrasında elde edilen değerlerin renklendirme işlemi için belirli aralıklarda ölçeklendirilmesi.

Renklendirme ('dithering'): Ekranda videonun oynatılacağı penceredeki piksellere ait renk değerlerinin belirlenmesi ve piksellerin renklendirilmesi.

Alınan ölçümlere göre XIL kütüphanesinin açma ve ölçeklendirme işlemleri çok kısa bir süre içinde tamamlanmaktadır. Sunum sırasında asıl zaman kaybı renklendirme işlemi sırasında ortaya çıkmaktadır. Çizelge 4.3 de farklı çerçeve hızları ve çözünürlüklerde XIL kütüphanesinin harcadığı süreler verilmektedir. Bu değerler bir MPEG oynatıcısının Sun Ultra 5 iş istasyonu üzerinde çalıştırılması sırasında ölçülmüştür.

Harcanan sürenin 320x240 çözünürlüğünde artmasının sebebi, bu çözünürlükte daha fazla sayıda pikselin renk değerinin belirlenmesidir. Renklendirme işlemi CPU yükünü çok arttıran bir işlem olduğundan işleme alınan piksel sayısındaki artış, çözünürlük arttığında renklendirme için harcanan sürede büyük bir artışa neden olmaktadır.

Çizelge 4.3'ten çıkan başka bir sonuç, tüm çerçeve türleri için harcanan CPU zamanının birbirine yakın oluşudur. Genel olarak I çerçeveleri P çerçevelerinin iki katı, P çerçeveleri B çerçevelerinin iki katı büyüklüğündedir. Çerçevelerin disk üzerinde kapladığı saha, sadece açma ve ölçeklendirme sırasında etkili olmaktadır. Bu iki işlem, renklendirmeye göre daha az CPU süresi kullandığından toplam CPU harcamasındaki payları düşüktür. Renklendirme için harcanan süre çözünürlüğe bağlı olduğundan ve her çerçeve aynı çözünürlüğe sahip olduğundan harcanan sürenin büyük bir kısmı renklendirme işleminden

3. Genel olarak, RTP protokolü için harcanan süre, paketleme yükü için harcanan süreden daha büyüktür. Bu fark bit hızı arttıkça daha da belirginleşmektedir.

Bu ölçümler servisçinin çalıştığı Sun Ultra 10 iş istasyonunda alınmıştır. Ölçümler alındığı anda birçok sistem programı da çalıştığından elde edilen değerler kesin değerler değildir. Kesin değerler hiçbir yazılımın çalışmadığı çıplak bir makine üzerinde servisçi yazılımının çalıştırılmasıyla elde edilebilir. Bu nedenle bu değerler, kullanıcı sayısının az olduğu bir anda, birkaç oynatış için harcanan sürenin ortalamasının alınmasıyla elde edilmişlerdir.

Çizelge 4.2 sadece çözünürlüğün 320x240 olduğu durum için geçerlidir. Aynı bit hızıyla sıkıştırılmış 160x120 çözünürlüğe sahip videolar, 320x240 çözünürlüğe sahip videolarla yaklaşık olarak aynı büyüklükte olduklarından servisinde harcanan süre iki çözünürlük için de birbirine yakındır. Bu nedenle 160x120 çözünürlük için gereken CPU zamanına ilişkin gözlemler verilmemiştir.

4.4.2 İstemci Yükü

İstemcide harcanan sürenin büyük bir kısmı sıkıştırılmış videonun açılması ve ekranda oynatılması sırasında ortaya çıkmaktadır. Bu yüzden sadece XIL kütüphanesinden kaynaklanan zaman harcaması üzerinde durulmuştur. XIL kütüphanesi sıkıştırılmış verileri üç aşamadan geçirir:

Açma ('decoding'): Sıkıştırılmış verilerin orijinal haline getirilmesi.

Çizelge 4.2. Servisçi tarafında paketlenme ve RTP kütüphanesi harcamaları (milisn cinsinden)

	Uyum Seviyesi 0 (30 fps)				Uyum Seviyesi 1 (20 fps)				Uyum Seviyesi 2 (10 fps)			
	CPU	RTP	CPU+RTP	M	CPU	RTP	CPU+RTP	M	CPU	RTP	CPU+RTP	M
100 Kbps	789	2102	2891	208	522	1126	1648	364	525	688	1213	495
200 Kbps	2135	3295	5430	110	1136	1390	2526	238	1008	1094	2102	285
500 Kbps	3346	4886	8232	73	2136	2237	4373	137	1234	2778	4012	150
1000 Kbps	6485	12085	18570	32	4005	4515	8520	70	3947	4461	8408	71
1500 Kbps	9770	17932	27702	22	6034	8238	14272	42	5609	7836	13445	45

	Uyum Seviyesi 3 (3 fps)				Uyum Seviyesi 4 (1 fps)			
	CPU	RTP	CPU+RTP	M	CPU	RTP	CPU+RTP	M
100 bps	523	332	855	702	462	190	652	920
200 Kbps	1012	568	1580	380	1006	236	1242	483
500 Kbps	1192	1972	3164	190	1110	1359	2469	243
1000 Kbps	2596	3694	6290	95	2463	2425	4888	123
1500 Kbps	5869	4466	10335	58	5385	2462	7847	76

4.4 Sistem Yüğü

4.4.1 Servisçi Yüğü

Çizelge 4.2 'de tek bir videonun gönderimi için servisçide harcanan süre verilmektedir. Servisçide harcanan süre iki elemandan oluşmaktadır:

1. Gönderim sırasında RTP kütüphanesi tarafından harcanan CPU süresi
2. Paketleme sırasında harcanan CPU süresi.

Her uyum seviyesi aynı zamanda farklı bir çerçeve hızına karşılık gelmektedir. Çizelge 4.2, servisçide harcanan sürenin çerçeve hızı ve sıkıştırma bit hızına bağlı olduğunu göstermektedir. Ayrıca, Çizelge 4.2' de, M sütununda, video süresinin toplam servisçi yüküne bölünmesiyle elde edilen aynı anda servis görebilecek teorik istemci sayısı gösterilmektedir.

Çizelge 4.2 incelendiğinde ortaya çıkan sonuçlar şöyle listelenebilir:

1. Uyum seviyesi arttığında harcanan toplam süre azalmakta ve teorik olarak desteklenebilecek istemci sayısı artmaktadır.
2. Sıkıştırmada kullanılan bit hızı arttıkça hem paketleme hem de RTP protokolü için harcanan süre artmakta, buna bağlı olarak aynı anda desteklenen istemcilerin teorik sayısı da düşmektedir.

gibi ilgili konfigürasyon parametrelerinin değişimini ve servisçi iş parçacığının bu değişikliklerden haberdar olmasını sağlar. Servisçi ve istemci arasındaki kontrol mesajları TCP kanalı üzerinden taşınır.

Şekil 4.6'da istemci modülünü oluşturan alt modüller detaylı olarak verilmektedir. İstemci modülü temel olarak iki iş parçacığından oluşmaktadır. Ana modül, okuyucu ve açıcı iş parçacıklarını harekete geçirdikten ve servisçiyle arasındaki veri iletimini başlatmak için gerekli kontrol mesajlarının TCP kanalı üzerinden iletimini sağlamakla görevlidir. Tampon saha ile ilgili akış kontrol işlemleri oynatıcı modül tarafından başlatılır. Tampon saha yöneticisi, tampon sahanın doluluğunu kontrol ederek boşalma ve taşma durumlarında servisçiye uygun komutlar göndererek servisçinin uygun işlemleri başlatmasını sağlar. Açıcı iş parçacığı, tampon sahadan aldığı verileri açarak oynatıcı modülün MPEG çerçevelerini ekranda göstermesini sağlar. Oynatıcı modül sunum amaçlı olarak XIL kütüphanesini kullanmaktadır.

İletişim için kullanılan RTP kütüphanesi Lucent Technologies'ten temin edilmiştir. Bu kütüphane, çok iş parçacıklı mimarileri desteklemediği için çok iş parçacıklı programlamada ihtiyaç duyulan senkronizasyon olanakları tarafımızca ayrıca geliştirilmiştir.

Şekil 4.3'de servisçi yazılımına ait 0. seviye akış diyagramı görülmektedir. Servisçi yazılımı üç tip girdi ile çalışmaktadır. MPEG video verileri bir MPEG video veri tabanından alınmaktadır. Servis kalitesi kontrol işlemlerinde kullanılacak RTCP istatistikleri servisçi ve istemci arasında kurulan RTCP kanalı üzerinden alınmaktadır. Alıcıya gelen kontrol mesajları da bir TCP kanalı vasıtası ile alınmaktadır. Servisçi modülü iki tür çıktı üretmektedir. Birincisi RTP kanalı ile istemciye gönderilen RTP paketleri, diğeri ise istemciye TCP kanalı üzerinden gönderilen kontrol mesajlarıdır.

Şekil 4.4'de istemci yazılımına ait 0. Seviye akış diyagramı görülmektedir. İstemci, iki tür veri ile çalışır. Birincisi RTP kanalı ile servisçiden gelen RTP paketleri, ikincisi TCP kanalı vasıtasıyla servisçiden gönderilen kontrol mesajlarıdır. İstemci üç tür çıktı üretir. Ağdan alınan RTP paketleri MPEG çerçevelerine dönüştürülerek ekranda görüntülenir. İletişim kalitesini gösteren RTCP istatistikleri RTCP kanalı üzerinden servisçiye gönderilir. *Hızlan, yavaşla, uyum_seviyesini_değiştir* gibi kontrol mesajları TCP kanalı ile servisçiye gönderilir.

Şekil 4.5'de servisçiye ait 1. Seviye akış kontrol diyagramında servisçi modülü alt modüllere ayrılarak gösterilmektedir. Her istemciye ait işlemleri yerine getirmek üzere ayrı bir iş parçacığı yaratılmaktadır. Servisçi iş parçacığı *RTP_paketleyicisi* ve *RTP_kütüphanesi*yle ortak çalışarak paketlerin gönderici iş parçacığı tarafından uygun zamanlarda ağa bırakılmasını sağlamaktadır.

Servis kalitesi izleyici modülü RTCP kanalından gelen RTCP paketlerinde tutulan paket kaybı istatistiklerini analiz ederek gerektiği durumlarda uyum modülü ile ilişkiye geçer. Uyum modülü çerçeve hızı

ÖZGEÇMİŞ

1970 yılında İzmir'de doğan Aylin Kantarcı, İzmir Eşrefpaşa Lisesi'nden 1988 yılında mezun oldu. 1992 yılında, Ege Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünden mezun oldu. Daha sonra aynı bölümden yüksek lisans derecesini 1994 yılında aldı ve 1994 yılında yine aynı bölümde doktora çalışmasına başladı. Halen Ege Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nde Araştırma Görevlisi olarak görev yapmaktadır.

KAYNAKLAR (devam)

- Shepherd W. D., Coulson G., Garcia F., and Hutchison D., 1991,** Protocol Support for Distributed Multimedia Applications, Proc. Second International Workshop on Network and Operating Systems Support for Digital Audio and Video.
- Steinmetz R. and Nahrstedt K., 1995,**Multimedia Computing, Communications and Applications, Prentice Hall Inc.
- Stevens W. R., 1998,** Unix Network Programming, Volume 1., Networking APIs: Sockets and XTI, Prentice Hall Ptr.
- Stevens W. R., 1999,** Unix Network Programming, Volume II., Interprocess Communication, Prentice Hall Ptr.
- Tan S., 1999,** Multimedia Network Subsystem Design, PhD. Thesis at University of Illinious at Urbana-Champaign.
- Thomas S. A., RTP for Real-Time Applications, IPng and the TVP/IP Protocols: Implementing the Next Generation Internet", John Wiley & Sons, 1996.**
- Tindel K. and Hunson H., 1995,** Realtime Systems and Fixed Priority Scheduling, Department of Computer Systems, Upsala University.
- Yeadon N., Garcia F., Hutchison D. and Shephard D., 1996,** Filters: WoS Support Mechenism for Multipeer Communications, IEEE Journal on Selected Areas in communications. Sept, 1996, 1245-1262 pp.

KAYNAKLAR (devam)

Patel M. and Rowe L.A., Design and Performance of the Berkeley Continuous Media Toolkit, SPIE Proceedings, Vol. 3020, California, February, 1997.

Pingali S., 1994, Protocol and RealTime Scheduling Issues for Multimedia Applications, PhD. Dissertation, University of Massachusetts.

Raghavan S. an Tipathi S. K., 1994, Networked Multimedia Systems: Concepts, Architecture and Design, Prentice Hall, New Jersey.

Rowe L. A., Patel K. D, SmithB.C. and Liu K., 1994, MPEG Video in Software: Representation, Transmission and Playback, IS&T/SPIE 1994 International Symposium on Electrical Imaging Science and Technology.

Savetz K., Randall N., and Lepage Y., 1996, Mbone: Multicasting Tomorrow's Internet, IDC Press.

Schulzrinne H., Casner S., Frederick R., and Jacobson J., 1996, RTP:A Transport Protocol for Real-Time Applications, RFC 1889.

Schulzrinne H., 1998, RTP Profile for Audio and Video Conerences with Minimal Control, IETF Internet. Draft draft-ietf-avt-profile-new-04.ps.

KAYNAKLAR (devam)

- Liu C.L. and Layland J. W.**, 1973, Scheduling Algorithms for Multiprogramming of Hard Real-Time Environment, Journal of ACM.
- Mercer W. C., Savage S. and Tokuda H.**, 1994, Processor Capacity Reserves for Multimedia Operating systems, Proceedings of IEEE International Conference on Multimedia Computing and Systems.
- MPEG 7 Requirement Group**, ISO/IEC JTC1/SC29/WG11, MPEG-7 Faq, 1997.
- Mullender S.**, 1993, Distributed systems, Addison Wesley.
- Nahrstedt K.**, 1995, An Architecture for End to End Quality of Service Provision and its Experimental Validation, PhD. Thesis at University of Pennsylvania.
- Nahrstedt K. and Steinmetz R.**, 1995, Resource Management in Networked Multimedia Systems, IEEE Multimedia Magazine.
- Ng J. Kee-Yin, Wai H. K, Xiong S. and Du X.**, 1998, A distributed MPEG video player system with feedback and QoS Control, the Proceedings of the Fifth International Conference on Real-Time Computing Systems and Applications (RTCSA'98).
- Orzessek M., Sommer P., and Victor L.**, 1998, ATM & MPEG-2 Integrating Digital Video into Broadband Networks, Prentice Hall, New Jersey.

KAYNAKLAR (devam)

Katcher K. Sathaye S. and Strosnider K., 1995, Fixed Priority Scheduling with Limited Priority Levels, IEEE Transactions on Computing.

Katcher K., Strosnider J. K., 1993, Dynamic Scheduling vs Fixed Priority Scheduling, IEEE Transactions on Software Engineering.

Keeton K. nad Katz R., 1995, Evaluating Video Layout Strategies for a High Performance Storage Server, ACM Multimedia Syatems, No:3, pp-43:52.

Koenen R., 1998, MPEG-4: Multimedia for our time, IEEE Spectrum.

Kunii T., Shinagawa Y., Paul R., Kahn M. F. and Khkhar A., 1995, Issues in Storage and Retrieval of Multimedia Data, ACM Multimedia Syatems, Vol:3, p:298-304.

Leslie I. M. ,McAuley D., and Mullender S.J., 1992, Pegasus – Operating System Support for Distributed Multimedia Systems, Pegasus Paper 92-2, University of Twente Faculty of Computer Science Memoranda Informatica.

Lewis B. And Berg D., 1998, Multithreaded Programming with Pthreads, Sun Microsystems Press, California.

Li X., Paul S., Pancha P and Ammar M., “Layered Video Multicast with Retransmission: Evaluation of Error Recovery Schemes”, Proceedings of NOSSDAV’97, St Louis, Missouri, May 1997.

KAYNAKLAR (devam)

ISO/IEC 13818-2, 1998, MPEG-2 Standard Documentation, Generic Coding of Moving Pictures and Associated Audio Information: Video.

Johnson N., 1996, Web Developer's guide to multimedia and video, Coriolis Group Books, Scottsdale, USA.

Jensen E. D., 1996, A Real-Time Manifesto, http://www.realtime.rtmanifesto/rtmani_0.html

Kantarci A., and Tunali T., 1999, Transmission of Multimedia Data Over IP, BAS'99 Computer Networks Symposium, Istanbul, Turkey.

Kantarci A., Özkasap Ö., and Tunali T., 1996, Dağıtık Çokluortam Ağları, BAS'96 Symposium, Computer Networks Symposium, Istanbul, Turkey.

Kantarci A., and Tunali T., 1997, Çokluortam Ağlarında Kaynak Yönetimi, Bilişim 97, İstanbul.

Kantarci A and Tunali T., 1998a, A New Scheduling Policy for Multimedia Applications, BAS'98, Computer Networks Symposium, Izmir.

Kantarci A. and Tunali T., 1998b, Priority Assignment Policies for Multimedia Tasks in General Purpose Operating Systems, ISCIS 98, Antalya.

KAYNAKLAR (devam)

- E. Emir, S. McCanne and R. Kotz**, 1998, An Active Service Framework and its Application to Real-Time Multimedia Transcoding", Proceedings of ACM SIGCOMM'98. Canada.
- Fisher S., and Keller R.**, 1995, Quality of Service Mapping in Distributed Systems, IEEE.
- Gallmeister B. O.**, 1995, Programming for the real world: POSIX4, O'Reilley & Associates, USA.
- Govindan K.**, 1992, Scheduling and IPC Mechanisms for Continuous Media, PhD. Thesis at University of North Carolina.
- Hodges M. E. and Sasnett R. M.**, Multimedia Computing, Addison Wesley, 1993.
- Hoffman D., Fernando G., Goyal V. and Civanlar M. R.**, 1998, RTP Payload Format for MPEG1/MPEG2 Video, RFC 2043.
- Herry M., Hengartner U., Steenkisle P., and Gross T.**, 1999, MPEG System Streams in Best-Effort Networks., PV1999 10th International Packet Video Workshop, New York.
- Hess C. K.**, 1998, Media Streaming Protocol: An adaptive protocol for the delivery of audio and video over the Internet, PhD. Thesis at University of Illinois at Urbana-Champaign.

KAYNAKLAR (devam)

Chang S.F., Eleftheriadis D., Anastassiou A., Jacobs S. and Kalva H. And Zamana J.,1997, Columbia's VoD and Multimedia Research Testbed with Heterogenous Network Support, Journal on Multimedia Tools & Applications.

Chatterjee S. and Strosnider J., 1995,A Generalized Admission Control Strategy for Heterogenous, Distributed Multimedia Sysyems, Proceedings of ACM Multimedia 95, San Francisco.

Chen Z., Tan S. M., Campbell R., and Li Y., 1995, Real-time Video and Audio in the World Wide Web , 4th International World Wide Web Conference.

Chuenlei L., 1998, Multimedia over IP: RSVP ,RTP ,RTCP, RTSP, Handbook of Communication Technologies: The Next Decade: CRC Press.

Campbell A., 1995, A QoS Architecture, PhD. Thesis at Computing Department, Lancaster University.

Campbell A., Coulson G., Garcia F., and Hutchison D., 1992, Continuous Media Transportand Orchestration Service, Proc. ACM SIGCOMM'92.

Colouris, 1994, Distributed System Concepts and Design, 2nd Edition, Addison Wesley.

KAYNAKLAR DİZİNİ

- Adelberg B., Moline H. G. and Kao B., 1994,** Emulating Soft RealTime Scheduling Using Traditional Operating System Schedulers, Technical Report, Stanford University.
- Blair G. S, Campbell A., Coulson G., Garcia F., Hutchison D., Scott A., and Shepherd W. D., 1993,** A Network Interface Unit to Support Continuous Media, IEEE Journal of Selected Areas in Communications, Special Issue on Network Interfaces.
- Buford J. F.K., 1994,** Multimedia Systems, ACM Press Books, SIGGRAPH Series, Addison Wesley.
- Burkow T., 1994,** Operating System Support for Distributed Multimedia Applications: A Survey of Current Research, Pegasus Paper:94-08, University of Twente Faculty of Computer Science Memoranda Informatica.
- Busse I., Deffner B., and Schulzrinne H., 1995,** Dynamic QoS Control of Multimedia Applications based on RTP, Proceedings of the First International Workshop on High Speed Networks and Open Distributed Platforms, St. Petersburg, Russia.
- Cen S., Pu C., and Staehli R., 1995,** A Distributed Real-time MPEG Video Audio Player, 5th International Workshop on Network and Operating System Support of Digital Audio and Video (NOSSDAV'95).

2. Sistemin PC'ler üzerinde yaygınlaştırılması:

Lucent Technologies'e ait RTP kütüphanesi, Solaris ve NT işletim sistemleri üzerinde çalışabilmektedir. İleride alıcının NT versiyonunun yazılmasıyla daha fazla sayıda kullanıcıya hitap etmesi sağlanacaktır.

3. Ses verilerinin işleme alınması:

Şu an XIL kütüphanesi sadece video verilerini destekleyebildiği için ses verileri üzerinde durulmamıştır. Sistem PC'lere taşınınca, ses verilerini destekleyen bir kütüphanenin bulunması durumunda, ses verileri de sisteme katılacaktır. Ses ve video görüntülerinin eş zamanlanması da bu kapsamda yerine getirilecek önemli bir konudur.

4. Sistemin WWW ortamına gömülmesi:

Sistemin WWW ortamına gömülmesi de ileride yapılması planlanan işler arasındadır. Bunun için JAVA ortamının çokluortam verilerini destekleme olanakları üzerinde araştırma yapılacaktır.

Tüm bunlara ek olarak, üniversite hastahanemizin odyovizyel merkezinde analog olarak kayıtlı bulunan ameliyat videolarının sayısallaştırılarak geliştirdiğimiz yazılım aracılığı ile tıp öğrencilerinin eğitimine sunulması da planlanan hedefler arasındadır.

Geliştirilen sistem, kampüs yerel ağı üzerinde denenerek sistem kapasitesi, kullanılan bant genişliği yönetimi, servis kalitesi yönetimi, tampon saha yönetimi ve akış kontrol mekanizmasına ilişkin başarımlar sonuçları alınmıştır. Alınan sonuçlar, geliştirilen uygulamanın kampüs ağları üzerinde başarılı bir şekilde kullanılabileceğini göstermektedir.

Geliştirilen sistem gerçek zamanlı veri işleme, iletişim, gösterim ve kontrol teknolojilerine ait en son ilerlemeleri başarılı bir şekilde bir araya getirdiğinden ayrı bir öneme ve değere sahiptir.

Sistemin daha da iyileştirilmesi için çalışmanın ileri safhasında yapılması planlanan işleri şöyle sıralayabiliriz:

1. MPEG-4 ve MPEG-7 standartlarına geçiş:

Yeterli donanımın sağlanması durumunda MPEG-4 ve MPEG-7 standartlarına geçiş yapılması planlanmaktadır. MPEG-4 standardı Internet uygulamalarını hedef alan nesne tabanlı ölçeklenebilir bir standarttır.

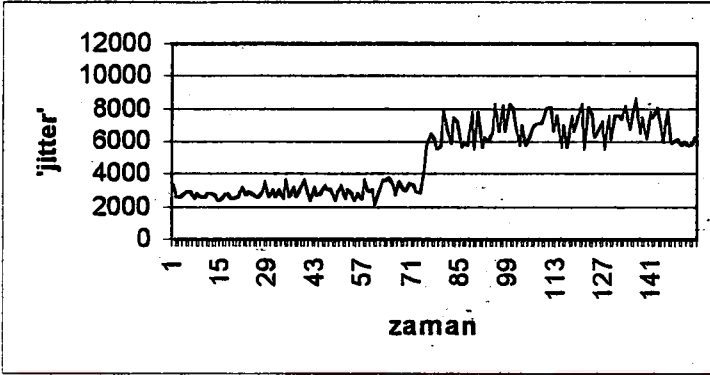
İndeksleme amaçlı MPEG-7 standartının kullanılması ileri ve geri alma gibi VCR işlemlerini kolaylaştıracaktır. Şu anki sistem üzerinde VCR fonksiyonları yerine getirilmemektedir. Ancak, video istatistiklerini tutan dosyada her bir çerçevenin başlangıç adresi bilgisi bulunmaktadır. Bu bilgiler kullanılarak istenen çerçevenin başlangıç adresine gidilebileceği için VCR fonksiyonları kolayca gerçekleştirilebilir. MPEG-7 standartının kullanılmaya başlanmasının bu işlemleri kolaylaştıracağı öngörülmektedir.

5. SONUÇ VE TARTIŞMA

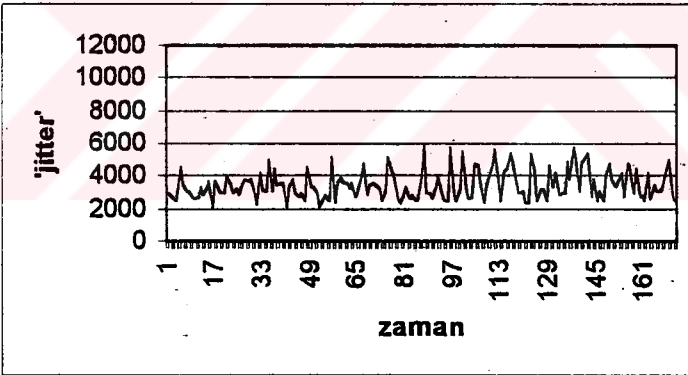
Bu tez kapsamında, saklanmış MPEG-1 videolarını Internet Protokolü üzerinden ileten bir uygulama geliştirilmiştir. Sistemin önemli bir özelliği sadece yazılım tabanlı bir sistem oluşudur. Sistem geliştirimi sırasında geleneksel işletim sistemleri üzerinde uygulamalara gerçek zaman desteği sağlama amacıyla kullanılan POSIX iş parçacıklı programlama ve POSIX 4 gerçek zamanlı programlama kütüphanelerinden yararlanılmıştır. Çok iş parçacıklı programlama teknikleri, servisçi mimarisini paralel bir yapıya büründürerek yazılım geliştirme işlemini kolaylaştırmış ve servisçi yazılımının etkinliğini arttırmıştır. İstemci yazılımının sahip olduğu boruhattı yapısı da çok iş parçacıklı programlama teknikleri ile gerçekleştirilmiş ve ortaya çıkan paralellik istemcide kullanıcıya sunulan veri miktarının mümkün olabilecek en yüksek düzeye çıkarılmasını sağlamıştır.

Video verilerinin iletimi için Lucent Technologies tarafından oluşturulmuş RTP kütüphanesi kullanılmıştır. MPEG-1 verilerinin RTP paketlerine dönüştürülmesi tarafımızca gerçekleştirilmiştir. Bant genişliği yönetimi için `INTERVAL_FOR_EACH_GOP` ve `INTERVAL_FOR_WHOLE_VIDEO` isimli yaklaşımlar geliştirilmiş ve birbirleriyle karşılaştırılmıştır. İstemcide bulunan tampon sahanın etkin bir şekilde kullanılarak gösterim sırasında sürekliliğin sağlanması amacıyla bir akış kontrol mekanizması geliştirilmiştir.

Sistem, ağın durumu hakkında kayıp oranı ve gelişler arası zamanlar arasındaki kayma gibi RTCP paketlerinde bulunan istatistikleri kullanarak ağdaki mevcut duruma uyum sağlayan dinamik bir sistemdir.

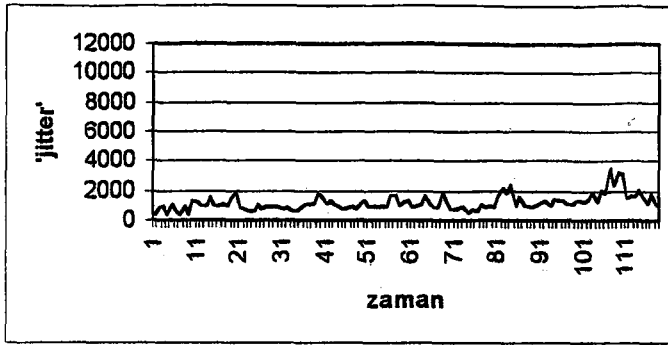


(d)

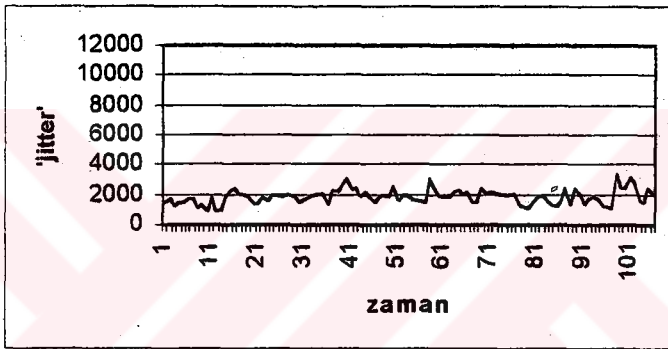


(e)

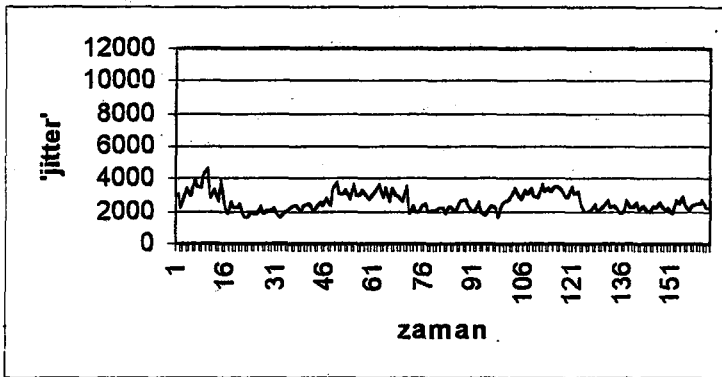
Şekil 4.13. 100, 200, 500, 1000 ve 1500 Kbps'lik sıkıştırma hızları için kayma değerleri



(a)



(b)



(c)

raporundaki kayma değeri sadece o pakete ait olduğundan sadece tek bir rapora bakarak karar vermek yanlıştır. Bu nedenle, çok sayıda paketin analiz edilerek daha sağlam bir karara varılması gerekir.

Şekil 4.13'de sırasıyla 100, 200, 500, 1000 ve 1500 Kbitlik sıkıştırma hızları için gözlenen kayma değerleri gösterilmiştir. Sıkıştırma hızı arttıkça ortalama kayma değerlerinin de arttığı gözlenmiştir. Şekil 4.10.d 'da 1000 Kbitlik sıkıştırma hızında iletimin başlangıcından bir süre sonra kayma değerinin aniden arttığı görülmektedir. Kayma değerinin yükselmesinden çok kısa bir süre sonra da paket kaybı istatistiklerinin 0'dan farklı olduğu gözlenmiştir. Bu örnekle birlikte, kaymadaki ani artışın gelecek paket kayıplarının bir habercisi olabileceği doğrulanmaktadır.

belirlenmiştir. Her çerçeveye ait saat değeri birbirinden farklıdır. Bir çerçeveye ait paketlerin tümünde zaman etiketleri aynıdır. t mikro saniye cinsinden sistem saatinin çalışmaya başladığından beri geçen süre olmak üzere, bir çerçeveye ait zaman etiketi (TS) şöyle hesaplanır.

$$TS = (t * 90000) / 1000000 = t * 9 / 100 \quad (4.1)$$

RTCP paketlerinde bu saat ünitesi cinsinden RTP paketlerinin gelişleri arasındaki kaymanın istatistiksel varyansını gösteren bir saha bulunur. Bu değer şu şekilde hesaplanır:

S_i i. Paketin RTP zaman etiketinin değerini, R_i alıcının bu paketi aldığı RTP zamanını gösterirse i ve j. Paketlerin ağa taşındığı süreler arası fark (4.2) ile hesaplanır.

$$D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \quad (4.2)$$

Gelişler arası kayma değeri, alınan her paket için öz yinelemeli olarak hesaplanır:

$$J_i = J_{i-1} + (D(i-1,i) - J_{i-1}) / 16, \quad J_0 = 0 \quad (4.3)$$

RTP taslağına göre, bu bağıntı birinci dereceden en iyi tahminlemeyi verir ve 1/16 parametresi en iyi gürültü azaltım oranını elde edilmesini sağlar (Schulzrinne, 1996).

Gelişler arası kayma kayıp oranından sonra ağdaki geçici sıkışıklıkların ikinci işaretidir. Toplam kayıp paket sayısı uzun vadeli sıkışıklıkların göstergesiyken, kayma değerleri yakın gelecekte paket kayıplarına neden olabilecek sıkışıklığın göstergesi olabilir. Bir RTCP

şekilde mevcut işlemcileri destekleyebilmektedir. İleride PC'lerin de sisteme dahil edilmesiyle daha çok istemcinin aynı anda çalışması sağlanacaktır. Bu konfigürasyonda kayıp istatistiklerinin daha yüksek değerlere sahip olması beklenmektedir.

Çizelge 4.5 Kayıp istatistikler

Sıkıştırma Bit hızı	Toplam çerçeve sayısı	Toplam paket sayısı	Kayıp paket sayısı	Kayıp çerçeve sayısı
100 Kbps	17835	20154	0	0
200 Kbps	17835	25114	21	3
500 Kbps	17835	38415	82	25
1000 Kbps	17835	80839	156	48
1500 Kbps	17835	138703	278	79

4.7.2 Kayma ('jitter')

RTP başlıklarına eş zamanlama ve kayma ('jitter') hesaplamaları için uygun çözünürlüğe sahip bir saatten elde edilen zaman etiketi bilgileri yerleştirilir. Bu saatin frekansı taşınan veri türüne bağlı olarak değişir. İlgili RTP taslak dökümanında değişik veri türlerine ait frekans değerleri listelenmiştir. RTP paketlerinin örnekleme zamanı sistem saati yerine bu saat cinsinden belirlenir. Örneğin, bir sabit hızlı ses uygulamasında zaman etiketini gösteren saat her örnekleme işlemi ile bir artırılır. Bir ses aygıtından her 160 örneklemeden sonra bir blok halinde veriler okunuyorsa, zaman etiketi her bloğun alınmasından sonra 160 birim artırılır. MPEG verilerinin saat frekansı 90 KHz olarak belirlenmiştir. Her çerçeveye ait saat değeri birbirinden farklıdır. Bir çerçeveye ait paketlerin tümünde zaman etiketleri aynıdır. t mikro saniye

verir. Bu iki rapor içinde yer alan paket sıra numarası değerleri arasındaki fark, bu aralıkta beklenen paket sayısına karşılık gelir. Bu iki değer arasındaki oran, bu aralıktaki kayıp paketlerin oranını verir. Bu iki paket ardışıkrsa, elde edilen değer, son RTCP paketinde bulunan kayıp oranı sahasındaki değere eşittir. Kayıp paketlerin sayısı RTCP paketlerinde yeralan ve saniye cinsinden ifade edilen NTP zaman etiketlerine bölünecek olursa saniyede kaybolan paketlerin oranı elde edilir. Alınan paket sayısı, beklenen paket sayısından kayıp paket sayısının çıkarılması ile elde edilir. Beklenen paket sayısı, kayıp tahminlerinin istatistiksel geçerliliği hakkında yorum yapmak için de kullanılır. Örneğin, 5 paket içinden 1 paketin kaybolması, 1000 paket içinden 200 paketin kaybolmasından daha az önemlidir.

Toplam kayıp paket sayısı, uzun vadeli kayıpların habercisidir. Kayıp oranı istatistiği ise tek bir rapora ait olup kısa vadeli kayıpların işaretidir.

Çizelge 4.5'te değişik sıkıştırma bit hızları için ağda kaybolan paket ve çerçeve sayısı istatistikleri verilmektedir. Bu sonuçlar, CLI-3 makinasında istemci yazılımının 30 çerçeve/sn. çerçeve hızıyla oynatılmasıyla alınmış sonuçlardır. CLI-3 makinası ve servisçi makinası arasında üç Ethernet anahtarı bulunmaktadır. Tüm bu verilere göre paket kaybının önemsiz olduğu ve sistemin kampüs-yerel ağı üzerinde başarıyla kullanılabileceği söylenebilir. İstemci programı, servisçi ile aynı binada bulunan CLI-1 ve CLI-2 makinalarında oynatıldığında ise kayıp oranı hemen hemen hiç yok gibidir.

Çizelge 4.2 incelendiğinde servisçinin hizmet vermesi beklenen en yüksek istemci sayısının oldukça yüksek olduğu görülmektedir. Şu an sistemimiz sınırlı sayıda istemciden oluştuğu için servisçi sorunsuz bir

6. Servisçiden gelen son RTCP raporunun alındığı an
7. Servisçiden gönderilen son RTCP raporunun alınmasından itibaren geçen süre.

İlk saha alıcının diğerlerinden ayırdedilmesi işlevini görür. Onu izleyen dört saha doğrudan iletimin kalitesi ile ilgili olduğundan alt bölümlerde bu sahalar hakkında bilgi verilecektir. Son iki saha alıcı ve gönderici arasındaki gidiş-dönüş süresinin hesaplanması için kullanılır. Sistemimizde gidiş-dönüş zamanı istatistikleri kullanılmadığı için bu iki saha üzerinde açıklama yapmaya gerek duyulmamıştır.

4.7.1 Kayıp İstatistikleri

RTCP raporlarında, kayıp paketlere ilişkin iki istatistik tutulmaktadır. Bunlardan biri, son gönderici ya da alıcı paketinden sonra kaybolan paketlerin yüzdesi, diğeri de iletimin başladığı andan itibaren toplam kayıp paket sayısını gösteren istatistiktir.

Kayıp paket oranı istatistiği kayıp paket sayısının beklenen paket sayısına bölümü ile elde edilir. Toplam kayıp paket sayısı istatistiği ise beklenen paket sayısından alınan paket sayısının çıkarımı ile elde edilir. Beklenen paket sayısı, RTCP raporlarında yer alan alınan en son ve ilk paket numaraları arasındaki farktan yararlanarak hesaplanır ve her RTCP paketine yerleştirilir.

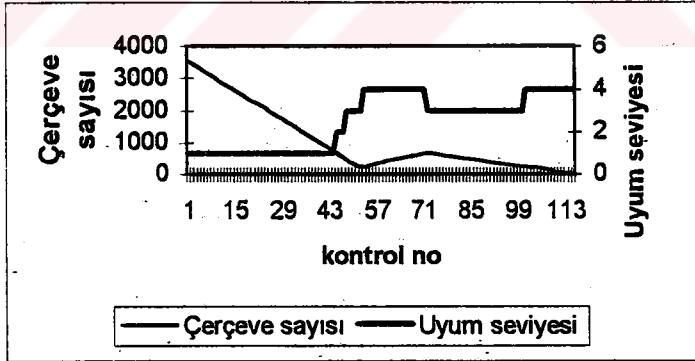
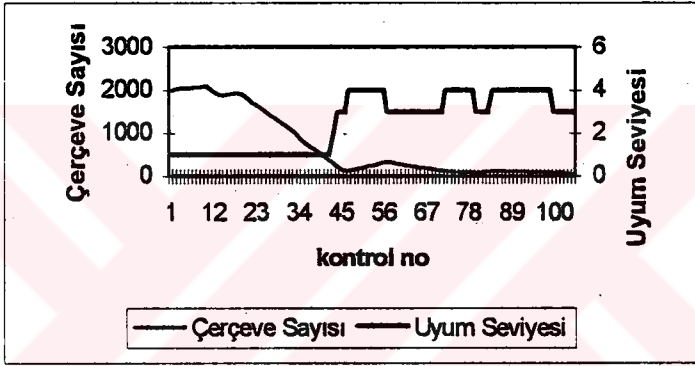
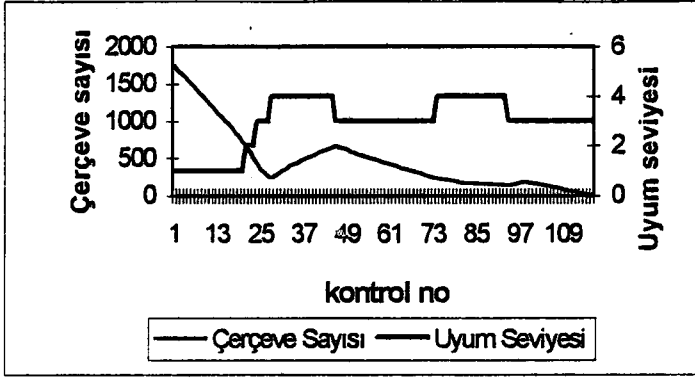
Farklı RTCP paketleri içindeki bilgiler kullanılarak farklı analizler yapılabilir. Örneğin, iki RTCP raporunda yer alan toplam kayıp paket sayısı değerleri arasındaki fark bu aralıkta kaybolan paketlerin sayısını

4.7 RTCP İstatistikleri

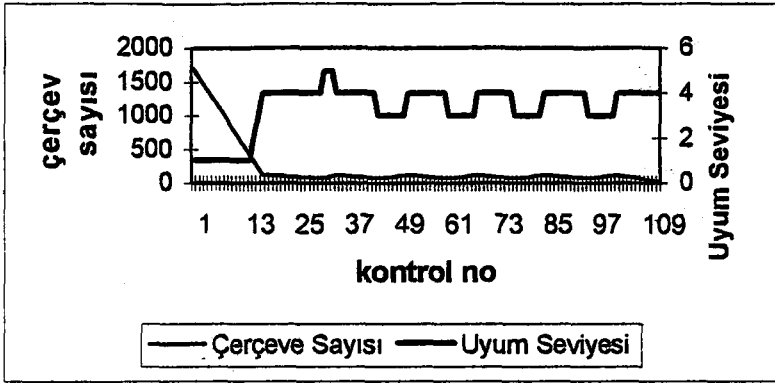
RTCP paketlerinin ana görevi, veri dağıtımının kalitesi hakkında geri besleme sağlamaktır. RTCP paketleri, diğer nakil protokollerinin sıkışıklık ve akış kontrolü fonksiyonlarını yerine getirdikleri için, RTP protokolünün tamamlayıcısıdır. RTCP paketleri temel olarak alıcı ve gönderici RTCP paketleri olmak üzere ikiye ayrılırlar. Sistemimizin 'unicast' yapısı nedeniyle sadece alıcılardan gönderilen RTCP paketleri servisçide işleme sokularak iletişim sırasında yaşanan hatalar belirlenir ve servisçinin uygun davranışa geçmesi sağlanır. Servisçide alıcılara RTCP raporları göndermektedir. Bu raporlar, sadece alıcıların RTP modülünde hazırlanan RTCP paketlerinde bulunan bazı sahalanın değerlerinin belirlenmesinde kullanılır. Uygulamanın diğer modülleri, servisçiden gelen RTCP raporlarını dikkate almazlar.

Alıcının gönderdiği RTCP paketleri şu sahalardan oluşur:

1. Alıcı numarası
2. Kayıp oranı
3. Toplam kayıp paket sayısı
4. Paket sıra numarası
5. Gelişler arası kayma

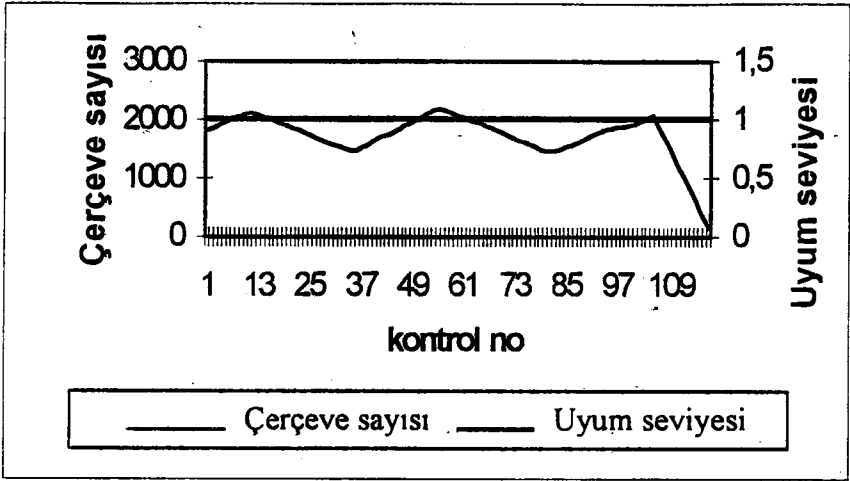


Şekil 4.12. 1000 Kbps için ön tamponlama süresinin 0.5, 1 ve 2 dak. olduğu durumlarda tampon saha doluluğu

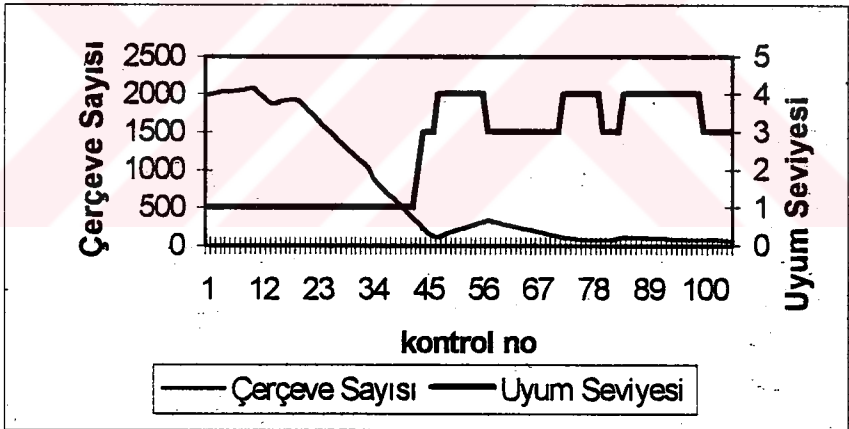


Şekil 4.11.e Sıkıştırma bit hızının 1500 Kbps olduğu durumda tampon sahada bulunan çerçeve sayısı

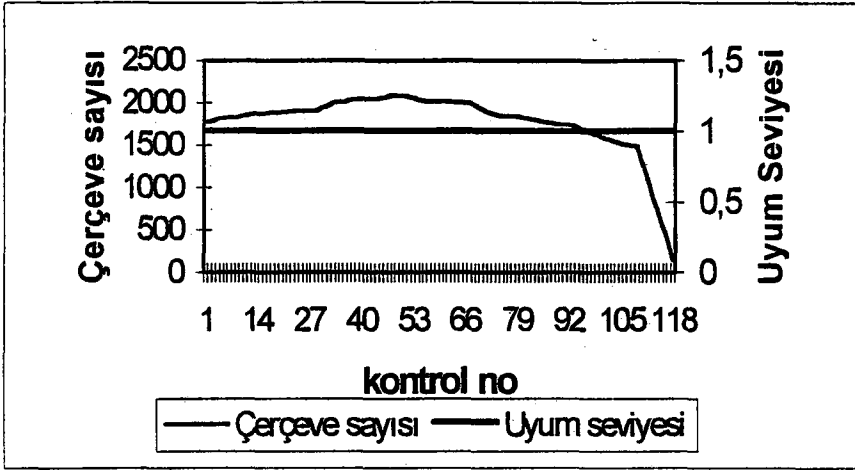
Ön tamponlama süresinin geçerli uyum seviyesi üzerindeki etkisi üzerine de deneyler yapılmıştır. Şekil 4.12'de 1000 Kbps için değişik ön tamponlama süreleri için gözlenen tampon saha doluluğu grafikleri verilmektedir. Şekil 4.12'den anlaşılabileceği gibi ön tamponlama süresi ne kadar yüksek olursa, alıcı istasyonda tampon saha doluluğunun boşalma bölgesine düşmesi için gereken süre de o kadar yüksek olmaktadır. Ön tamponlama süresi ve buna bağlı olarak tampon saha miktarı yükseldikçe, sunum başladığında geçerli olan uyum seviyesi daha uzun bir süre boyunca korunmaktadır.



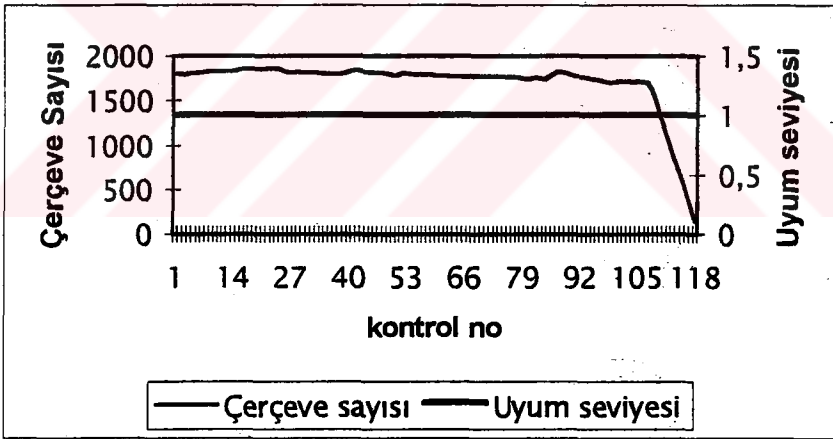
Şekil 4.11.c Sıkıştırma bit hızı 500 Kbps olduğunda tampon sahada bulunan çerçeve sayısı



Şekil 4.11.d Sıkıştırma bit hızının 1000 Kbps olduğu durumda tampon sahada bulunan çerçeve sayısı



Şekil 4.11.a Sıkıştırma bit hızının 100 Kbps olduğu durumda tampon sahada bulunan çerçeve sayısı



Şekil 4.11.b Sıkıştırma bit hızının 200 Kbps olduğu durumda tampon sahada bulunan çerçeve sayısı

dakika iken 1000 ve 1500 Kbpslik hızlarda bu süre 2.5 dakika ve 3.5 dakikaya yükselmektedir. Ayrıca, tüm videonun oynatımı için geçen süre de bu sıkıştırma hızlarında beklenen süreden yüksek olmaktadır. Yüksek bit hızlarında teorik ve gerçek paketler arası aralığın aynı olmasını sağlayacak çerçeve hızları kullanılarak bu beklemelerin ve tampon sahadaki düşüşlerin önüne geçilebilir.

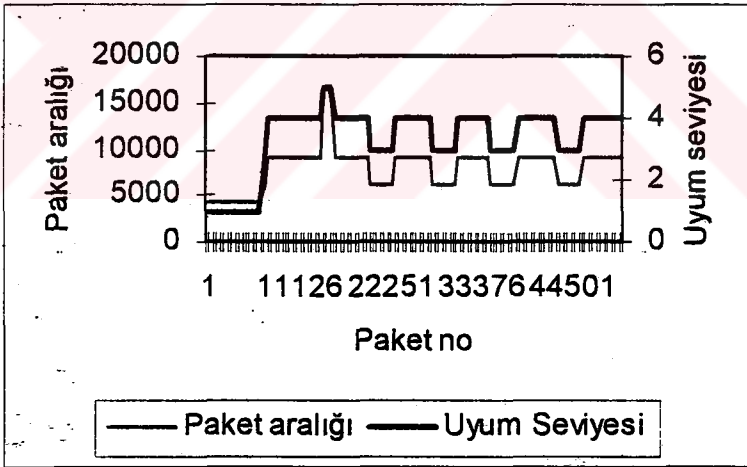
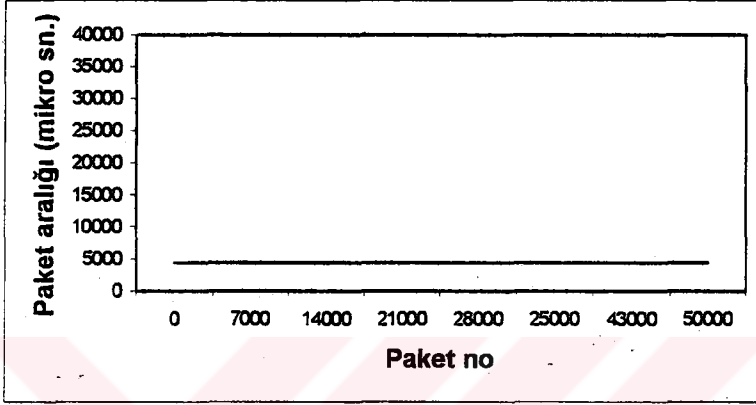
Sistemimizde tampon sahanın doluluk kontrolleri her 10 GOP'un gösteriminden sonra gerçekleşmektedir. Bu 30 çerçeve/sn hız için yaklaşık 5 saniyeye karşılık gelmektedir. Kontrollerin sıklığının düşmesi, tampon sahadaki boşalmaların belirlenmesini güçleştirebilir. Sıklığın azaltılması, boşalmaların daha erken farkına varılmasını sağlayabildiği gibi, kısa süreli doluluk miktarı değişikliklerinin önemli değişiklikler gibi algılanmasına ve gereksiz işlemlerin başlatılmasına neden olabilir. 1000 ve 1500 Kbpslik sıkıştırma hızlarında, tampon sahanın doluluk kontrolleri her 5 GOP'un gösteriminden sonra da denenmiştir. Gerçek paket aralığının teorik paket gönderim aralığından daha yüksek olması nedeniyle tampon sahadaki düşüşün önüne geçilememiş ve 10 GOP'luk aralığın kullanıldığı duruma benzer sonuçlar elde edilmiştir.

4.6 Tampon Saha Yönetimi

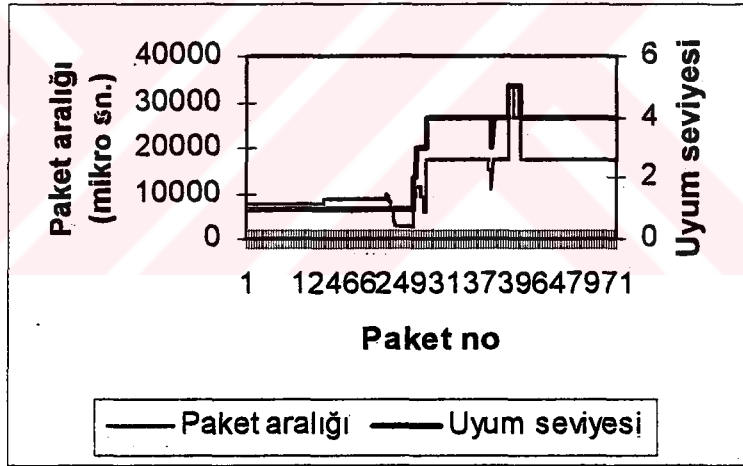
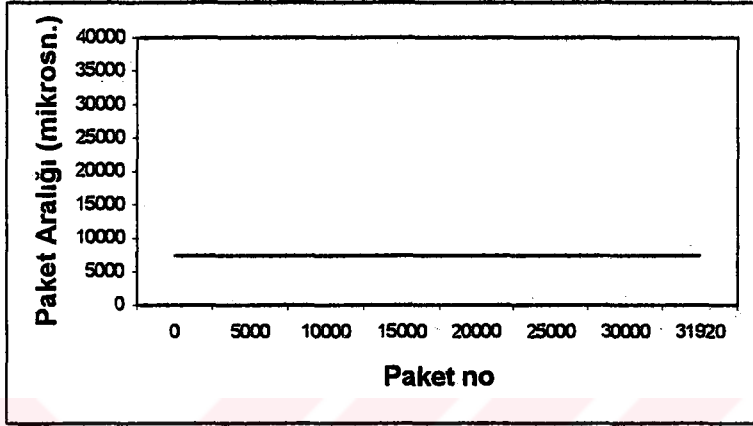
Şekil 4.11’de tampon saha yönetimi için geliştirdiğimiz metoda ait tampon saha doluluk grafikleri görülmektedir. Sıkıştırma bit hızının 100 ve 200 Kbps olduğu durumlarda, tampon sahada bulunan çerçeve sayısı kararlı bölgenin sınırları içinde kalmaktadır. Tampon sahanın doluluk miktarı sıkıştırma bit hızının 500 Kbps olduğu durumda dalgalanma göstermeye başlamıştır. Tampon saha doluluğu kararlı bölgenin altına düştüğü anda tampon saha yönetim mekanizması hemen tepki göstermiş ve tampon sahadaki çerçeve sayısını kararlı bölge sınırları içine yükseltmiştir.

Sıkıştırma bit hızının 1000 ve 1500 Kbps olduğu durumlarda tampon sahanın boşalma tehlikesi baş göstermiştir. Bu sıkıştırma hızlarında, çerçevelerin ağdan alınış hızının, oynatıcı iş parçacığının tampon sahadan çerçeve alma hızından düşük olması nedeniyle tampon saha hızla boşalmaktadır. 1000 ve 1500 Kbps’lik sıkıştırma bit hızlarında 30 çerçeve/sn. hızda paketler arası teorik gönderim aralığı yaklaşık 7 ve 4 milisn. olarak hesaplanmıştır. Çizelge 4.2 ve Çizelge 4.4 kullanılarak yapılan hesaba göre göre 1 paketin daha kısa sürede işlenmesi ve iletilmesi gerekmektedir. Dolayısıyla, gerçekte paketler ağa teorik olarak belirlenen aralıktan daha büyük aralıklarla gönderilmektedirler. Bu nedenle, istemci tarafında çerçevelerin ağdan alınış hızı oynatıcı iş parçacığının tampon sahadan alma hızından düşük olmakta, sonuç olarak da tampon saha hızla boşalmaktadır.

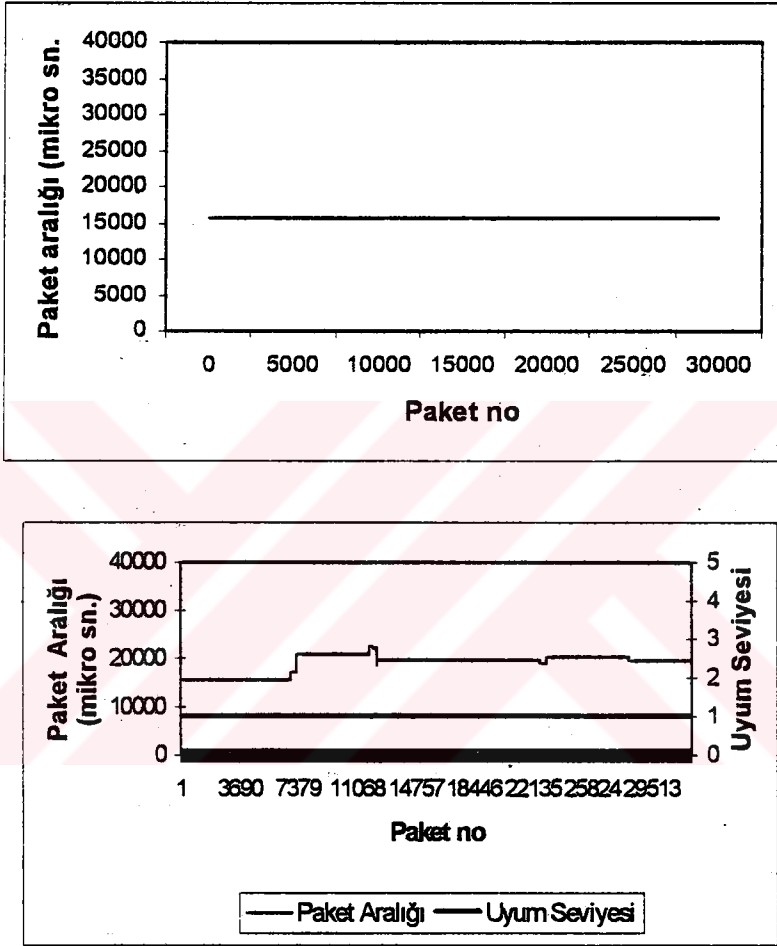
Aynı nedenden ötürü, 1000 ve 1500 Kbps’lik sıkıştırma hızlarında oynatıcı istasyonda ilk çerçevenin ekranda gösterilmesi için gereken bekleme süresi, beklenen 1 dakikalık süreden uzun olmaktadır. 100, 200 ve 500 Kbps’lik bit hızlarında ön tamponlama gecikmesi yaklaşık 1



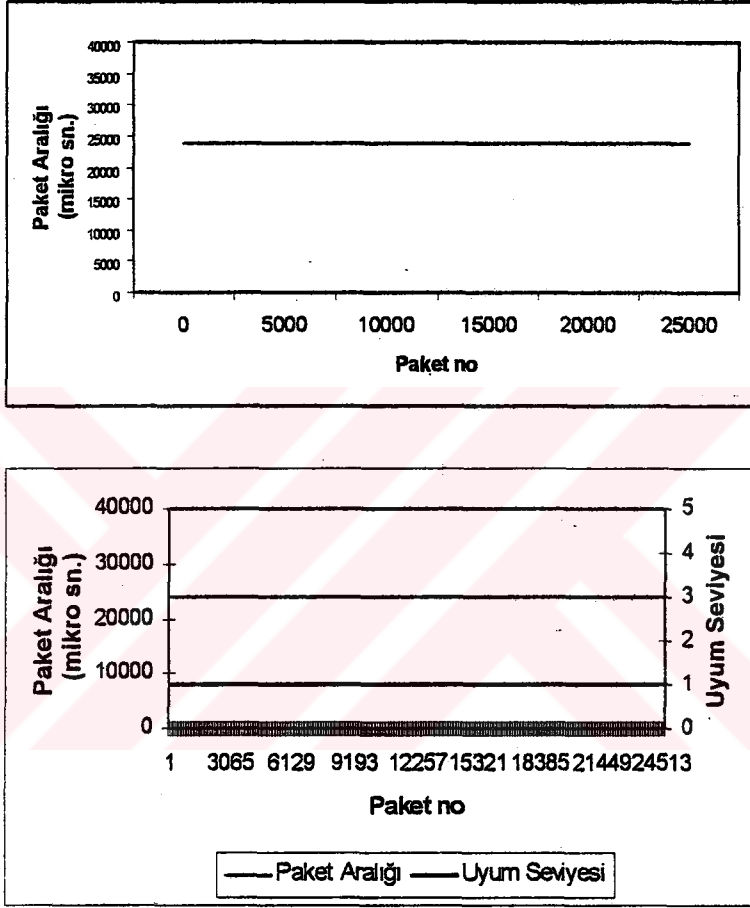
Şekil 4.10.e INTERVAL_FOR_WHOLE_VIDEO yöntemine göre 1500 Kbps için beklenen ve gözlenen paketler arası aralık



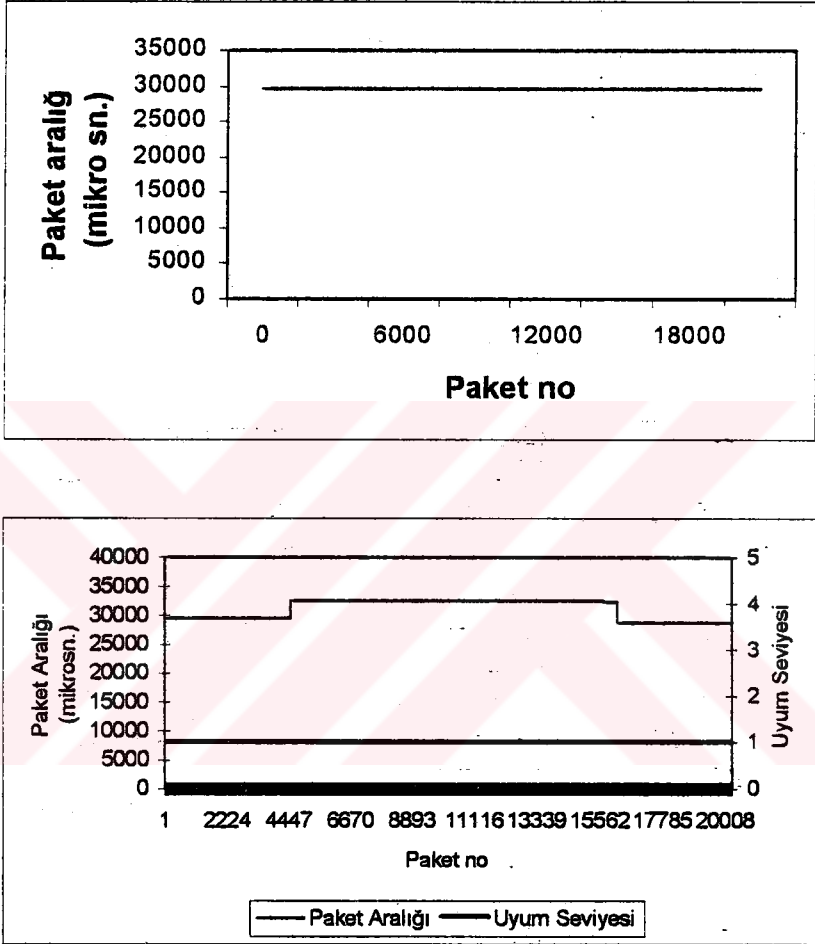
Şekil 4.10.d INTERVAL_FOR_WHOLE_VIDEO yöntemine göre 1000 Kbps için beklenen ve gözlenen paketler arası aralık



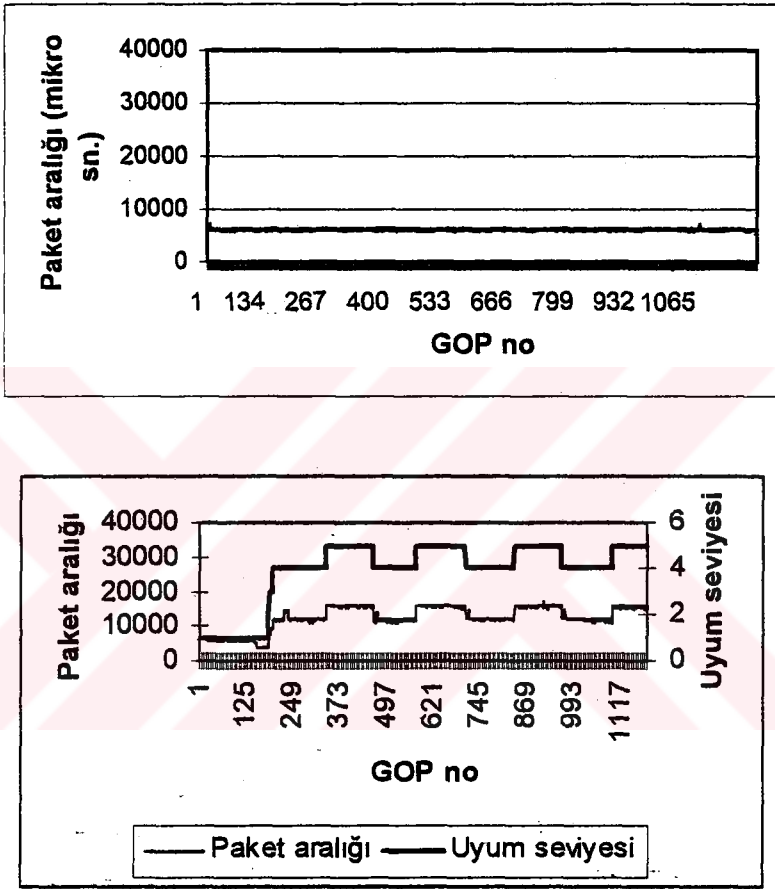
Şekil 4.10.c INTERVAL_FOR_WHOLE_VIDEO yöntemine göre 500 Kbps için beklenen ve gözlenen paketler arası aralık



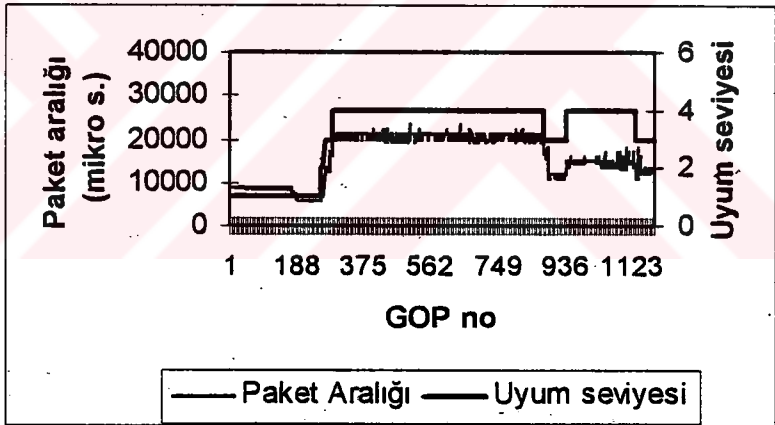
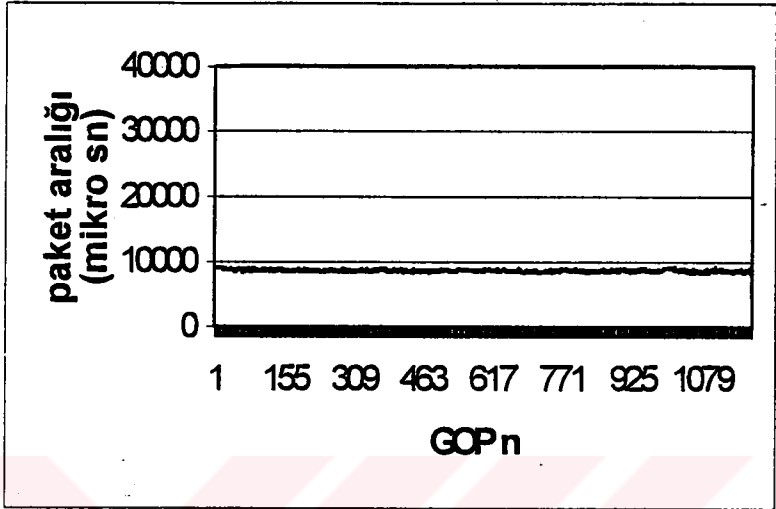
Şekil 4.10.b INTERVAL_FOR_WHOLE_VIDEO yöntemine göre 200 Kbps için beklenen ve gözlenen paketler arası aralık



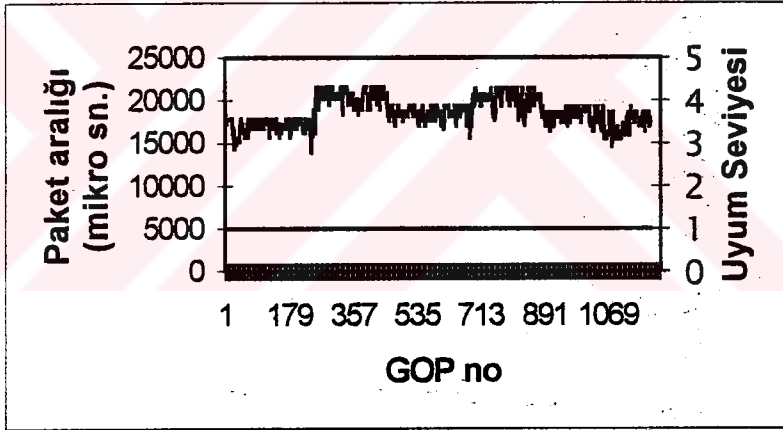
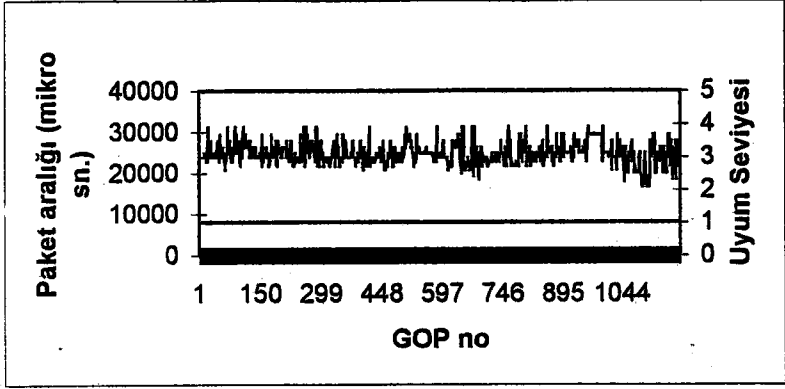
Şekil 4.10.a INTERVAL_FOR_WHOLE_VIDEO yöntemine göre 100 Kbps için beklenen ve gözlenen paketler arası aralık



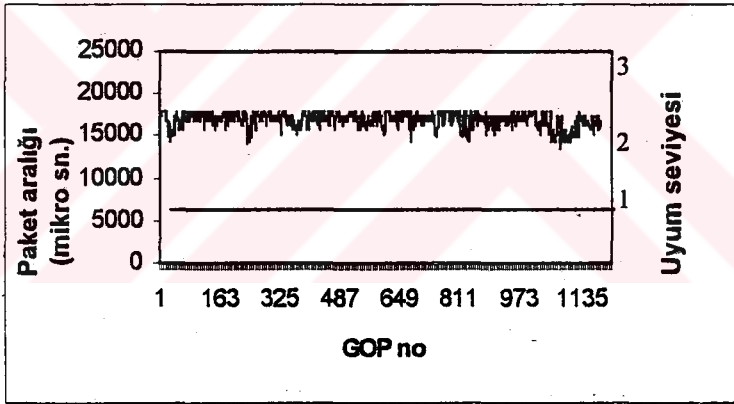
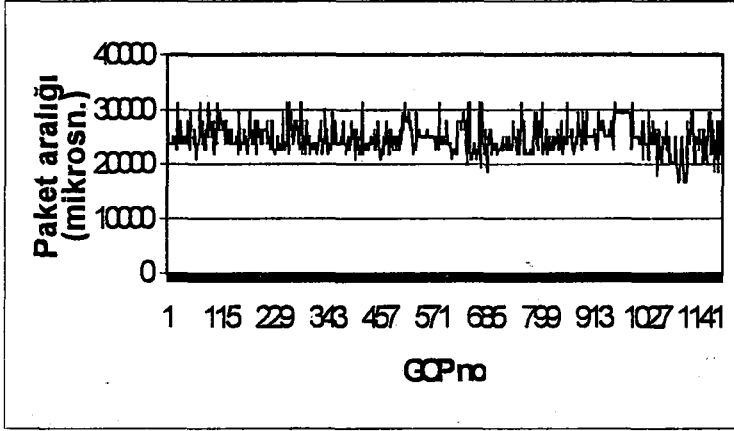
Şekil 4.9.e INTERVAL_FOR_EACH_GOP yöntemine göre 1500 Kbps için beklenen ve gözlenen paketler arası aralık



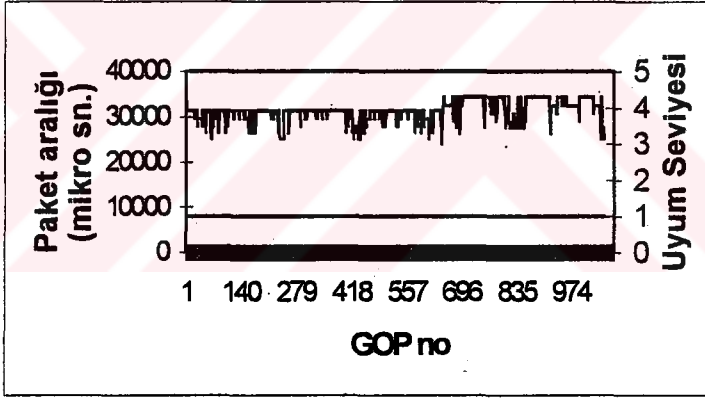
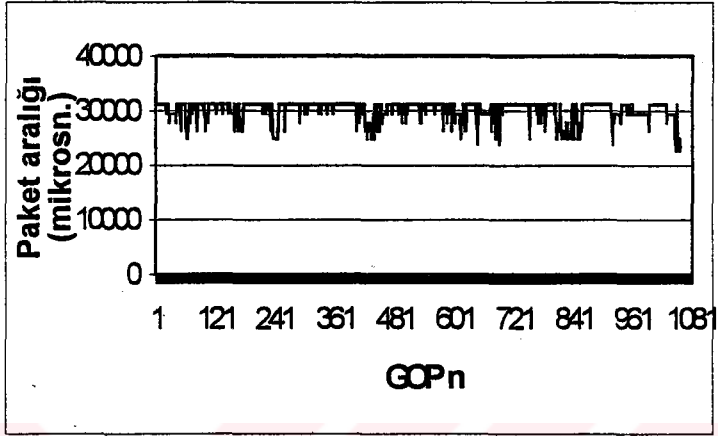
Şekil 4.9.d INTERVAL_FOR_EACH_GOP yöntemine göre 1000 Kbps için beklenen ve gözlenen paketler arası aralık



Şekil 4.9.c INTERVAL_FOR_EACH_GOP yöntemine göre 500 Kbps için beklenen ve özlenen paketler arası aralık



Şekil 4.9.b INTERVAL_FOR_EACH_GOP yöntemine göre 200 Kbps için beklenen ve gözlenen paketler arası aralık



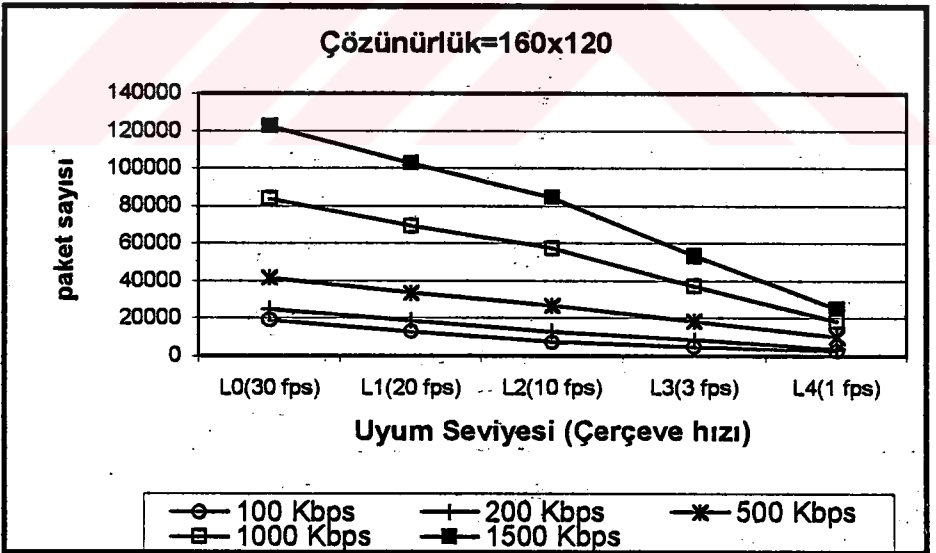
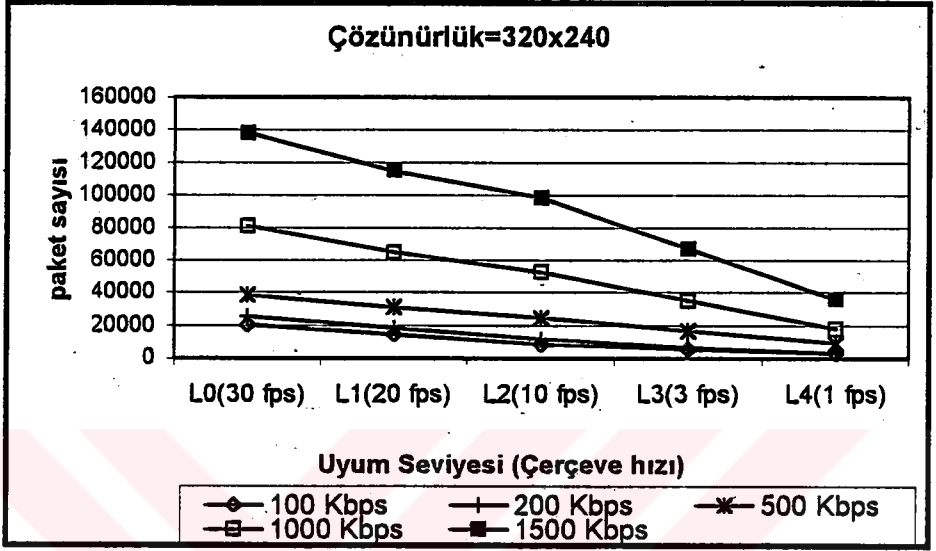
Şekil 4.9.a INTERVAL_FOR_EACH_GOP yöntemine göre 100 Kbps için beklenen ve gözlenen paketler arası aralık

Şekil 4.9 ve Şekil 4.10'da paketler arası aralıkları belirlemede kullanılan iki y nteme ilişkin grafikler sunulmaktadır. INTERVAL_FOR_EACH_GOP y ntemi i in hesaplanan paket aralıđı tahminlemeleri, her GOP i in GOP b y kl đ ne bađlı olarak farklı aralık belirlediđi i in oynamalar g stermektedir. INTERVAL_FOR_WHOLE_VIDEO y ntemi, bir videonun b l neceđi paket sayısı istatistiklerini kullanarak t m video i in tek bir aralık belirlediđinden tahmini g nderim aralıkları video boyunca sabittir. Ger ek g nderim aralıkları ise uygulanan tampon saha kontrol y ntemi ve akıř kontrol mod llerinin de iřleme katılması nedeniyle beklenenden farklıdır. INTERVAL_FOR_WHOLE_VIDEO y ntemi, paketler arası zamanı daha uzun bir s re sabit kıldıđı i in ađ  zerindeki kararlılıđı ve tahminlenebilirliđi arttırmaktadır. Bu y zden, bu y ntemin ilkinden daha bařarılı olduđunu s yleyebiliriz.

İletiřim sırasında Şekil 4.6 ve 4.7'da g r ld đ  gibi akıř kontrol mod l  gerektiđinde servis iden paket aralıđını deđiřtirmesini istemektedir. İki y ntemde de 100, 200, 500 Kbpslik sıkıřtırma hızlarında paket aralıkları uygun sabitlerle  arpılarak uyum seviyesi deđiřtirmeye gerek kalmadan deđiřtirilmektedir. İki y ntemde de 200 Kbps i in paket aralıđı, tampon saha doluluđu kararlı b lgede kaldıđı i in orijinal deđerleri korumaktadır. INTERVAL_FOR_EACH_GOP y nteminde 500 Kbps'de paket aralıđı tampon saha doluluđu miktarındaki dalgalanmaya bađlı olarak deđiřmektedir. INTERVAL_FOR_WHOLE_VIDEO y nteminde ise iletiřim sırasında ayarlamalarla paket aralıklarının kararlı bir deđere ulařtıđı g r lmektedir. 1000 ve 1500 Kbps'de tampon saha doluluđunun sık sık hızlı bir řekilde d řmesi nedeniyle, iki y ntemde de uyum seviyesinin deđiřtirilmesi s z konusu olmuřtur.

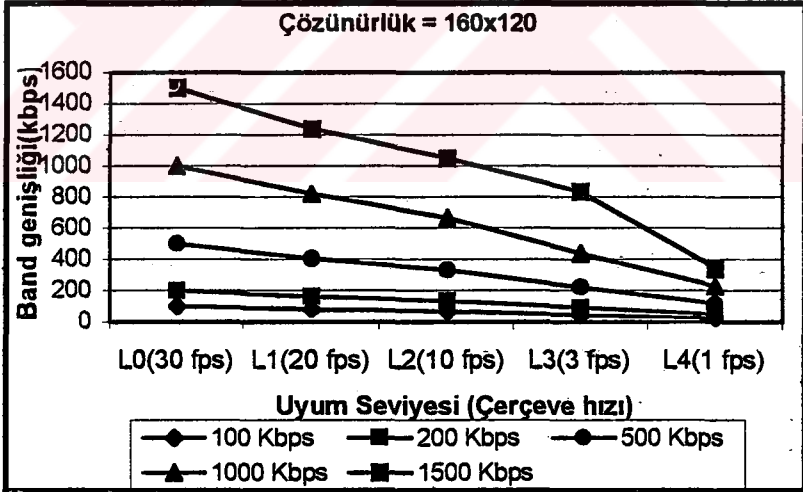
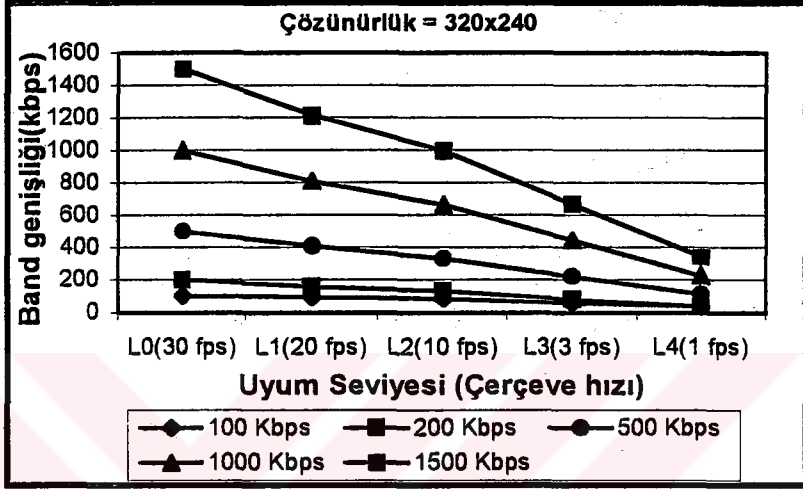
Çizelge 4.4 Farklı uyum seviyeleri için hesaplanan tahmini paket sayıları ve gönderilen gerçek paket sayıları

	Uyum Seviyesi	Tahmini Paket Sayısı	Gerçek Paket Sayısı
100 Kbps	1	20154	20249
	2	14209	14305
	3	8539	8360
	4	6211	5502
	5	3833	2715
200 Kbps	1	24345	25114
	2	17952	18177
	3	9656	11606
	4	7201	6799
	5	4281	2934
500 Kbps	1	35495	38415
	2	28660	31198
	3	14308	24332
	4	11467	16723
	5	8501	9186
1000 Kbps	1	69161	80838
	2	58014	64610
	3	29689	52402
	4	23121	35189
	5	16802	17969
1500 Kbps	1	97333	138015
	2	82559	114417
	3	41676	98270
	4	32518	66942
		23919	35550



Şekil 4.8 Farklı uyum seviyelerinde ve sıkıştırma bit hızlarında kullanılan paket sayısı

yüzden, gönderilen gerçek paket sayısı tahmini paket sayısından yüksek olmaktadır. Kullanılan sıkıştırma bit hızının bir resmin bir pakete sığmasına neden olacak şekilde küçük olması durumunda da gerçek paket sayısı tahminlenen değerden çok yüksek olacaktır.



Şekil 4.7 Farklı uyum seviyelerinde ve sıkıştırma bit hızlarında ihtiyaç duyulan bant genişliği miktarı

4.5 Bant Genişliği Gereksinimi

Şekil 4.7’de farklı uyum seviyeleri, dolayısıyla farklı çerçeve hızları için ihtiyaç duyulan bant genişliği miktarları verilmiştir. Bu şekilden şu sonuçlar çıkarılabilir:

1. Bant genişliği gereksinimi çözünürlüğe bağlı değildir. Bant genişliği gereksinimi sıkıştırma sırasında kullanılan bit hızı ve birim zamanda gönderilen çerçeve sayısının bir fonksiyonudur.
2. Video kalitesi ve gereksinim duyulan bant genişliği arasında doğru bir orantı ortaya çıkmaktadır. Video kalitesi arttıkça, başka bir ifadeyle sıkıştırma için kullanılan bit hızı arttıkça birim zamanda gönderilmesi gereken veri miktarı artar.
3. Uyum seviyesi arttıkça birim zamanda gönderilen çerçeve sayısı azalmakta, dolayısıyla bant genişliği gereksinimi de azalmaktadır.

Şekil 4.8’de ise gönderilen paket sayısına ilişkin grafikler sunulmaktadır. Bu grafikler de bizi yukarıda benzer sonuçlara götürür.

Çizelge 4.4’te teorik olarak hesaplanan paket sayısı tahmini ile gerçekte gönderilen paket sayısı listelenmektedir. Düşük bit hızlarında hesaplanan değer ve gerçek paket sayısı arasındaki fark düşükken, bit hızı arttıkça iki değer arasındaki fark da artmaktadır. Bu durum, RTP protokolünün MPEG videolarını paketlere dönüştürme kurallarından ileri gelmektedir. Yüksek bit hızlarında kesitler tek bir pakete sığmadığı için paketlere ayrılırlar. Bir sonraki kesitin bir paketin başına yerleştirilmesi gerektiğinden, o anki kesidin yerleştirildiği son paket yarım kalır. Bu

kaynaklanmaktadır. Bu nedenle her çerçeve türü için harcanan süre birbirine yakındır.

Çizelge 4.3. Milisaniye cinsinden XIL kütüphanesinde harcanan süre

Bit hızı/çözünürlük		160x120	320x240
100 Kbps	I	8,3	24,5
	B	8,3	24,6
	P	7,2	18,8
200 Kbps	I	8,9	25,6
	B	8,9	25,6
	P	8,3	19,5
500 Kbps	I	9,4	27,7
	B	9,4	27,7
	P	10,1	23,9
1000 Kbps	I	10,8	27,1
	B	10,7	27
	P	12,5	25,9
1500 Kbps	I	12,2	28
	B	12,2	28
	P	14,1	27,1