



T.C.  
EGE ÜNİVERSİTESİ  
Fen Bilimleri Enstitüsü

# SANALLAŞTIRMA VE YTA TEKNOLOJİLERİNE DAYALI VIDEO İLETİMİ

Doktora Tezi

Mustafa TÜKER

Uluslararası Bilgisayar Anabilim Dalı

İzmir

2023







T.C.  
EGE ÜNİVERSİTESİ  
Fen Bilimleri Enstitüsü

**SANALLAŞTIRMA VE YTA TEKNOLOJİLERİNE  
DAYALI VIDEO İLETİMİ**

Mustafa TÜKER

Danışman: Doç. Dr. Müge SAYIT

Uluslararası Bilgisayar Anabilim Dalı  
Bilgi Teknolojileri Doktora Programı

İzmir  
2023



Mustafa TKER tarafından doktora tezi olarak sunulan ‘‘Sanallařtırma ve YTA Teknolojilerine Dayalı Video İletimi’’ bařlıklı bu alıřma E Lisansst Eđitim ve đretim Ynetmeliđi ile E Fen Bilimleri Enstits Eđitim ve đretim Ynergesi’ nin ilgili hkmleri uyarınca tarafımızdan deđerlendirilerek savunmaya deđer bulunmuř ve 18.10.2023 tarihinde yapılan tez savunma sınavında aday oybirliđi/oyokluđu ile bařarılı bulunmuřtur.

**Jri yeleri:**

**İmza**

<b>Jri Bařkanı</b>	: Do. Dr. Mge SAYIT	.....
<b>Raportr ye</b>	: Prof. Dr. Geylani KARDAŐ	.....
<b>ye</b>	: Prof. Dr. Berk CANBERK	.....
<b>ye</b>	: Do. Dr. Vahid AKRAM	.....
<b>ye</b>	: Do. Dr. Kaya OđUZ	.....



## EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

### ETİK KURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Doktora Tezi olarak sunduğum “Sanallaştırma ve YTA Teknolojilerine Dayalı Video İletimi” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

18.10.2023

Mustafa TÜKER



**ÖZET****SANALLAŞTIRMA VE YTA TEKNOLOJİLERİNE DAYALI  
VIDEO İLETİMİ**

TÜKER, Mustafa

Doktora Tezi, Uluslararası Bilgisayar Anabilim Dalı

Tez Danışmanı: Doç. Dr. Müge SAYIT

Ekim 2023, 92 sayfa

Günümüzün ileri ağ teknolojileri, Yazılım Tanımlı Ağlar (YTA) ve Ağ Fonksiyonlarının Sanallaştırılması (NFV) gibi yenilikçi yöntemlerle desteklenmektedir. Bu teknolojiler, internet uygulamalarını daha özelleştirilebilir ve esnek bir yapıya kavuşturarak, özellikle kaynak ihtiyaçlarının sürekli arttığı multimedya uygulamaları için büyük bir önem taşımaktadır. Artan video akışı talepleri, kullanıcılara üst düzey bir Kalite Deneyimi (QoE) sunmak için ağların yeni teknolojilere adaptasyonunu zorunlu kılmaktadır. Bu tez çalışmasında, YTA ve NFV teknolojilerinin Büyük Paket Protokolü (BPP) ile nasıl kullanılacağı ve son dönemdeki ağ teknolojileriyle nasıl uyum sağlayabileceği ele alınmaktadır. Video iletiminin performansını artırmak için YTA, NFV ve BPP kavramlarının bir araya getirilerek optimize edilmesine dair yenilikçi bir yaklaşım sunulmaktadır. Bu amaçla video akışlandırma ihtiyaçlarına cevap veren yeni BPP paket yapıları sunulmuştur. Ayrıca, YTA denetleyicisinin yönetimindeki bir ağda, ağ içi kalite adaptasyonu yapısı ve kenar ağda sanallaştırılmış BPP işlevlerini video iletimi için kullanan bir mimari önerilmektedir. Deneysel sonuçlar, kenar ağlarda ağ içi kalite adaptasyonunun uygulanmasının ve sanallaştırılmış BPP işlevinin kullanılmasının ölçeklenebilirlik ve kullanıcı deneyiminde gözle görülür bir iyileşme sağladığını göstermiştir.

**Anahtar sözcükler:** Yazılım Tanımlı Ağlar, Ağ Fonksiyonlarının Sanallaştırılması, Büyük Paket Protokolü, Kullanıcı Deneyimi



**ABSTRACT****VIDEO TRANSMISSION BASED ON VIRTUALIZATION AND SDN  
TECHNOLOGIES**

TÜKER, Mustafa

PhD in International Computer Department

Supervisor: Assoc. Prof. Dr. Müge SAYIT

October 2023, 92 pages

Today's advanced network technologies are supported by innovative methods such as Software-Defined Networks (SDN) and Network Function Virtualization (NFV). These technologies, by making internet applications more customizable and flexible, hold significant importance especially for multimedia applications where resource requirements are continuously increasing. Increasing video streaming demands necessitate the adaptation of networks to new technologies to deliver a high level of Quality Experience (QoE) to users. In this thesis, how SDN and NFV technologies can be used with the Big Packet Protocol (BPP) and how they can be compatible with recent network technologies are handled. An innovative approach is presented to combine and optimize the concepts of SDN, NFV and BPP to improve the performance of video transmission. For this purpose, new BPP package structures that respond to video streaming needs are presented. Besides, an architecture that uses in-network quality adaptation structure and virtualized BPP functions in the edge network for video transmission in a network managed by the SDN controller is proposed. Experimental results demonstrate that the application of in-network quality adaptation at the edge and the use of virtualized BPP functions provides a noticeable improvement in scalability and user experience.

**Keywords:** Software-Defined Networks, Network Function Virtualization, Big Packet Protocol, Quality of Experience



## ÖNSÖZ

Bu tezin amacı Yazılım Tanımlı Ağlar (YTA) ve Sanallaştırılmış Ağ İşlevleri (NFV) gibi yenilikçi teknolojilerin kullanımını inceleyerek, bu teknolojilerin gelişmiş ağ protokollerinin uygulanabilirliği üzerindeki etkisini ele almaktır. Bu tez çalışması, “Sanallaştırma ve YTA Teknolojilerine Dayalı Video İletimi” başlıklı çalışmanın sonucudur ve doktora derecemi alabilmek için tamamladığım tüm çalışmaları temsil etmektedir. Bu çalışma sırasında edindiğim bilgi ve deneyimlerin, kariyerimdeki yükselme aşamalarına katkıları büyüktür. Araştırma sürecinde karşılaştığım zorluklar, beni daha iyi bir öğrenci ve araştırmacı yapmıştır. Bu tez çalışmasında, bu alanda yapılmış yol gösterici birçok değerli kaynaktan yararlanılmıştır. Bu kaynaklar, çalışmamın temelini oluşturmakta ve literatürde yer bulabilecek bir tez yazabilmem için gereken bilgileri sağlamaktadır. Bu çalışma danışmanım Doç. Dr. Müge SAYIT’ ın rehberliği altında gerçekleştirilmiştir ve onun bilgi ve deneyimi benim için büyük bir ilham kaynağı olmuştur. Umarım bu çalışma, ilgili alandaki araştırmacılara faydalı olur ve gelecekteki çalışmalar için bir temel oluşturur.

İZMİR

18.10.2023

Mustafa TÜKER



**İÇİNDEKİLER**

	<u>Sayfa</u>
ÖZET .....	vii
ABSTRACT .....	ix
ÖNSÖZ .....	xi
İÇİNDEKİLER DİZİNİ .....	xiii
ŞEKİLLER DİZİNİ .....	xvi
TABLolar DİZİNİ .....	xix
SİMGELER VE KISALTMALAR DİZİNİ .....	xx
1.GİRİŞ .....	1
1.1. Tezin Sunduğu Katkılar .....	3
1.2. Tezin Organizasyonu .....	4
2.KULLANILAN TEKNOLOJİLER VE LİTERATÜR TARAMASI .....	6
2.1 Yazılım Tanımlı Ağlar .....	6
2.1.1 Yazılım tanımlı ağ mimarisi .....	7
2.1.2. YTA denetleyicileri .....	9
2.1.3. ONOS .....	10

## İÇİNDEKİLER (devam)

	<b><u>Sayfa</u></b>
2.2. Ağ Fonksiyonlarının Sanallaştırılması .....	13
2.3. Mininet .....	14
2.4. Ölçeklenebilir Video Kodlama (SVC) .....	14
2.5. Büyük Paket Protokolü (BPP).....	16
2.6. Literatür Taraması .....	17
3. BPP PAKET YAPISI VE PAKET STRATEJİLERİ .....	24
3.1. Video İletimi İçin BPP'de Yapılan Geliştirmeler .....	24
3.1.1. BPP paket yapısı .....	25
3.1.2. Video akışı için oluşturulan ek BPP blok alanları .....	28
3.1.3. Video akışında BPP bloğunda kullanılan paket küçültme komutu .....	32
3.1.4. Video akışının BPP' ye eşlenmesi .....	34
3.1.5. BPP paketleri için oluşturulan SVC H264 yapısı .....	35
3.2. BPP Paketleme Stratejileri .....	37
4. YTA ORTAMINDA BPP ÜZERİNDEN VIDEO İLETİMİ İÇİN GELİŞTİRİLEN UYGULAMALAR .....	42
4.1. YTA ile Ağ İçi Video Akış Sistemi Uygulaması .....	45
4.2. Sanallaştırılmış BPP İşlemlerinin Kenar Ağda Uygulanması .....	49
4.2.1. CPU yükünün ölçülmesi .....	49
4.2.2. Sanallaştırılmış BPP işlemi .....	52

**İÇİNDEKİLER (devam)**

	<u>Sayfa</u>
4.2.3. Paket küçültme için kullanılan algoritma .....	53
4.2.4. QOE hesaplama formülü .....	56
5. DENEYSEL SONUÇLAR .....	58
5.1. Ağ İçi BPP Kalite Uyarlaması Kullanılarak Yapılan Karşılaştırma Testleri	58
5.2. Ağ İçi Kalite Uyarlaması ile HAS Karşılaştırması .....	63
5.3. Sanallaştırılmış BPP İle Ağ İçi BPP Yönteminin Karşılaştırılması .....	65
5.4. Sanallaştırılmış BPP Yöntemi Performans Testleri .....	67
5.5. Paket Stratejileri Karşılaştırma Testleri .....	70
6. SONUÇLAR .....	75
KAYNAKLAR DİZİNİ .....	78
TEŞEKKÜR .....	89
ÖZGEÇMİŞ .....	91

## ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1 YTA Mimarisi.....	8
2.2 REST API Arayüzü.....	11
2.3 OPA (Off-Platform Application) ile REST API Etkileşimi.....	12
2.4 SVC Video Dosyası Yapısı.....	16
3.1 BPP Başlık ve Blok Yapısı .....	25
3.2 BPP Komut ve Metadata Bloklarının Alanları .....	26
3.3 BPP Paket Yapısı .....	27
3.4 Parça (Chunk) Alanları .....	29
3.5 Paket Küçültme (Packet Wash) İşlemi.....	33
3.6 Paket yapısı .....	35
3.7 Makro Bloklar .....	36
3.8 Oluşan Parça Boyutları- (a) Eşit Bölme Stratejisi, (b) Dinamik Bölme Stratejisi, (c) Sırayla Bölme Stratejisi, (d) Tamamen Dolu Bölme Stratejisi .....	38
3.9 Eşit Bölme ve Dinamik Bölme Paket Stratejisi .....	39
3.10 Sırayla Bölme Paket Stratejisi.....	40
3.11 Tamamen Dolu Bölme Paket Stratejisi .....	40

## ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
4.1 YTA Denetleyicileri İle BPP İşlemi Yapılması .....	45
4.2 BPP İşleminin Gerçekleştirilmesi .....	46
4.3 BPP İşlemlerinin İşlemci Üzerindeki Etkisi .....	50
4.4 Çok Sayıda İstemciye Ait BPP İşlemlerinin İşlemci Üzerindeki Etkisi ..	51
4.5 Sanallaştırılmış BPP Yapısı .....	53
4.6 Paket Küçültme Algoritması .....	54
5.1 Değişik Bant Genişliği Değerlerinde Alınan Çerçevesel- (a) 0,5 Mbps BPP Alınan Video Katmanları, (b) 0,5 Mbps UDP Alınan Video Katmanları, (c) 0,8 Mbps BPP Alınan Video Katmanları, (d) 0,8 Mbps UDP Alınan Video Katmanları, (e) 1,5 Mbps BPP Alınan Video Katmanları, (f) 1,5 Mbps UDP Alınan Video Katmanları, (g) Artan Bw BPP Alınan Katmanlar, (h) Artan Bw UDP Alınan Katmanlar, (i) Azalan Bw BPP Alınan Katmanlar, (j) Azalan Bw UDP Alınan Katmanlar .....	59
5.2 BPP ve UDP İçin PSNR Değerleri.....	62
5.3 BPP ve UDP Oluşan Video Dosyası Örneği- (a) Çerçeve 81: BPP İstemcine ait videonun ekran görüntüsü, (b) Çerçeve 81: UDP İstemcine ait videonun ekran görüntüsü.....	62
5.4 ONOS Kontrolör Üzerinden Gerçekleştirilen Test Sonuçları- (a) Ortalama Bit Hızı, (b) Duraksama Zamanı, (c) Kalite Değişimi, (d) PSNR değeri .64	64
5.5 Geliştirilen İki BPP Yönteminin Karşılaştırılması- (a) Ortalama Bit Hızı, (b) Duraksama Zamanı, (c) Kalite Değişimi, (d) PSNR değeri .....	66

**ŞEKİLLER DİZİNİ (devam)**

<u>Şekil</u>	<u>Sayfa</u>
5.6 Sanallaştırılmış BPP Yönteminin Sabit Bant Genişliği ile Kalite Parametreleri- (a) Ortalama Bit Hızı, (b) Kalite Değişimi, (c) Genel QoE, (d) QoE Doğruluğu.....	68
5.7 Sanallaştırılmış BPP Yönteminin Dinamik Bant Genişliği ile Kalite Parametreleri- (a) Genel QoE, (b) QoE Doğruluğu .....	70
5.8 Paketleme Stratejilerine Göre Genel QoE Sonuçları- (a) Sabit Bant Genişliği Genel QoE, (b) Dinamik Bant Genişliği Genel QoE .....	71
5.9 Paketleme Stratejilerine Göre Toplam Duraksama Süreleri- (a) Sabit Bant Genişliği Duraksama Süresi, (b) Dinamik Bant Genişliği Duraksama Süresi.....	72
5.10 Paketleme Stratejilerine Göre Ortalama Bit Hızı- (a) Sabit Bant Genişliği Ortalama Bit Hızı, (b) Dinamik Bant Genişliği Ortalama Bit Hızı.....	73
5.11 Paketleme Stratejilerine Göre Toplam Kalite Değişimi- (a) Sabit Bant Genişliği Kalite Değişimi, (b) Dinamik Bant Genişliği Kalite Değişimi	73

**TABLolar DİZİNİ**

<u>Tablo</u>	<u>Sayfa</u>
1.1 Tez sürecinde gerçekleştirilen yayınların listesi.....	4
3.1 BPP Paketine Ait Alanlar .....	31
4.1 Öncelik Değerleri Tablosu.....	44
5.1 Ağ İçi BPP QoE Metriklerinin Karşılaştırılması .....	61
5.2 Ağ İçi BPP QoE Değerlerinin Karşılaştırılması .....	65
5.3 Testlerde Kullanılan Video Dosyasının Özellikleri.....	71
5.4 Üretilen Paket Sayıları .....	71

## SİMGELER VE KISALTMALAR LİSTESİ

<u>Simgeler</u>	<u>Açıklama</u>
$\alpha$	bit hızı parametre ağırlığı
$\beta$	kalite değişimi parametre ağırlığı
$\delta$	duraksama parametre ağırlığı
$Q_k$	$k'$ inci segmente ait video bit hızı
$l_k$	$k'$ inci segmentin katmanı
$dp_k$	$k'$ inci segmentin duraksama süresi
$ts_k$	$k'$ inci segmentin alındığı zaman
$p_k$	$k'$ inci segmentin oynatma zamanı

<u>Kısaltma</u>	<u>Açıklama</u>
<b>API</b>	Uygulama Programlama Arayüzü
<b>AR</b>	Artırılmış Gerçeklik
<b>BPP</b>	Büyük Paket Protokolü
<b>CDN</b>	İçerik Dağıtım Ağı
<b>CGS</b>	Kaba taneli ölçeklenebilirlik
<b>CLI</b>	Komut Katmanı Arayüzü
<b>CRC</b>	Döngüsel Artıklık Kontrolü
<b>DASH</b>	HTTP Üzerinden Dinamik Uyarlamalı Akış

**SİMGELER VE KISALTMALAR LİSTESİ (devam)**

<u>Kısaltma</u>	<u>Açıklama</u>
<b>ETSI</b>	Avrupa Telekomünikasyon Standartları Enstitüsü
<b>FPS</b>	Saniye Başına Kare
<b>GUI</b>	Grafiksel Kullanıcı Arayüzü
<b>HAS</b>	HTTP Adaptif Akışlandırma
<b>IPC</b>	Prosesler Arası İletişim
<b>KPI</b>	Hedef Performans Göstergeleri
<b>MEC</b>	Mobil Kenar Bulutları
<b>MGS</b>	Orta taneli ölçeklenebilirlik
<b>MPLS</b>	Çok Protokollü Etiket Anahtarlama
<b>MTU</b>	Maksimum İletim Birimi
<b>NAL</b>	Ağ Soyutlama Katmanı
<b>NAT</b>	Ağ Adres Çevirisi
<b>NON VCL</b>	Video Olmayan Kodlama Katmanları
<b>NFV</b>	Ağ Fonksiyonu Sanallaştırma
<b>OPA</b>	Platform Dışı Uygulama
<b>ONOS</b>	Açık Ağ İşletim Sistemi
<b>PSNR</b>	Tepe Sinyal Gürültü Oranı

**SİMGELER VE KISALTMALAR LİSTESİ (devam)**

<u>Kısaltma</u>	<u>Açıklama</u>
<b>RPC</b>	Uzak Yordam Çağrısı
<b>QoE</b>	Deneyim Kalitesi
<b>QoS</b>	Hizmet Kalitesi
<b>QUIC</b>	Hızlı UDP İnternet Bağlantıları
<b>RPC</b>	Uzak Metod Çağırımı
<b>SDN</b>	Yazılım Tanımlı Ağ
<b>SNR</b>	Sinyal-Gürültü Oranı
<b>SRTP</b>	Güvenli Gerçek Zamanlı Aktarım Protokolü
<b>SVC</b>	Ölçeklenebilir Video Kodlama
<b>TCP</b>	Transfer Kontrol Protokolü
<b>ToS</b>	Hizmet Tipi
<b>TSN</b>	Zamana Duyarlı Ağ İletişimi
<b>UDP</b>	Kullanıcı Datagram Protokolü
<b>UHD</b>	Ultra Yüksek Çözünürlük
<b>VCL</b>	Video Kodlama Katmanları
<b>VNF</b>	Sanal Ağ Fonksiyonu
<b>VoD</b>	Video Dağıtım Verimliliği

<b>VoIP</b>	İnternet Protokolü Üzerinden Ses
<b>VR</b>	Sanal Gerçeklik
<b>v-SDN</b>	Sanal Yazılım Tanımlı Ağ
<b>WAN</b>	Geniş Alan Ağı





## 1. GİRİŞ

Günümüz ağ teknolojilerinde, Yazılım Tanımlı Ağlar (YTA) ve Sanallaştırılmış Ağ İşlevlerinin (NFV) kullanılması gibi son dönemde ortaya çıkan teknolojiler, yeni protokollerle birlikte, internet uygulamalarına özgü çeşitli gereksinimlere göre özelleştirilebilecek esnek bir altyapı sağlamaktadır. YTA, kontrol ve veri düzlemlerini birbirinden ayırarak yeni bir ağ tasarım yaklaşımı sunmaktadır (Binsahaq et al., 2019). Bu iki düzlemin ayrılması, ağın merkezi bir noktadan doğrudan programlanabilir, sanallaştırılabilir ve yönetilebilir olmasını sağlamaktadır. YTA' nın temel yararlarından biri, ağ üzerinde çalışan uygulamaların ihtiyaçlarına özel olarak yönlendirme algoritmaları ve servis tanımlamaları geliştirebilme kabiliyetidir.

İnternet' teki en popüler uygulamalardan biri olan multimedya uygulamaları, kaynak taleplerini her geçen gün artırmaktadır. Bu durum, kullanıcı gereksinimlerinde ve beklentilerinde artışa sebep olmaktadır. Ayrıca Uzaktan cerrahi, Artırılmış Gerçeklik (AR)/ Sanal Gerçeklik (VR), gözetim sistemleri ve mobil yayıncılık gibi zorlu uygulamaların gereksinimleriyle birlikte video akışlandırma konusundaki ihtiyaçlar sürekli olarak artmaktadır. Video akışlandırma konusundaki artan talep, kullanıcılara daha iyi bir Kalite Deneyimi (QoE) sunulabilmesi için temel ağ koşullarını göz önünde bulundurarak yeni teknolojilerin geliştirilmesini gerektirmektedir. Bu nedenle, günümüzdeki video akışı uygulamaları ve standartları iyi bir performans sunuyor olsalar da gelecekteki uygulamaların daha düşük gecikme süresi, daha yüksek güvenilirlik ve daha fazla uyumluluk gereksinimleri şu an evrimleşmeye devam eden ağ ve hizmet operatörlerini zorlamaya devam edecektir. Video akışlandırma uygulamaları tarafından oluşturulan yüksek trafik hacmi, paket kayıplarında bir artışa neden olabilmektedir ve bu kayıplardan dolayı yeniden iletim gerçekleştirilmesi ağda gecikmenin artmasına yol açmaktadır. Bu sorunu sunucu ve istemcide ele almak için mevcut birçok teknik bulunmasına rağmen, Tez içerisinde uygulanan Paket Küçültme yöntemi kaybedilen paket sayısını azaltmak için kullanılacak yeni ve olumlu katkılar sunan bir tekniktir. Paket Küçültme, ağ üzerinden bir paketin boyutunu ayarlamak için son yıllarda geliştirilmiş yeni bir fikirdir (Dong and Li., 2019a; Westphal et al., 2019). Paket Küçültme yaklaşımı ile ağ üzerindeki bant genişliği sınırlı hale geldiğinde, bir paketin tamamını düşürmek yerine paket sınırlı bant genişliği değerine göre küçültülerek herhangi bir yeniden iletim maliyeti olmadan aktarılmaktadır.

Şu anda, standart protokollerin hiçbiri paket küçültmeyi doğrudan desteklememektedir. Büyük Paket Protokolü (BPP), paket küçültmeyi kullanabilecek şekilde tasarlanmış bir protokoldür. BPP, gelecekteki ağlardaki düşük gecikmeli ve yüksek güvenilirlikli uygulamalar için kullanılan protokollerden biri olarak tasarlanmıştır. 2018' deki tanıtımından bu yana (Li et al., 2018), BPP ile ilgili birçok çalışma bulunmaktadır ve bu çalışmalar BPP' nin kullanımının avantajlarını göstermektedir. Paketleri küçültme sürecinin kullanılması, paketlerdeki parçaların küçültülmesiyle gecikme değerinin azaltıldığını göstermektedir (Dong and Li., 2019c).

Literatürde sunulan BPP ile ilgili çalışmalar genel anlamda bu protokolün sağlayabileceği olası avantajlardan söz etse de protokolün günümüz teknolojileriyle uygulanabilirliği konusundaki yaklaşımlar ve öneriler sınırlıdır. Tez çalışması kapsamında, YTA ve NFV teknolojilerini BPP protokolünün gerçekleşmesinde kullanarak; hem bu protokolün son dönemin gelişmiş ağ teknolojileriyle uygulanabilirliğine ışık tutmakta olan, hem de YTA, NFV ve BPP kavramlarını video iletim performansının artırılması amacıyla özelleştiren bir mimari sunan bir çalışma ortaya konmuştur.

Bu tez çalışmasında, video akışı uygulamaları için düşük gecikme ile yüksek QoE sunmak üzere yazılım tanımlı ağ ortamında paket küçültme ve BPP protokolünü kullanan bir mimari önerilmektedir. Paket Küçültme işlemlerinin uygulanabilmesi için, ağ sanallaştırma tekniklerinden faydalanılmaktadır. HTTP Adaptif Video Akışı (HAS) kullanarak video akışı, istemcilerin gözlemlenen ağ parametreleri temelinde video kalitesini adapte ettiği en yaygın ve başarılı uygulamadır, ancak bu çalışmada, paket küçültme yaklaşımıyla ağ içi video kalite adaptasyonu gerçekleştirilmektedir.

H264 Ölçeklenebilir Video Kodlama (SVC) kodlayıcı (Schwarz et al., 2007), BPP ile kullanımda avantaj sağlayan katmanlı videolar oluşturulmasına izin verdiği için seçilmiştir. SVC kodlama kullanılarak oluşturulan BPP paketlerinin, video akışı uygulamalarının performansını artırmak için etkili bir yol olduğu (Tuker et al., 2023) yayınıyla gösterilmiştir. Tez çalışması içerisinde aktarılan video, zenginleştirme katmanları çıkarılarak kalite adaptasyonuna izin veren özellikleri nedeniyle video paket küçültmesini uygun hale getiren H.264 SVC katmanlı kodek kullanılarak kodlanmıştır.

Tez çalışması kapsamında geliştirilen yaklaşım, istemcide kesintisiz akışlandırmayı, düşük gecikmeyle sürekli paket teslimatını sağlamaktadır. Bu çalışmanın amacı, yazılım tanımlı ağ ortamında sanallaştırılmış ağ işlevlerini kullanarak video akışı uygulamalarının performansını artırmaktır. Bu tezde önerilen çalışmalar iki sınıfa ayrılabilir:

- Akışlandırmaya uygun yapıda BPP Paket yapılarının oluşturulması için gerçekleştirilen çalışmalar
- BPP işlemlerinin gerçekleştirilmesi için YTA denetleyici üzerinde ve sanallaştırma teknolojileriyle kenar ağda oluşturulan uygulamalar

### 1.1. Tezin Sunduğu Katkılar

Literatürde Paket küçültme yöntemi ve BPP kullanılarak gerçekleştirilmiş herhangi bir video akışlandırma çalışması yokken tez çalışması kapsamında bir model tasarlanarak (Clayman et al., 2021a), (Clayman et al., 2021b) yayınları ile bazı ön değerlendirmeler sunulmuştur. Elde edilen ilk sonuçlar, ağ içi video kalitesi adaptasyonunun, mevcut ve gelecekteki video akışlandırma uygulamalarının gereksinimlerini karşılayabilecek verimli bir teknik olabileceğini göstermektedir. Bu çalışmalarda, BPP fonksiyonlarını gerçekleştiren bir YTA denetleyici kullanılmaktadır. Aynı zamanda bu çalışmalar uygulama mimarisinin QoE üzerinde dikkate değer bir etkisi olduğunu göstermektedir. Çekirdek ağdaki işlem, merkezde birden çok akışı ele alırken daha fazla yük oluşturmaktadır, ayrıca istemcilerden uzakta ve çeşitli istemci farklılıklarıyla başa çıkmak daha zor olmaktadır. Bu nedenle sonraki çalışmalarda, istemcilere daha yakın olan ve işleme yükünü yayabilen daha yüksek QoE değerleri elde edilmesini sağlayan yeni bir mimari önerilmektedir (Tuker et al., 2023).

Bu tez çalışmasının katkıları şu şekildedir:

İlk olarak, geliştirilen BPP protokolü mevcut durumda kullanılan UDP, TCP ve HAS ile karşılaştırılarak ağ içi kalite adaptasyonunun potansiyel avantajları gösterilmektedir.

Tez çalışmaları kapsamında sunulan bir diğer katkı, Open Network Operating System (ONOS) YTA denetleyicisinde doğrudan paket küçültme işleminin uygulanmasının neden olduğu sorunların belirlenmesidir. ONOS' ta Derin Paket

İncelemesi (DPI) ve paket güncellemelerinin yüksek derecede CPU yoğunluğu gerektirdiği ve ölçeklenebilir olmadığı gösterilmektedir.

Son olarak, sanallaştırılmış BPP fonksiyonlarının ağıın kenar kısmına alınarak kullanılmasını gerçekleştiren bir mimari sunulmaktadır. Paket küçültme algoritması ve paket küçültmenin akışlar üzerindeki etkileri gösterilmektedir. Çeşitli protokoller ile karşılaştırmalar yapılarak, elde edilen QoE' yi gösteren deneysel sonuçlar sunulmaktadır. Bu yeni mimari, 5G/6G ve NFV ile uyumludur ve ölçeklenebilir ve optimize edilmiş sonuçlar sağlamaktadır.

Tez çalışması içerisindeki konulardan 3 adet Uluslararası Konferans bildirisi ve 1 Adet SCI Dergi Makalesi yayını gerçekleştirilmiştir. Ayrıca tez konusu TÜBİTAK (121E373) projesi iş paketleri arasında yer almaktadır. Tablo 1.1. içerisinde tez sürecinde gerçekleştirilen yayınların bir listesi sunulmaktadır.

Tablo 1.1. Tez sürecinde gerçekleştirilen yayınların listesi

No	Yayın Adı	Yayınlandığı Yer	Durum
1	Managing Video Processing and Delivery using Big Packet Protocol with SDN Controllers	IEEE International Conference on Network Softwarization (NetSoft) 2021	Yayınlandı
2	The Future of Media Streaming Systems: Transferring Video over New IP	IEEE International Conference on High Performance Switching and Routing (HPSR) 2021	Yayınlandı
3	In-Network Video Quality Adaption using Packet Trimming at the Edge	26th Conference on Innovation in Clouds, Internet and Networks (ICIN) 2023	Yayınlandı
4	Using Packet Trimming at the Edge for In-Network Video Quality Adaption	Annals of Telecommunications	Yayınlandı

## 1.2. Tezin Organizasyonu

Bu tez şu şekilde organize edilmiştir:

- Tezin 2. bölümünde tez içerisinde yer alan teknolojilerin anlatımı ve video akışlandırmada kullanılan yöntemlerin literatür taraması yer almaktadır.
- Tezin 3. bölümünde BPP işlemlerine uygun olması için geliştirilen paket yapısı ve kullanılan paketleme stratejileri yer almaktadır.

- Tezin 4. bölümünde BPP operasyonunun gerçekleştirilmesini sağlayan uygulamaların geliştirilmesi ve özelliklerinden bahsedilmektedir.
- Tezin 5. bölümünde yapılan tüm geliştirmelere ait testler ve sonuçları yer almaktadır.
- Tezin 6. bölümünde yapılan çalışma ve elde edilen sonuçlara yönelik değerlendirmeler ve gelecekte yapılması planlanan çalışmalar yer almaktadır



## 2. TEZ ÇALIŞMASINDA KULLANILAN TEKNOLOJİLER VE LİTERATÜR TARAMASI

Bu bölümde tez çalışmasında kullanılan önemli teknolojilerin bilgilendirilmesi yapılmaktadır. İlk olarak YTA ve NFV kavramları açıklanmakta ve kullanılan YTA denetleyici özellikleri açıklanmaktadır. Ayrıca tezin hazırlık sürecinde kullanılan araçların tanıtımları ve özellikleri verilmektedir. Tez çalışmasında yer alan teknolojilere ait literatür çalışmaları bu bölümün son kısmında yer almaktadır.

### 2.1. Yazılım Tanımlı Ağlar

Geçtiğimiz birkaç yıl içerisinde, Yazılım Tanımlı Ağ teknolojileri bilgisayar ağ cihazlarının yönetimi ve veri düzlemi ile kontrol düzleminin birbirinden ayrıştırılması gibi özellikleriyle gelecek nesil ağ mimarisi olarak ortaya çıkmıştır (SDN., 2023). Teknolojik gelişmelerle birlikte günümüzde, veri merkezi ağları, bulut bilişim, sosyal ağlar ve modern ağ ortamlarında sıklıkla kullanılmakta olan bu kavram gelecek ağları için oldukça yüksek bir potansiyel oluşturmaktadır.

Günümüz internetinde, video ve ses akışı, ağ depolama ve çevrimiçi video oyunları gibi öne çıkan ağ hizmetleri ve uygulamaları, ağ altyapı kaynaklarını zorlamaktadır. Var olan QoS (Hizmet Kalitesi) garantisi sağlama çabalarına rağmen, internetin geneli bu garantileri sağlamakta yetersiz kalmaktadır. YTA ise ağ kaynaklarına daha fazla esneklik, soyutlama, kontrol ve programlama yeteneği sunarak uçtan uca QoS garantisi sağlama vaadiyle hızla gelişen bir ağ paradigmasıdır (Binsahaq et al., 2019). İnternetin giderek büyümesi ve farklı türdeki ağ cihazlarının artış göstermesiyle, İnternet Protokolü üzerinden Ses (VoIP), e-mail, video konferans, video akışlandırma, online oyun ve e-ticaret gibi farklı türdeki ağ hizmetlerinin ortaya çıkmasına sebep olmuştur. Video akışlandırma gibi bu uygulamalar, gecikme ve bant genişliği kısıtları açısından ağ üzerinden iletim için çok daha hassastır. Video konferans, ağ bant genişliğinin büyük bir kısmını tüketirken güçlü gecikme gereksinimlerine sahiptir. Trafik mühendisliğinde, YTA, esnek denetleyici aracılığıyla video konferans performansını optimize eder. Günümüzde kullanılan en iyi çaba modeli, günümüz ve gelecek ağlarında ortaya çıkmakta olan bu hizmetlerin artan gereksinimlerinin karşılanması için yeterli değildir. Geleneksel ağ mimarilerinde, bant genişliğinin hızlı değişmesi, ağ topolojisinin ve yönlendirme bilgisinin sürekli olarak değişmesi ve depolama kapasitelerinin yetersiz kalması gibi artan bu gereksinimlerin

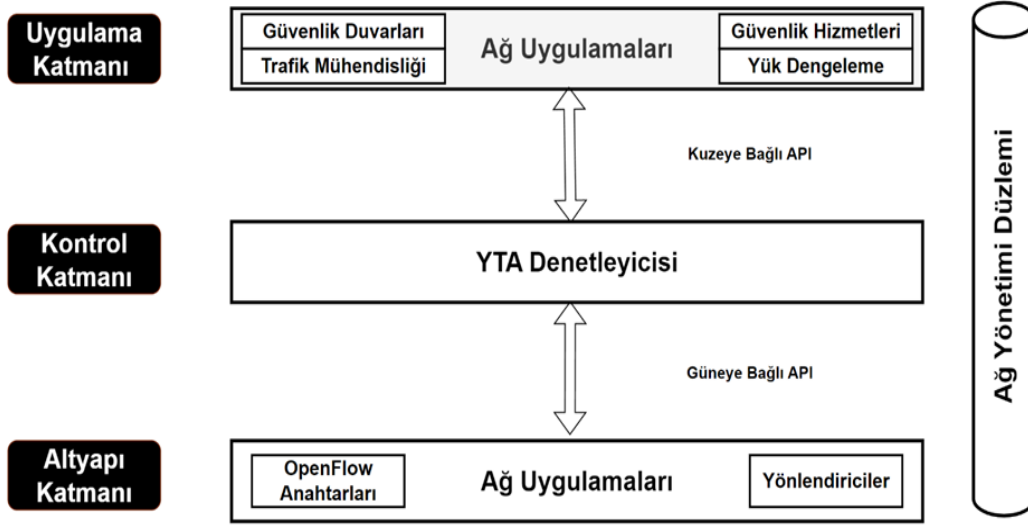
karşılanmasına yönelik olarak YTA bir çözüm mimarisi olarak ortaya çıkmıştır (Karakus and Durresi, 2017).

Büyük ölçekli bilgisayar ağları, tipik olarak, anahtarlama ve yönlendirmeden, paket filtreleme veya Ağ Adres Çevirisi (NAT) gibi gelişmiş ağ işlevlerine kadar çeşitli görevleri gerçekleştiren çeşitli aygıtlardan oluşur. Bu ağ cihazları geleneksel ağ ortamlarında, farklı yönetim arayüzlerine sahiptir ve genellikle cihaz bazında yapılandırılırlar. Yazılım Tanımlı Ağlarda ise, anahtar cihazlar üzerinde yönlendirme kararlarını veren merkezi olarak konumlandırılmış bir denetleyicinin var olması ile veri düzlemi, kontrol düzleminden ayrılmaktadır. Veri düzleminde yer alan anahtar cihazları, denetleyici tarafından verilen yönlendirme kararlarını uygulamaktan sorumludur. Kontrol ve veri düzlemi arasındaki bu ayrım, veri düzlemi cihazlarının genel ve basitleştirilmiş bir şekilde tasarlanmasına olanak tanımaktadır. Başka bir deyişle, ağ cihazları, sadece kontrol düzlemi tarafından belirtilen akış kurallarına bağlı olarak paketleri işlemekten sorumlu olan basit, akıllı olmayan cihazlar haline gelebilir. YTA etkin yönlendirme öğeleri, genellikle bir eşleştirme-eylem soyutlamasını ortaya çıkarır. Böyle bir cihazdan geçen ağ paketlerinin, paket başlığının bir alt kümesini bir kuralda tanımlanan bir başlık alanıyla eşleştirerek iletme veya bırakma gibi bir eyleme atandığı anlamına gelmektedir. Mantıksal nitelikte merkezi konumlanan denetleyiciyse bir yazılım ağı üzerindeki geniş topolojinin bütünüyle yönetilmesini sağlamaktadır (Michel and Keller., 2017).

YTA, yönlendirme protokollerini kullanarak anahtar ve yönlendirici cihazların yönlendirme kararlarının bir denetleyiciye bağlı olması noktasında geleneksel ağlardan farklılaşmaktadır. Geleneksel ağlar üzerindeki ağ cihazları, küresel ağ topolojisinden habersizken, YTA küresel topoloji hakkında bilgi sahibidir. Geleneksel ağ altyapılarının yönetiminin zor olması sorununa çözüm olarak önerilen YTA paradigması, kontrol ve veri katmanlarını ayırarak ölçeklendirme zorluklarını aşmayı amaçlamaktadır.

### **2.1.1. Yazılım tanımlı ağ mimarisi**

Uygulama katmanı, ağlar için farklı kontrol mekanizmaları sağlayan YTA denetleyicisi ile iletişim kurmak için kuzeye bağlı Uygulama Programlama Arayüzleri (API) kullanmaktadır. Güneye bağlı Uygulama Programlama Arayüzleri, denetleyici katmanı ile veri düzlemi cihazları arasındaki iletişim arayüzünü oluşturur; bu da uygulamanın, esnek ve programlanabilir yollarla iletim cihazlarının kontrol edilmesini sağlamaktadır.



Şekil 2.1 YTA Mimarisi (Islam et al., 2018)

- **Altyapı katmanı:** Şekil 2.1’ de görüldüğü gibi altyapı katmanı en alt katmandır ve yönlendiriciler, fiziksel/sanal anahtarlar, erişim noktası vb. gibi ağ cihazlarından oluşmaktadır. Bu cihazlar, YTA denetleyiciler tarafından Güneye Bağlı Uygulama Programlama Arayüzleri (API) aracılığıyla erişilebilir ve yönetilebilir. Ağ öğeleri ve denetleyicileri TLS bağlantısı gibi güvenli bağlantılar aracılığıyla iletişim kurabilir (Alsmadi and Xu, 2015). OpenFlow protokolü, denetleyiciler ile veri katmanı cihazları arasındaki iletişim için en yaygın kullanılan Güneye Bağlı API standardıdır.
- **Kontrol katmanı:** YTA’ nın çekirdeği olan kontrol katmanı, küresel ve dinamik bir ağ görünümünü mantıksal olarak koruyan, uygulama katmanından istekleri alan ve standart protokoller aracılığıyla ağ cihazlarını yöneten merkezi bir denetleyiciden oluşmaktadır (Li and Chen, 2015). Paket iletimi, ağ anahtarlama ve yönlendirmesinin temel mantık programlamasını içermektedir (Alsmadi and Xu, 2015). Temel talimatlar ve karara dayalı bilgiler ağ cihazlarına yani veri katmanına gönderilmektedir. Kontrol katmanında, bazı programlanabilir uygulama programlama arayüzleri (API) ile Uygulama ve Kontrol Katmanlarını bağlamaktan sorumlu Kontrol Katmanı elemanları bulunmaktadır.
- **Uygulama katmanı:** Bu katman, ağ hizmetlerine odaklanan bir dizi uygulamayı kapsar ve bunlar çoğunlukla kontrol katmanı ile iletişim kuran yazılım uygulamalarıdır (Li and Chen, 2015). IPS, IDS, güvenlik sistemleri, trafik izleme hizmetleri, yük dengeleme, bant

genişliği yönetimi, hizmet kalitesi gibi çeşitli uygulamalardan oluşmaktadır. Ayrıca, ağ cihazlarına erişebilen erişim uygulamalarını da içermektedir.

- **OpenFlow protokolü:** YTA ağında en yaygın kullanılan iletişim protokolüdür. Altyapı ve kontrol katmanı arasındaki iletişimi sağlar ve cihazların programlanabilirliğini desteklemektedir. Denetleyici tüm iletim kurallarını gerçekleştirirken, anahtarlar sadece önceden tanımlanmış eylemlerle iletim yapmaktadır. Bu, bir denetleyici gibi dış bir varlığın ağ akış paketlerini manipüle etmesine olanak tanımaktadır. OpenFlow anahtarlar, belirli bir anahtar için bir paketin giriş ve çıkış yollarını temsil eden akış tablolarına sahiptir. OpenFlow protokolü, denetleyicinin akış tablolarına erişmesini sağlar. Bu erişim, güvenli bir iletişim kanalı üzerinden sağlanmaktadır (Alsmadi and Xu, 2015).

### 2.1.2. YTA denetleyicileri

YTA altyapısını merkezi olan YTA denetleyicisi, kontrol katmanında ağ cihazları ve uygulamalar arasındaki iletişimi yönetme rolüyle öne çıkmaktadır. Hem açık kaynak topluluklarından hem de ticari sektörlerden birçok YTA denetleyicisi ortaya çıkmıştır.

Denetleyici özelliklerinin geliştirilmesine yönelik geniş çaplı araştırmalar, ölçeklenebilirlik, erişilebilirlik ve maliyet etkinliği gibi yönleri kapsamaktadır. Günümüzde, Linux Vakfı ve ONF (Open Network Foundation) gibi önde gelen kuruluşlardan birçok YTA denetleyicisi ortaya çıkmıştır. Örneğin, 2008’ de Nicira network tarafından önerilen Nox denetleyici, OpenFlow protokolünü destekleyen ilk YTA teknolojisi olarak bilinmektedir (Prabha et al., 2022). Diğer dikkat çeken bir isim olan Pox, Python tabanlı bir yan proje olarak ortaya çıkmıştır. Stanford Üniversitesi’nde hazırlanan Beacon, Java tabanlı bir OpenFlow denetleyicisidir ve etkinlikleri yönetme ve çoklu iş parçacıklı görevleri yürütme konusunda uzmandır (Paliwal et al., 2018). Bir diğer YTA denetleyicisi Maestro, Rice Üniversitesinden gelen java tabanlı çok iş parçacıklı bir denetleyicidir. Bu denetleyici, paralellikten yararlanarak maksimum performans elde etmek için ek verimlilik optimizasyon tekniklerini keşfeder. Temel çalışma prensibi, bir paketin ilk olarak denetleyiciye gönderilmesi ve güvenlik kontrolünden sonra uygun veri akışı girişlerinin veri düzleminde iletilmesidir. Ancak, saniyede yaklaşık 10 milyon isteğin geldiği veri merkezi gibi senaryolarda, bu çalışma yöntemi etkili değildir. Maestro’ nun temelde getirdiği yenilikler, kullanıcılardan gelen toplu istekleri toplu işleme, aktif olarak

çalışan bir kodun bir CPU'dan diğerine taşınmasındaki zorluklara getirdiği çözümler, farklı görevlere farklı önceliklerin atanması ve kuyrukta bekleyen görev miktarını belirleme özellikleridir (Paliwal et al., 2018). Diğer bir yanda ise, Cisco, IBM, HP ve Juniper gibi büyük liderlerden tedarikçiler de kendi YTA denetleyicilerini piyasaya sürmüş, kendi vizyonlarını ürünlerine enjekte etmişlerdir. Bununla birlikte, bu denetleyiciler de açık kaynak yaklaşımı ve geniş topluluk katılımlarıyla yeniliklerin merkezi olmaya devam etmektedir.

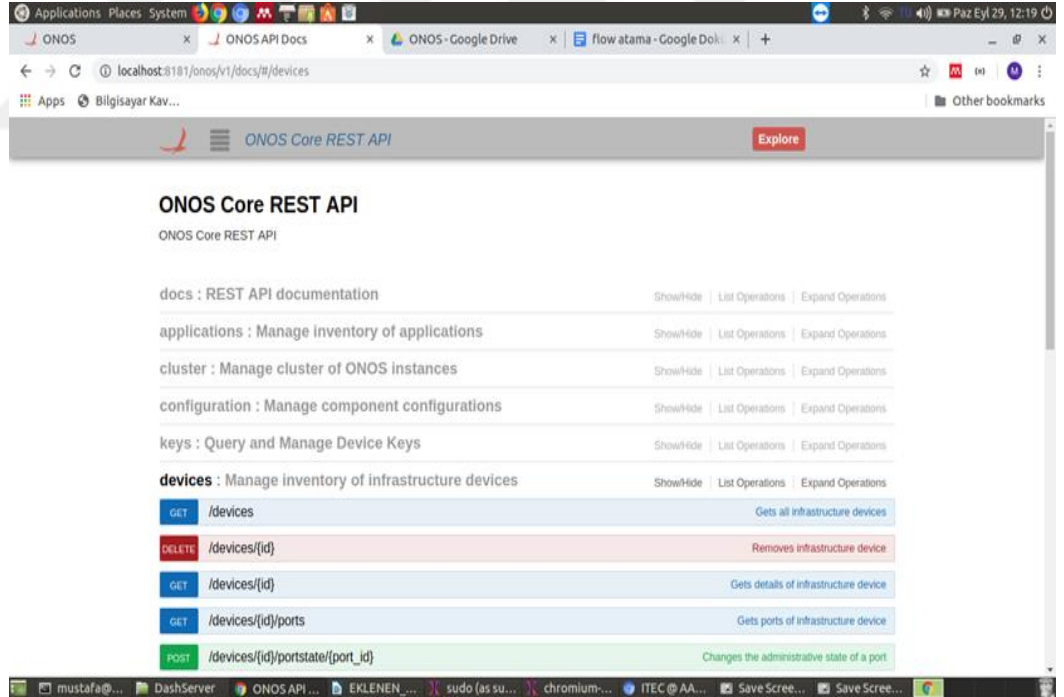
Bu alandaki bir dönüm noktası, 8 Nisan 2013' te Linux Vakfı tarafından sunulan OpenDayLight adlı açık kaynaklı YTA denetleyicisidir. Beacon modeli üzerine inşa edilmiştir (Paliwal et al., 2018). OpenDayLight tasarım felsefesi olarak modelleme tabanlı yazılım mühendisliği yaklaşımıyla geliştirilmiştir. Floodlight, Beacon modelinden kökenlerini alan başka bir önemli OpenFlow denetleyicisidir. Çeşitli uygulamalar için bir platform olarak tasarlanmıştır. 2014 yılında Stanford Üniversitesi tarafından geliştirilmiştir. Hem fiziksel hem de sanallaştırılmış OpenFlow anahtarları aracılığıyla çalışmaktadır. Bu denetleyici, Java tabanlıdır ve olaya dayalı bir yapıya sahiptir. Floodlight' ın öne çıkan bir özelliği de ağ eklentisi olarak çalışabilmesidir; bu da onun OpenStack ile entegre olmasını sağlar (Prabha et al., 2022). Bu özellik, araştırmacılara ağ kaynaklarını dinamik olarak görselleştirmeleri için olanak tanımaktadır.

### 2.1.3. ONOS

ONOS, servis sağlayıcı ağları için tasarlanmış güçlü bir YTA platformudur. ONOS bir YTA denetleyicisi olsa da tez çalışmaları bu denetleyici üzerinde gerçekleştirildiği için daha fazla bilgi vermek adına ayrı bir başlık altında incelenecektir. YTA' nın kontrol düzlemi olarak davranış gösteren ONOS, ağ öğelerini denetler ve çeşitli kümelerle iletişim kurar. ONOS denetleyicisini öne çıkaran şey, kümelerin çoklu makineler üzerinde merkezi olmayan bir şekilde dağıtılmasıyla elde edilen dayanıklılığı ve hata toleransıdır. Bir küme arızalandığında, başka bir YTA denetleyicisi sorunsuz bir şekilde rollerini ve yönetilen cihazları devralmaktadır. Ayrıca, ONOS hem geliştiricilerin hem de topluluk üyelerinin Komut Katmanı Arayüzünü (CLI) ve Grafiksel Kullanıcı Arayüzünü (GUI) kullanmalarına olanak tanımaktadır. ONOS yapısı, denetleyiciyle arayüz oluşturan birçok modül ve bir dizi uygulamadan oluşmaktadır. Kullanıcılar, bu uygulamaları gereksinime göre Komut Katmanı Arayüzü ve Grafiksel Kullanıcı Arayüzü aracılığıyla açıp kapatabilir. ONOS, Güney sınırı arayüzünü kullanarak bir dizi ağ ekipmanı, anahtarlar ve yönlendiriciler ile iletişim kurabilmektedir. Sistemin tasarımı, ONOS' un birkaç

sunucuda dağıtık bir şekilde çalışmasına izin vermektedir. ONF üyesi birçok şirketin desteğiyle ONOS, açık kaynaklı bir girişim olarak geliştirilmektedir. Global bir geliştirici topluluğu, onun çeşitli amaçlarına ve yaklaşan vizyonlarına katkıda bulunmaktadır. Docker konteynırları ve Vagrant gibi çeşitli araçlar ve platformlar, ONOS platformlarının dağıtımını kolaylaştırır, ancak donanım doğrudan kurulumları da mümkündür (ONOS, 2023). Bu donanım daha sonra YTA denetleyicisi olarak hizmet edebilmektedir.

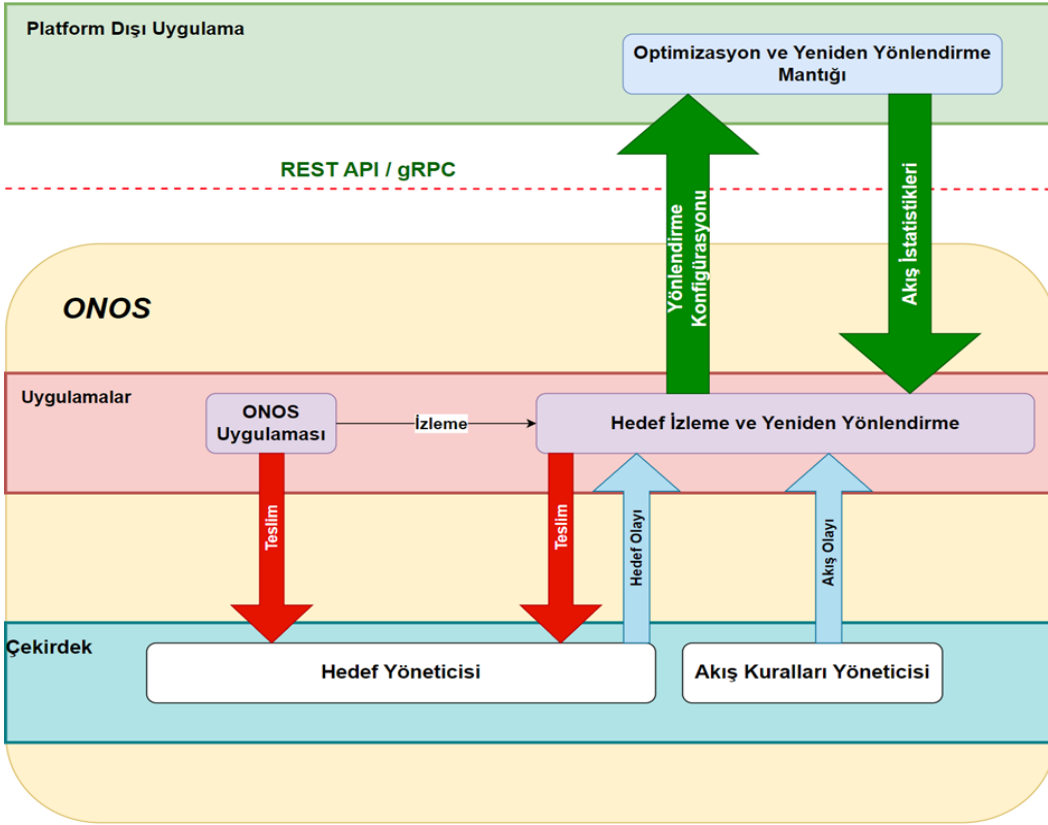
Bu tez çalışması için ONOS denetleyicisi, veri düzlemi ağ cihazlarını yönetmede kilit bir rol oynamaktadır. Tez çalışmasında, YTA denetleyicisi olarak ONOS kullanılmıştır. ONOS, servis sağlayıcıları, cihaz üreticileri, internet operatörleri ve ONF tarafından açık kaynak kodlu geliştirilmiş bir ağ işletim sistemidir. Kullanıcıları ağ cihazlarının yapısından, özelliklerinden ve protokollerinden soyutlayan, ticari uygulamalara dönük mimarisi ile diğer YTA denetleyicilerinden ayrılmaktadır. Dünyada, günümüzün önde gelen ağ ve telekomünikasyon firmaları ONOS kullanımını desteklemektedir (ONOS., 2023).



Şekil 2.2 REST API Arayüzü

REST API, web uygulamalarının ve komut dosyası dillerinin ONOS sistemiyle etkileşime girmesini kolay bir yolla sağlamak için vardır. Şekil 2.2' de gösterilen web arayüzüne <http://localhost:8181/onos/v1/docs/#/> adresinden lokal

olarak erişilebilmektedir. Buradan topolojiye ait birçok bilgi edinilebilmektedir. Atanan akışların bilgileri GET metodu ile elde edilebilir. İstenen bir akışın silme işlemi gerçekleştirilebilir. Yeni akışlar atanabilir. Oldukça kullanışlı bir arayüze sahiptir. Topoloji üzerinde gerçekleştirilecek işlemleri test etmek için faydalı bir platformdur. REST API en başta akış içeriklerini görme amaçlı kullanılırken okunan bir yayında Şekil 2.3'te gösterilen OPA (Off-Platform Application) kavramına rastlanmıştır (Sanvito et al., 2018).



Şekil 2.3 OPA (Off-Platform Application) ile REST API Etkileşimi (Sanvito et al., 2018)

ONOS üzerinde yüksek performans gerektirecek bir optimizasyon aracının çalıştırılması ONOS' un performansına olumsuz etki yapabilir. Bu yüzden mevcut bir uygulamayı (Örneğin Reactive Forwarding) düzenlemektense REST API ile iletişime geçen platform dışında bir uygulama oluşturmak performans açısından daha mantıklı bir çözüm olabilmektedir. Problemin boyutu gitgide büyüdükçe optimizasyon hesaplamalarının dışarıda yapılması daha mantıklı bir hale gelmektedir. OPA' lar ONOS ile REST API üzerinden etkileşime geçen ayrı bir işlem olarak yürütülen uygulamalardır. Burada REST API' nin tek dezavantajı yavaş olmasıdır. Bu problemin çözülmesi için de ONOS topluluğu REST API' nin yerine daha yüksek performans sağlayan gRPC 'nin eklenmesi yönünde çalışmalar

gerçekleştirmektedir. gRPC, programlar arası iletişimi sağlayan bir RPC(Remote Procedure Call) yöntemidir. Bu yöntem, uygulamalar arası iletişim (IPC), uzak metot çağırımı (RPC) ve Web Servis gibi haberleşme yapılarının temeli olacak şekilde kullanılabilir (GRPC., 2023).

## 2.2. Ağ Fonksiyonlarının Sanallaştırılması (NFV)

2012 yılında Avrupa Telekomünikasyon Standartları Enstitüsü (ETSI) tarafından ilk olarak önerilen NFV, özel donanım uygulamalarını sanallaştırılmış bir ortamda çalışan yazılım örnekleriyle değiştirme yeteneğiyle dikkat çekici bir çözüm olarak ortaya çıkmıştır (S. Yang et al., 2021). Ağ fonksiyonlarının sanallaştırılması, donanım ve yazılımın birbirinden ayrılmasını sağlayan ve ağ hizmetlerinin bulut bilişim platformları üzerinde çalışmasına olanak tanıyan bir kavramdır. Sanallaştırma teknolojileri, ağ altyapısı için getirdiği yeniliklerle yeni ortaya çıkan hizmetlerin daha az riskle daha hızlı devreye alınmasına olanak tanırken, var olan hizmetlerin iyileştirilmesini destekler (Joshi and Benson, 2016).

Diğer bir yandan, fiziksel bilgi işlem kaynaklarını soyutlamaya olanak tanıyarak verimlilik ve esneklik sağlar ve VNF' lerin yaşam döngüsü boyunca kolayca yönetilmesine olanak tanımaktadır. Geleneksel ağ mimarilerine kıyasla, Ağ Hizmetleri Sanallaştırma kavramı donanım fiyatlarının azaltılması, performansın iyileştirilmesi, ağ konfigürasyonu ve yer ayırma işlemlerinin iyileştirilmesi, ağ fonksiyonlarının devreye alınmasında esneklik sağlanması ve enerji verimliliği sağlanması gibi özellikleriyle daha avantajlıdır (Yang and Fung, 2016). Günümüz ve gelecek iletişim ağları; bulut hizmet sağlayıcıları, kurumsal ağlar, içerik teslim ağları, mobil kullanıcılar gibi birçok farklı aktöre sahip olup, daha fazla performans, güvenilirlik ve güvenlik talep etmektedir (Cotroneo et al., 2014). Modern ağlar, bu gibi ihtiyaçları karşılamak için güvenlik duvarı, saldırı tespit ve önleme sistemleri, trafik yönlendiricileri gibi birçok hizmeti bir arada sunan middlebox olarak adlandırılan ağ cihazlarını kullanırlar. Bu ağ cihazları genellikle bazı özel fonksiyonları yerine getirmek üzere eğitilirler.

Ağ fonksiyonlarının Sanallaştırılması kavramı, donanımların üzerine yüklenen fonksiyonların sanal ortamlara, zaten var olan veri merkezlerine ya da ağ düğümlerine yazılım çözümleri olarak taşınmasını öneren bir yaklaşımdır (Cotroneo et al., 2014). Dolayısıyla, ağ hizmetlerinin yazılımlaştırılması kavramı bir dizi yazılım fonksiyonunun zinciri olarak tanımlanan Hizmet Zincirleri

Fonksiyonu kavramını ortaya çıkarmaktadır. Hizmet zincirleri fonksiyonu da hizmetler için veri merkezlerinde yer ayrılması olarak tanımlanan hizmet yerleştirme kavramını ortaya çıkarmaktadır (Karimzadeh-Farshbafan et al., 2020). Yazılım Tanımlı Ağ ve Ağ Fonksiyonlarının Sanallaştırılması kavramları birbirlerini tamamlayıcı olarak kullanılan kavramlardır. YTA, Sanal Ağ Fonksiyonları (VNF) arasında programlanabilir yönetim ve bağlantı sağlayarak Ağ Fonksiyonlarının Sanallaştırılmasına hizmet edebilir. Benzer şekilde Ağ Fonksiyonlarının Sanallaştırılması, ağ işlevlerini (NFs) veya hatta bir Sanal Ağ Fonksiyonu (Sanal-YTA denetleyici) olarak çalıştırılabilecek denetleyiciyi sanallaştırarak YTA' ya hizmet edebilir (Ben Azzouz and Jamaï, 2019). Ağlar sadece anahtarlar ve yönlendiricilerden oluşmaz, aynı zamanda önbellekler, derin paket incelemesi, güvenlik duvarları vb. sağlayan ara bileşenlerden de oluşur ki bunlar YTA denetleyici için karmaşık görevlerdir. Bu nedenle, NFV kavramı, ağ işlevlerini altında yatan özel donanımdan ayırarak onları yazılım formunda gerçekleştirme fikrini temsil eder. Bu yazılım formuna Sanal Ağ Fonksiyonları (VNFs) denir (Yi et al., 2018). Bu yaklaşım, ağın daha esnek maliyetli, etkin ve kolay yönetilebilir olmasına yardımcı olmaktadır.

### 2.3. Mininet

Mininet, YTA ortamı için özel olarak tasarlanmış sanal bir ağ emülatörüdür ve ağ elemanlarını sanallaştırma yeteneğine sahiptir. Tek bir Linux Çekirdeği içerisinde bir dizi anahtar, son kullanıcı bilgisayarı ve bağlantı barındırmaktadır (Lantz and O'Connor, 2015). Mininet' in bu kabiliyeti, tek bir sistemin tüm bir ağı sanal olarak kopyalamasını mümkün kılmaktadır. Mininet içerisindeki her ana bilgisayar, gerçek bir ana bilgisayarın işlemlerini taklit eder. Bu da SSH bağlantıları kurma ve temel Linux işletim sisteminden uygulamalara erişme gibi işlevselliği mümkün kılmaktadır. Bu sanal varlıkların, ister ana bilgisayarlar, anahtarlar veya bağlantılar olsun, gerçek dünya muadillerini taklit eden farklı Ethernet arayüzleri vardır. Gerçekleştirilen tez çalışmasında, Mininet önemli bir rol üstlenmektedir, YTA denetleyicisiyle bağlantılı belirli ağ topolojileri oluşturma imkânı tanımaktadır. Ayrıca, Mininet, araştırmacılara özel topoloji dosyaları oluşturarak özel ağ topolojileri tasarlama yeteneği sunmaktadır.

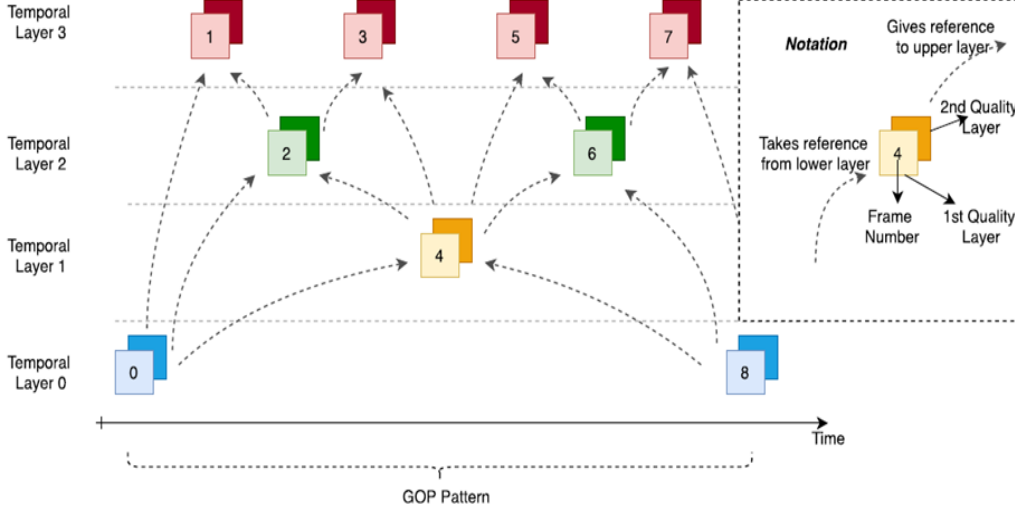
### 2.4. Ölçeklenebilir Video Kodlama (SVC)

Video kodlama, ham video verisinin sıkıştırma yapılarak saklama ve ağ üzerinden iletme hazır boyutlara indirgenmesi amacıyla gerçekleştirilmektedir. Sıkıştırılan video, farklı tipte karelerden oluşmaktadır. I tipi kareler kendi başına

çözümünebilirken, P ve B tipi kareler, I tipi karelerden ya da birbirlerinden referans alabilirler, çok daha düşük boyutlu olsalar da çözümlenmeleri için referans karelere ihtiyaç duyarlar. Bir video dizisini farklı kalitelerde kodlayabilmek için ayrı ayrı farklı kodlama parametreleriyle birden çok kodlanma yapılabilmektedir. Bu yöntemle birden fazla kodlanmış video dosyası oluşmaktadır. SVC (Ölçeklendirilebilir Video Kodlama) ile kodlamada ise, kodlanmış tek bir video dosyası içerisinde farklı kalitelerde video oluşturulabilmektedir. SVC ile kodlanan video, bir taban katman ve birden fazla genişleme katmanından oluşmaktadır. Taban katman, videonun çözülebilmesi için gereklidir, bu katman üzerine eklenen her genişleme katmanı video kalitesini uzaysal (spatial), zamansal (temporal) ya da kalite açısından artırmaktadır. Başka bir deyişle, her genişleme katmanı videonun çözünürlüğünü, saniyede oynatılan kare sayısını ya da PSNR (Peak Signal-to-Noise Ratio) değerini arttırabilmektedir. ITU-T VCEG ve MPEG gruplarının ortak olarak 2020 Temmuz ayında çalışmasını tamamladığı H.266 standardının SVC uzantısının, H.266 ile elde edilen kodlama performansı sayesinde (H.265'e oranla %50 daha az bit kullanarak aynı kalite elde edilebilme, kodlama 9 kat, çözümlenme 1,6 kat daha büyük hızla tamamlanabilmektedir) yakın gelecekte ticari sistemlerde oldukça yaygınlaşacağı raporlanmıştır (Sullivan, 2020).

SVC kodlamanın avantajları ve yakın zamanda yaygınlaşacağı öngörüsü ile tez kapsamında kullanacağımız videolar SVC ile kodlanmıştır. Ayrıca BPP üzerinden video gönderimi SVC ile oldukça uyumludur. SVC' nin bir başka avantajı ise, daha önceden tampona alınan video karelerinin kalitelerinin, daha sonra, daha üst katmanlarının alınmasıyla ağ kaynaklarını boşa harcamadan ve yeniden gönderime gerek kalmadan arttırılabilesidir (Ozcan vd., 2017). Tez çalışması kapsamında ayrıca, SVC' nin bu özelliğinden yararlanılmıştır. Geleneksel bir kodek ile aynı video dosyasının değişik kalitelerdeki versiyonlarını elde etmek amacıyla video birkaç kez kodlanır. Bu durum, her kalite seviyesi için ayrı bir video dosyası sonucunu doğurur. Ancak SVC, bu duruma bir alternatif sunmaktadır (Schwarz et al., 2007). SVC kullanarak, tek bir video dosyasından çeşitli kalite seçenekleri elde edilebilmektedir. Bu yöntemde, bir video bir temel katman ve birkaç ilave katman şeklinde kodlanmaktadır. Bu temel katman, minimum video kalitesini sunarken, her ilave katman ondan önceki katmana bağlıdır. SVC, cihaz özelliklerine ve ağ koşullarına dayalı olarak dinamik uyarlama sağlar. SVC; depolama, önbellek performansı ve uyarlama açısından avantajlar sağlamaktadır (Grafl et al., 2013). SVC, Gelişmiş Video Kodlama (AVC)' nin bir uzantısını temsil eder ve bir video segmentinin, bir temel katman ve çok sayıda geliştirme katmanı dahil olmak üzere N katmana kodlanmasına izin verir. Daha

fazla katmanın birikmesi genellikle video kalitesinin artmasına yol açmaktadır. (İbrahim vd., 2014).



Şekil 2.4 SVC Video Dosyası Yapısı (Clayman and Sayıt., 2023)

SVC’ de geçici, uzamsal ve kalite katmanları olmak üzere farklı kalite katman türleri bulunmaktadır. Her türlü çerçeve, Makro Bloklardan oluşan Blok Grupları (GOP) ile temsil edilir. Şekil 2.4, bir videonun çerçeve yapısını ve aralarındaki bağımlılıkları göstermektedir. Her bir çerçeve, 1 saniyelik video için 8 çerçeveden oluşan bir Resim Grubu (GOP) içerisinde yer almaktadır. En düşük zamansal katmanda yalnızca I çerçeveleri bulunmaktadır ve bu I çerçeveleri diğer çerçeveleri referans almazlar. Şekil 2.4’ te, SNR’ ı geliştiren kalite katmanları da gösterilmektedir; fakat bunlar çerçeve çözünürlüğü üzerinde bir etkiye sahip değildir. Gösterilmeyen bir diğer katman türü ise uzamsal katmandır; bu katman videonun çözünürlüğünü iyileştirmektedir. SVC’ nin video yayınında avantajlar sağladığı bilinmektedir (Yang et al., 2011), ve bazı uzmanlar bu avantajları maksimize etmek için DASH ile SVC’ yi birleştirmiştir (Grafl et al., 2013; İbrahim vd., 2014). Bu tez çalışmasında, SVC’ nin katmanlı yapısından yararlanılmakta ve BPP’ nin yetenekleriyle bütünleştirilmektedir.

## 2.5. Büyük Paket Protokolü (BPP)

Büyük Paket Protokolü (BPP) yüksek verimli uygulamaların ve ağ mimarilerinin ihtiyaçlarını karşılamak için tasarlanmış yeni bir ağ protokolüdür (Li et al., 2018). 2030’da beklenen gelecek ağların bir parçası olarak düşünülmektedir. BPP’ nin öncelikli hedeflerinden biri, yüksek bant genişliğine sahip video, AR veya

VR gibi uygulamaları içeren yüksek hassasiyetli hizmetlerin ağ gereksinimlerini karşılamaktır. BPP, bu hedeflere ulaşmak için bir ağ düğümünün, özellikle iletim elemanlarının, akışın ve bireysel paketin seviyesinde uygulamaya özel bir davranışı tanımlamasına ve uygulamasına izin vermektedir. Bu tür bir işlevselliği desteklemek için, her BPP paketi bir başlık ve bir dizi veri parçasına bölünmektedir. Bu parçaların bazıları, başlığa enjekte edilen içerik meta verilerine ve ağın yüküne bağlı olarak iletim sırasında ağ düğümü tarafından atılabilir (Li et al., 2019). BPP ile sunulan yaklaşım, bir paketin tamamını atmamak, ancak sadece bazı parçalarını atmaktır. Paket boyutunun uyarlanması, ağ koşullarını ve herhangi bir uygulama geri bildirimini göz önünde bulundurarak gerçekleştirilir. BPP ile geliştirilen strateji, alıcıya sürekli gelen paket akışını korurken, tüketilen bant genişliğini azaltarak ağ üzerindeki yükü azaltmaktır. BPP, TCP' den çok UDP' ye benzeyen, ancak kısmi güvenilirlik sağlayan yeni bir taşıma katmanı protokolü türü olarak tasarlanmıştır. BPP, yeniden iletme sahip olmanın yeniden gönderme sürelerini çok yavaşlattığı düşük gecikme süresi ve yüksek güvenilirlik uygulamaları için kullanılmak üzere tasarlanmıştır (Li et al., 2018). Etkinliğini belirlemek için bir dizi çalışma da değerlendirilmiştir. BPP hakkında detaylı bilgilendirme tezin bir sonraki kısmında gerçekleştirilecek ve oluşturulan paket yapısı açıklanacaktır.

## 2.6. Literatür Taraması

Tez çalışmasıyla ilgili yapılan literatür taramasında ilk olarak BPP ile yapılmış olan tüm çalışmalardan bahsedilmektedir. Sonrasında YTA ortamında video akışlandırma amaçlı yapılmış olan çalışmalar ve video akışlandırma için kullanılan protokollerle ilgili çalışmalardan bahsedilmektedir.

BPP yakın zamanda önerilen bir protokol olsa dahi literatürde BPP' nin sağladığı avantajları kullanarak gerçekleştirilen çalışmalar bulunmaktadır. (Clemm and Eckert, 2020) çalışmasında BPP kullanarak gecikme sınırı belirlenen paketler için iletimin bu sınırlara uyacak şekilde yapılmasını sağlayan bir yaklaşım önermişlerdir. Çalışmada, gecikme sınırları, BPP alanları içerisine uygulama tarafından yazılmakta, anahtarlama cihazı bu sınırlara uyum sağlayamayacak paketleri tahminleyerek düşürmektedir. (Handley et al., 2017) çalışmasında ise Veri Merkezi için kırpma konusu üzerinde çalışılmıştır. Bu çalışmada, çok çeşitli senaryolarda kısa aktarımlar ve yüksek akış verimi için optimuma yakın tamamlanma sürelerine ulaşan yeni bir veri merkezi aktarımı olan NDP tanımlanmaktadır. NDP ile, anahtarlar paketleri yalnızca üstbilgilere göre kırpar ve ardından üstbilgileri iletmeye öncelik verir. Yazarlar, "Paket kırpma nedeniyle, bir

paketin gerçekten kaybolmasının çok nadir olduğunu" gözlemlemişlerdir. Ağ üzerindeki gidip gelme sürelerini iyileştirmek için (Westphal et al.,2019)' da yazarlar, yükün tamamını kesip sadece başlığı korumanın, yükü bırakan düğüme yükün yeterince hızlı yeniden iletilmesi durumunda Veri Merkezlerinde iyi çalışabileceğini öne sürmektedirler. Böyle bir veri merkezi yaklaşımının WAN' lara genellenemeyeceği vurgulanmıştır. Paketin tamamını değil de seçici bir şekilde kesmek, geniş alan ağında çalışacak daha verimli bir mekanizma sunacaktır. Uçtan uca taşımayı geliştirmek için, yük paketinin parçalarını seçici olarak bırakarak tıkanıklığa tepki veren ve yükün bir kısmının kaybını işlemek için hafifletme mekanizmalarıyla birlikte yeni bir taşıma protokolü QUCO tanımlanmıştır (Albalawi et al., 2020). Paket kırpma şeması, ağdan geçen paket sayısındaki değişimi azaltır ve iletimdeki paket sayısı ve ayrıca anahtar arabelleklerinin boyutu üzerinde daha sıkı hedeflerin belirlenmesine izin verir. QUCO, TCP' ye göre daha az gecikmeye ve daha az gecikme varyasyonuna sahiptir ve bunun sonucunda jitterde azalma olur. (Adrian et al., 2022) çalışmasında, paket kırpma için bir donanım mekanizmasının uygulanmasına yönelik özel olarak tasarlanmamış programlanabilir anahtarlarda kırpmanın nasıl uygulanabileceği araştırılmıştır. Bu uygulamada Tofino anahtarı ASIC üzerinde P4' ü kullanır ve kırpmanın gerçekleştirilebileceğini gösterir ve kırpmanın üretim sınıfı bir veri merkezi anahtarına entegre edilebileceğini gösterir. (Makhijani et al., 2019) çalışmasında, Zamana Duyarlı Ağ İletişiminin (TSN) ölçeklenebilirlik ve karmaşıklık sorunlarından kaynaklanan sınırlamaların üstesinden gelmek için BPP TSN ile birleştirilmek üzere kullanılmıştır. (Dong and Li, 2019b) çalışmasında, Mobil Uç Ağlarda hesaplama boşaltma için BPP kullanılmıştır. Bu makalelerin hiçbiri BPP üzerinden multimedya üzerinde durmamaktadır.

(Li et al., 2019) çalışmasında BPP' nin paket parçalarının iletim esnasında silinebilme özelliğinin, video uygulamaları için avantajlı olacağı vurgulanmıştır. Ancak bu konuda bir çalışma önerilmemiştir. BPP' nin video akışlandırma uygulamalarında, HAS (HTTP Adaptif Akışlandırma) benzeri yaklaşımla makroblok seviyesinde farklı kalitelere kodlanmış video parçaları iletildiği durumda avantajlı olabileceğine dair bir çalışma (Clayman, 2019) tarafından önerilmiş, ancak olası uygulama ayrıntıları düşünülmemiştir. BPP medya iletimi için tasarlanmış olsa da önceki çalışmalarımız olan (Clayman et al., 2021a; 2021b), BPP kullanarak video akışının ilk gerçek uygulaması olarak literatüre geçmiş ve temel olarak BPP üzerinden video göndermenin etkilerinin araştırıldığı çalışmalar olduğu belirtilmiştir. Bu iki çalışmada, gecikme ve kesintiler açısından BPP' nin TCP ve UDP' den daha iyi olduğu vurgulanmış ve BPP' nin özellikle bazı yüksek

hassasiyet gereksinimleri dikkate alındığında video akışı için makul bir alternatif olduğu gösterilmiştir. (Clayman and Sayit, 2021) çalışmasında, BPP ile birlikte SVC kullanıldığında ağ içi uyarlamamanın sağlanabileceği gösterilmiştir. BPP Paket Küçültme mekanizmasını kullanarak SVC video akışlarını sunuculardan istemcilere taşımak için kullanılan tekniklerin ve mekanizmaların ayrıntılı bir açıklaması (Clayman and Sayit, 2023) çalışmasında sunulmuştur. BPP tanıyan cihaz olmadığı için BPP yükleri UDP üzerinden gönderilmektedir.

Video iletimi genellikle iki yoldan gerçekleşir. RTP üzerinden UDP, hızlıdır ancak paket kaybına neden olabilir ve HTTP üzerinden TCP, güvenilirdir ancak gecikmeye neden olabilir (Yang et al., 2011). Medya verilerinin iletimini desteklemek için, ses ve video taşımak için bir dizi protokol tanımlanmıştır. RTP/RTSP protokolleri, etkileşimli ortam akışı varlıklarının yük tipi, kayıp paket bilgileri, zaman damgaları ve sıra numaraları gibi bilgileri paylaşmasına izin veren bir mekanizma sağlar (Crowcroft, 1999). RTP (Schulzrinne et al., 2003), medya akışlarını bir UDP aktarımıyla taşımak için tasarlanmıştır ve yaklaşımı Uygulama Katmanı Çerçevelemesine (ALF) (Clark and Tennenhouse., 1990) dayanan bir protokoldür. RTP paketlerinin görece zaman damgaları vardır ve hem senkronizasyon için kullanılabilen NTP zaman damgasına göre eşlemesi olan gönderici raporlarını hem de paket kaybı/jitteri gösteren alıcı raporlarını destekler. RTSP, RTP oturumları için bant dışı istatistikler ve kontrol bilgileri sağlamak üzere tasarlanmıştır. RTSP (Rao et al., 1998), RTP kullanan ortam teslimi için kullanılan durum bilgisi olan bir protokoldür. WebRTC 500 milisaniyenin altında gecikmeyle bir web tarayıcısı aracılığıyla Gerçek Zamanlı İletişim sağlayan standartların, protokollerin ve JavaScript API'lerinin bir kombinasyonudur (WebRTC., 2023). WebRTC, değiştirilen verilerin güvenli ve doğrulanmış olduğundan emin olmak için SRTP'yi (Güvenli Gerçek Zamanlı Aktarım Protokolü) kullanır. HTTP protokolü (Belshe et al., 2015), TCP üzerinde çalışır ve İçerik Sağlayıcı Ağlar (CDN) tarafından kullanılanlar gibi yaygın olarak video iletimi içindir, günümüzün fiili akış teknolojisi olan HTTP Uyarlanabilir Video Akışını (HAS) içerir. HTTP Üzerinden Dinamik Uyarlanabilir Akış (DASH) (MPEG., 2023) HTTP Uyarlanabilir Video Akışı (HAS) dağıtımlarının katılımcıları arasında birlikte çalışabilirlik sağlamak için MPEG tarafından geliştirilmiş bir standarttır. DASH, HTTP'nin üstünde çalışır, bu nedenle TCP tabanlıdır. DASH'ın yaklaşımı, videoları segmentlere ayırmak ve ağ koşullarına göre adaptif olarak akıtmaktır. Bunun TCP ile kullanılması, gecikmeyi etkileyebilir. Bu gecikme sorununu iyileştirmek için çalışmalar devam etmektedir (Bitmovin., 2023; Lim et al., 2020; Taraghi et al., 2023). Ağda tıkanıklık olduğunda video dağıtımını optimize etmek

için değerlendirilen başka bir yaklaşım, kullanıcıya yakın video kod çevrimidir. (Erfanian et al., 2021) çalışmasında yazarlar, uçta hafif kod dönüştürmeyi etkinleştirerek HTTP Uyarlamalı Akışta video akış maliyetlerini azaltan bir sistem sunarlar. Genel olarak, HTTP üzerinden ortam akışı için (Bentaleb et al., 2019)'te incelenen bir dizi bit hızı uyarlama şeması vardır. QUIC (Hızlı UDP İnternet Bağlantıları) aktarım protokolü (Bishop, 2022), web hizmetlerinin QoE' sini iyileştirmek için tanıtılmış ve HTTP protokolünün TCP yerine UDP üzerinden çalışması sağlanmıştır. QUIC, video iletimi için iyi bir aday olarak önerilmiş olsa da (Seufert et al., 2019)' da yazarlar YouTube akışını değerlendirmişler ve TCP üzerinden HTTP ile karşılaştırıldığında QUIC kullanılırken bir QoE artışı olduğuna dair hiçbir kanıt olmadığı sonucuna varmışlardır. (Palmer et al., 2018)' de, QUIC ile gerçek zamanlı video trafiği değerlendirilmiş ve QUIC' in hala çok güvenilir olduğu, bazen video akışı için TCP'den daha kötü performans gösterdiği ve QUIC ile kullanılan ABR şemalarının TCP' ye kıyasla zayıf çalıştığı gözlemlenmiştir. Güvenilir aktarımların video akışı için uygun olmadığı ve güvenilir olmayan bir QUIC sürümü önerdikleri sonucuna varılmıştır. Hem SVC hem de QUIC' in birlikte kullanımını (Nguyen et al., 2020) çalışmasında değerlendirilmiştir. Yazarlar, HTTP 'nin Hat Başlı Engelleme ve TCP nedeniyle üç yönlü anlaşma gecikmesinden muzdarip olduğunu ve UDP üzerinden çalışan QUIC' in bu sorunları çözebileceğini belirtmişlerdir. Hızlı kurtarma ve Hat Başlı Engelleme yeteneklerinin ortadan kaldırılmasıyla QUIC' in, paket kaybının olduğu yerlerde daha iyi performans sağlamak için birden fazla akışı destekleyebileceğini belirlemişlerdir.

(Kleinrouweler et al., 2016) çalışmasında, YTA paradigması temel alınarak bir ağ mimarisi tanıtılmıştır. Bu mimari, DASH oynatıcılarına hedef bit hızları sinyalleme ve ağ içerisinde dinamik trafik kontrolü sağlamak için iki adaptasyon mekanizması sunar. Sonuç olarak, bu YTA tabanlı mimari, video bit hızını iki katına çıkararak ve kalite değişikliklerini azaltarak izleme deneyiminin kalitesini artırmaktadır. (Farahani et al., 2021b) çalışmasında, YTA ve NFV yeteneklerini kullanarak kenar ve YTA destekli bir video akış çerçevesi olan ES-HAS' ı önermişlerdir. ES-HAS, sanallaştırılmış kenar bileşenlerini kullanarak, HAS istemcilerinin taleplerini toplayabilmekte ve ağ bilgilerini zaman dilimli bir şekilde getirebilmektedir. Kenar Hesaplama, ek hesaplama, veri işleme ve ağ kaynaklarının uç sistemlere daha yakın yerleştirildiği bulut bilişimin bir uzantısı olarak kabul edilir. Sonuç olarak, yalnızca merkezi bulut sunucularında değil, uç sistemler ve bulut sunucuları arasında veri işleme, ağ oluşturma, veri yönetimi ve depolama süreçleri gerçekleştirilebilir. Kenar Hesaplama, düşük gecikmeli uygulamalar ve ayrıca

istemcilerde yakınlık veya bant genişliđi veya zaman sınırlamaları nedeniyle bulut sunucularına gerçek zamanlı olarak pratik olarak aktarılamayan çok büyük miktarda veri üreten uygulamalar için son derece yararlı olabilir (Cao et al., 2020). Son zamanlarda, merkezi uygulamalar, sanallaştırılan ve özellikle uçta otonom olarak yürütölen çok özel görevlerle ilgilenen küçük, bağımsız ve gevşek bağı sistemlerle hizmet odaklı mimarilere ve mikro hizmetlere doğru evrilmiştir.

(Mu et al., 2016) çalışmasında, YTA tabanlı 5G kablosuz ağlar için tampon farkındalıklı bir HTTP canlı akış yaklaşımı önerilmiştir. Bu yaklaşım, şu anki oynatma süresine dayalı olarak medya segmentlerinin ağırlıklarını değerlendirir ve YTA' daki yönlendiricilerin ve anahtarların kullanımını ve istikrarına göre uygun iletim yolu kararını alır. (Al-Issa et al., 2019) çalışmasında yazarlar, zaman serisi tahmini gücünden faydalanan ve ağ tabanlı rehberlik için doğru ve ölçeklenebilir bir yönlendirme sağlayan zeki bir yayın mimarisi olan BBGDASH+' ı önermektedirler. Ayrıca, kablosuz bir test ortamında tahmin algoritmalarının parametre ayarlarının ilk incelemesini gerçekleştirmişlerdir. BBGDASH+, gerçek zamanlı bant genişliđi tahmini sağlayan QoE odaklı bir ağ destekli mimari olarak sunulmaktadır. (Farahani et al., 2021a) çalışmasında, YTA, NFV ve kenar hesaplama yeteneklerini kullanarak, YTA Destekli Adaptif Video Yayımında CDN-Farkında QoE Optimizasyonu (CSDN) sunulmaktadır. CSDN, istemcilerin isteklerini optimal önbellek sunucularını seçerek hizmete sunmak için sanal kenar varlıklarını kullanarak çeşitli bilgi öğelerini toplar ve yeni bir sunucu/segment seçim yaklaşımıyla zaman dilimli bir optimizasyon modeli çalıştırmaktadır. HTTP Adaptif Akışlandırma (HAS) teknolojisinin öne çıkmasına paralel olarak, kullanıcının QoE ve ağ kullanımını artırmak için YTA, NFV ve kenar hesaplama gibi modern ağ paradigmasını kullanan ağ destekli video akış şemaları tanıtılmıştır. Ancak mevcut ağ destekli HAS şemaları, kenar işbirliđi tekniklerini ve YTA yeteneklerini tam anlamıyla kullanmamıştır. Bu boşluğu doldurmak amacıyla (Farahani et al., 2022) çalışmasında, HTTP uyarlamalı video akışı için İşbirlikçi Kenar ve YTA Destekli bir çerçeve olan LEADER tanıtılmaktadır. LEADER, merkezi bir optimizasyon modeli çalıştırarak HAS istemcilerinin hizmet süresini en aza indirmeyi hedeflerken, ağın ve kenar sunucuların kaynak sınırlamalarına tabi olmaktadır.

(Binsahaq et al., 2019) çalışmasında, otomatik bilgi işlem (AC) kavramı uygulayarak QoS gereksinimlerini otomatik olarak desteklemek, sağlamak, izlemek ve sürdürmek amacıyla YTA alanındaki QoS provizyonu üzerine mevcut çalışmalar derinlemesine incelenmektedir. Bu incelemeye göre, otomatik temelli

QoS provizyonu için sınıflandırma, taksonomi ve karşılaştırmalı analiz elde edilmiştir. Ancak, literatürde otomatik temelli QoS provizyonunu daha da geliştirebilecek birçok araştırma alanı bulunmaktadır. (Nam et al., 2014) çalışmasında önerilen YTA uygulaması, akışın ağ koşullarını gerçek zamanlı olarak izlemek ve çok protokollü etiket anahtarlama (MPLS) trafik mühendisliği (TE) kullanarak yönlendirme yollarını dinamik olarak değiştirmek için tasarlanmıştır. Video dağıtım mimarisinde, bir dağıtım düğümü bir istemciye bağlandığında kullanıcının algıladığı video kalitesini takip etmek zordur. İstemci tarafında sadece çözünürlüklerin değiştirilmesi, video QoE' sini düşüren darboğaz sorunlarını çözmeyebilir. Bu çalışmada, YTA üzerinde MPLS-TE' yi kullanarak video QoE' sinin iyileştirilmesi amaçlanmıştır. (Ahmad et al., 2020) çalışmasında, video akışı için yeni QoE yönetimi ve izleme prosedürlerinin tasarımı ve değerlendirmesi için bir YTA tabanlı emülasyon platformu olan Timber tanıtılmıştır. Timber' ın ana işlevleri arasında; QoE farkındalıklı yönetim kararları alabilen bir YTA uygulaması, ağ dilimleme ve çok protokollü etiket anahtarlama (MPLS) tabanlı önceliklendirme işlemleri gibi ağ yönetimi eylemlerini uygulamak için ağın hizmet kalitesini izleyen bir YTA denetleyicisi ve bir multimedya sunucusu ile DASH tabanlı bir istemci video oynatıcısını içeren tam bir video akışı uygulaması bulunmaktadır. (Georgopoulos et al., 2014) çalışmasında, son kullanıcıya olabildiğince yakın video içeriği önbelleklemeyle talep üzerine video (VoD) dağıtım verimliliğini artırmayı amaçlayan, yüksek derecede yapılandırılabilir, verimli ve şeffaf bir ağ içi önbellekleme hizmeti olan OpenCache sunulmuştur. OpenCache, YTA' yı kullanarak son kullanıcı için kalite deneyimini artırıp avantaj sağlamaktadır. (Ben Letaifa, 2017) çalışmasında, internet üzerinden veri hizmetlerinin kullanıcı algısı ve tatminini değerlendirmek için QoE metriklerine odaklanılmıştır. Bu çalışmada, bir yandan geleneksel olarak ağ optimizasyonu, QoS gibi ağ özellikleri optimize edilmeye çalışılırken, bir yandan da uçtan uca QoE metrikleri optimize edilmeye çalışılmış ve bu durum gerçek zamanlı olarak takip edilmiştir. (Yu et al., 2015) çalışmasında, ARVS adında QoS destekli YTA üzerinde video akışı için adaptif bir yönlendirme yaklaşımı önerilmiştir. Önerilen yaklaşım, video bit akışlarının temel katman ve artırma katmanı paketlerini iki farklı QoS akışı (seviye-1 ve seviye-2 QoS akışları) olarak ayrı ayrı ele alır.

(Park and Song, 2019) çalışmasında, YTA destekli 5G ağlar üzerinde video yayın hizmetleri için kooperatif bir baz istasyonu önbelleği ve X2 bağlantı trafiği aktarma sistemi sunulmuştur. Önerilen sistem, içerikleri etkili bir şekilde baz istasyonlarında önbelleğe alarak ve ana taşıyıcı trafiğinin bir kısmını X2 bağlantılarına yönlendirerek hücresele ağıdaki ana taşıyıcı trafiğini hafifletmeyi

amaçlamaktadır. (Shen, 2019) çalışmasında, TCAM alanını YTA anahtarlarında dikkate almayan önceki çalışmalardan farklı olarak, adaptif YTA tabanlı SVC çoklu yayın (ASCast) adlı yeni bir SVC çoklu yayın akış şeması önerilmektedir. ASCast, video katmanlarının önemine ve ağ bilgisine dayanarak farklı video katmanlarının yönlendirmesini kontrol eden yenilikçi bir SVC çoklu yayın akış şemasını sunmaktadır. (Zacarias et al., 2017) çalışmasında, YTA prensipleri kullanılarak, çoklu İnsansız Hava Araçlarının bir YTA etkin ağ aracılığıyla veri sağlayıcıları olarak kullanıldığı askeri gözetim için video akışının bir analizi sunulmaktadır. Sonuçlar, YTA' nın heterojen ve fırsatçı bağlantılara sahip ağlarda başarıyla uygulanabileceğini göstermektedir. (Henni et al., 2020) çalışmasında, YTA ağında alınacak kararların tutarlılığını garantileyen bir ağ mimarisi önerilmektedir. Önerilen mimari, öncelikli trafiğin kalite düşüşünü engellerken kaynak kullanımını optimize eden tutarlı bir QoS yönlendirme stratejisi sunmaktadır. Bu amacı gerçekleştirmek için trafik dağıtımını üzerine bir sezgisel yaklaşım sunulmuştur. Önerilen yaklaşımın etkinliği, Mininet ortamında yapılan emülasyon sonuçlarıyla mevcut çerçevelere göre üstün olduğu gösterilmiştir. Özetle, bu çalışma, video akışı kalitesini ve en iyi çaba ile aktarımını artırmak için YTA da QoS yönlendirmesi için tutarlı bir ağ mimarisi sunmaktadır.

Literatür taraması kısmında video akışlandırma için kullanılan yöntemler ve protokollerle ilgili çalışmalardan bahsedilmektedir. Mevcut çalışmalar incelendiğinde BPP ile ilgili tanımlamalar yer almaktadır fakat herhangi bir uygulama bulunmamaktadır. Önceki çalışmalarımız arasında yer alan (Clayman et al., 2021a; 2021b) çalışmaları, BPP kullanarak video akışının ilk gerçek uygulaması olarak literatüre geçmiş ve temel olarak BPP üzerinden video göndermenin etkileri araştırılmıştır. Tez çalışması kapsamında literatürde yer alan çalışmalardan farklı olarak yazılım tanımlı ağ ortamında sanallaştırılmış ağ işlevlerini kullanarak video akışı uygulamalarının performansını artırma amacıyla BPP kullanımı gerçekleştirilmiştir. Literatürde daha önce çalışılmamış bir konu olan kenar ağda sanallaştırılmış BPP fonksiyonu kullanılmasını gerçekleştiren bir mimari sunulmaktadır. Sunulan mimari detaylı olarak açıklanmakta ve daha önceki yöntemlerle karşılaştırılarak deneysel sonuçlar sunulmaktadır.

### 3. BPP PAKET YAPISI VE PAKET STRATEJİLERİ

Video akış uygulamalarının BPP' den faydalanabilmesi için video içeriğinin uygun şekilde hazırlanması ve kodlanması gerekmektedir. Gönderen taraftaki paket hazırlama yöntemi, BPP iletim sürecinden haberdar olmalı ve iletim sırasında bazı BPP parçalarının çıkarılabilmesi için paket bu duruma uygun bir şekilde oluşturulmalıdır. Bu bölümde BPP paket yapısının geliştirilmesi için yapılan çalışmalar ayrıntılı bir şekilde açıklanmıştır. İlk olarak literatürde önerilen BPP paket yapısındaki alanlar ve özellikleri tanımlanmıştır. Sonrasında video akışlandırma için BPP'ye eklenen alanlar açıklanmıştır. Son olarak BPP iletimi, için oluşturulan farklı paketleme stratejileri açıklanmıştır.

#### 3.1. Video İletimi İçin BPP' de Yapılan Geliştirmeler

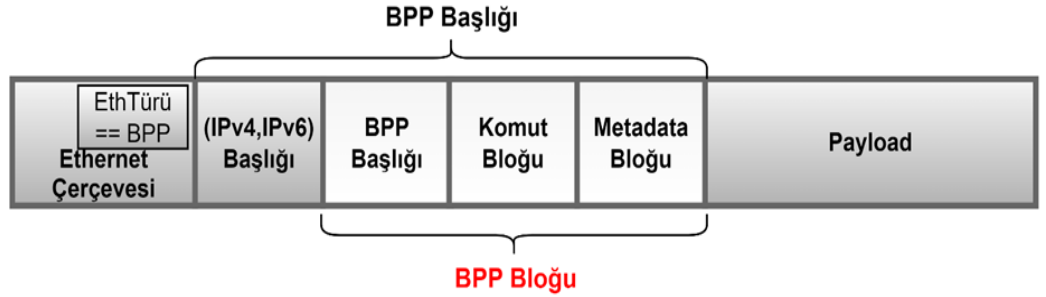
BPP' nin parça atma süreci, verinin paketlere konulup alıcıya ağ üzerinden gönderildiği akış uygulamalarına avantajlar sağlamaktadır. Sadece parçaların kaldırılmasına dayalı bu süreç, alıcıya düzenli olarak paketlerin ulaştığı sürekli bir teslimat sağlamaktadır. Genel olarak, video oynatıcılara videonun tamamının ulaşması gerekir, ancak birçok ağ video alıcısı, iletim sırasında bazı video karelerinin kaybedilmesi durumunda bile devam edebilmesi için uyarlanabilir olarak tasarlanmıştır. Bu eksik kareler, hangi karenin kaybolduğuna bağlı olarak kullanıcıya farklı kalite azalmaları olarak yansımaktadır. Buna bağlı olarak, bir YTA denetleyicisinin BPP paket yapısını ve paket değiştirme yeteneklerini kullanarak gelişmiş uçtan uca video iletimi sağlayabileceği söylenebilir.

BPP' yi video için etkili bir şekilde kullanmak, YTA denetleyicisine uygun paket değişiklikleri yapma yeteneği sağlamak ve alıcının en yüksek kaliteli akışı almasını sağlamak için video kodlayıcı ve dekodlayıcının seçimi önemlidir. Seçime göre, her kare için birden fazla kodlama sağlanabilir ve video akışı, video dosyasının bir temel katmanı ve bir dizi geliştirme katmanı içerecek şekilde kodlanabilir. Temel katman en düşük video kalitesini sağlarken, diğer herhangi bir katmana kod çözme gereksinimi duymaz; geliştirme katmanları, kodlanabilir olmaları için altındaki katmanlara bağımlıdır. Bu katman bağımlılıkları, BPP' nin paket değişikliğiyle uyumludur ve YTA denetleyicisine videonun katmanlı yapısını bilerek iletim sırasında bazı parçaları silme olanağı tanır. Ağ bant genişliği sınırlı hale gelirse, YTA denetleyicisi geliştirme katmanlarının bazı parçalarını silebilir, ancak temel katmanı her zaman korur, bu da istemci tarafında videonun oynatılmasına izin vermektedir (Clayman, 2019).

SVC katmanlı kodlamada, her video çerçevesi bir temel katmana (L0) ve birkaç geliştirme katmanına sahiptir ve her katman bir dizi paket halinde oluşturulmaktadır. Geliştirilen uygulamada, her BBP özellikli ağ cihazı her zaman temel katmanı (L0) tutarken, geliştirme katmanlarına (L1 ve L2) ait parçaları çıkarabilir. Böylelikle alıcı her zaman temel katmana ait video verilerini (L0) içeren bir paket aldığından, alıcı tarafında kesintisiz video oynatımının gerçekleştirilmesi sağlanmaktadır (Clayman et al., 2021).

### 3.1.1. BPP paket yapısı

BPP tarafından sunulan yaklaşım, IPv4 ya da IPv6 paketlerinde görülen ToS (Type of Service) ya da Akış Etiketli (Flow Label) gibi alanlardan daha farklı bir şekilde, başlık (header) içerisinde protokol kapsamında tanımlanan birçok yeni alan ile, uygulamaların paket bazında oldukça hassas işlemler yapabilmesine olanak tanımaktadır. BPP başlığı içerisinde yer alan BPP bloğu, bu protokole ait farklı alanları barındırmaktadır.



Şekil 3.1 BPP Başlık ve Blok Yapısı (Li et al., 2018)

İlk olarak 2018 yılında ortaya atılan BPP protokolüne ait başlık ve blok Şekil 3.1’ de gösterilmiştir (Li et al., 2018). BPP başlığı içerisinde Komut ve Metadeta blokları yer almaktadır. Bu blokların amaçları paket içerisinde bulunan veriyle ilgili metadatanın bilgisinin BPP uyumlu ağ düğümlerine iletilmesi, buna bağlı olarak da paket alındığında gerçekleştirilecek işlemlerin belirtilmesidir. Komut ve Metadeta bloklarının içerisinde yer alan alt alanlar Şekil 3.2’ de verilmiştir. Yük (Payload) kısmı parçalara (chunk) ayrılmıştır, her bir parçayla ilgili bilgiler Metadeta içerisinde yer alırken, Komut (Command) alanı, paketin içerisinde ne tip bir işlem uygulanacağını, Koşul (Condition) alanı ise hangi durumda bu işlemin yapılacağını tanımlamaktadır.

Komut PaketKüçültme	Şart	Eşik Değeri	Off <sub>1</sub>			Off <sub>2</sub>			...			Off <sub>n</sub>		
			CRC <sub>1</sub>	SIG <sub>1</sub>	OF <sub>1</sub>	CRC <sub>2</sub>	SIG <sub>2</sub>	OF <sub>2</sub>	...	...	...	CRC <sub>n</sub>	SIG <sub>n</sub>	OF <sub>n</sub>

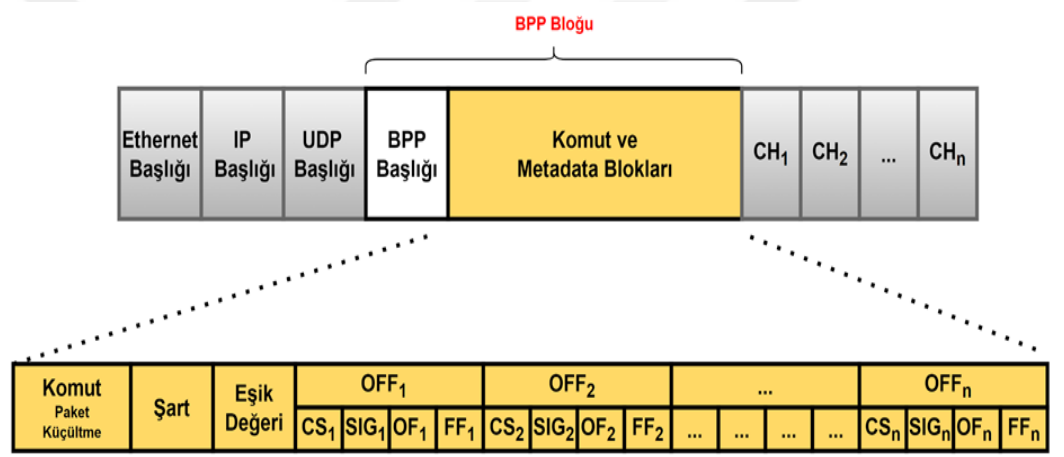
Şekil 3.2 BPP Komut ve Metadata Bloklarının Alanları (Li et al., 2019)

Paket Küçültme (PacketWash) komutu, paketin iletimi esnasında bazı parçalarının silinebileceğini BPP-uyumlu düğümlere sinyalleme amacıyla önerilmiştir (Li et al., 2019). Eşik (Threshold) alanı silinmemesi gereken parçalarla ilgili bilgi taşımak içindir. Off<sub>i</sub>, CRC<sub>i</sub>, SIG<sub>i</sub> ve OF<sub>i</sub>, i. parçanın sırasıyla paket içindeki yerini, doğrulama için CRC (Cyclic Redundancy Check) kodunu, önem derecesini ve yol boyunca düşürülüp düşürülmediği bilgisini tutmaktadır. BPP bloğu, sahip olduğu alanlar ile bir paket içerisindeki bazı kısımların iletim esnasında silinebilmesini sağlamaktadır. Böylelikle, farklı uygulamalar, kendi gereksinimleri doğrultusunda ağ düğümlerini sinyallebilir ve herhangi bir sıkışıklık anında anahtarlama cihazları paketi tamamen düşürmektense bazı kısımlarını çıkararak iletimini sağlayabilir. Bu tip bir protokol, özellikle gecikmeye duyarlı uygulamalar için oldukça avantaj sağlayabilir (Li et al., 2019).

Bir sunucudan istemciye iletilen paketlerde, hatta sıkışıklık olması durumunda paket boyutu küçültülerek, paketlerin taşıdığı önemli/kritik verinin karşı tarafa i) kısa sürede ii) mümkün olduğunca kayıpsız olarak ulaşması mümkündür. Sıkışık bir hat üzerinde UDP ile iletim yapıldığında paketler kaybolabilmekte, TCP ile iletim yapıldığında ise kayıp paketler için yapılan yeniden gönderimler ve sıkışıklık kontrol penceresinin daraltılması, paketlerin alıcı tarafına iletimini geciktirebilmektedir. UDP güvenilir olmayan, TCP güvenilir bir protokol olarak tanımlanmıştır. BPP ise kısmi güvenilir bir protokol olarak tanımlanabilir. Ayrıca, BPP, ağ içerisinde kalite adaptasyonu yapmaktadır. Mevcut video akışlandırma sistemleri ile sunucu, istemci tarafında ya da ağ destekli kalite uyarlaması yapılabilmektedir. Tüm bu sistemler, ancak paket kayıpları yaşandıktan sonra ya da belirli metriklerin toplanması ve analizinden sonra sıkışıklıklara tepki verebilmektedir. BPP ile sıkışıklık yaşanmaya başlandığı anda paket kayıplarına mümkün olduğunca izin vermeden kalite uyarlaması yapmak mümkündür.

Video iletimi için BPP kullanımı, video gönderen ve alan uygulamalar üzerinde pozitif yönde büyük bir etkiye sahip olmaktadır. Yalnızca BPP kullanmak, gelişmiş davranış veya performansı garanti etmemektedir. BPP kullanımı için öncelikle BPP paketlerinin yapısının açıklanması gerekmektedir. Temel olarak bir

bant genişliği daralması durumunda ağ düğümlerinde belirli paketlerin atılması yerine parça ismi verilen paket parçaları kaldırılarak paketi belli bir oranda küçültmek için işlemler gerçekleştirilmektedir. Ana bloklarla birlikte bir BPP paketinin yapısı (Li et al., 2018)' de tanımlanmış ve Şekil 3.3' te sunulmuştur. Bir BPP bloğu, paket işleme için yönlendiricilere rehberlik sağlamak üzere kullanıcı veya uygulamayla ilgili bilgileri içerir. Bir BPP Bloğu, bir Komut ve Meta Veri Bloğu içermektedir. Komut bölümü komutları ve bunların koşullarını ve parametrelerini taşırken, Meta veri bölümü ek meta verileri tutar. Uygulamaların yalnızca birini kullanması ve diğerini kullanmaması gerekebileceğinden, bu parçalar isteğe bağlı olmaktadır. BPP paket yapısının alanları ve katmanlı video taşımak için gerekli ek alanlarla birlikte açıklaması şu şekildedir:



Şekil 3.3 BPP Paket Yapısı ( Clayman and Sayit., 2023)

- **BPP Bloğu:** Bir BPP Bloğu, bir BPP paketi oluşturmak için düzenli IP paketi yapısına eklenen alan kümesidir. BPP Bloğunun alanları, ağ düğümlerinin paket işleme rehberi olarak kullanılabileceği uygulama ile ilgili bilgileri sağlamaktadır. Bir BPP Bloğu içinde üç tür blok bulunmaktadır: BPP Bloğu Başlığı, Komut Bloğu ve Meta Veri Bloğu. Bunların nasıl ve ne zaman kullanılacağı, uygulama sağlayıcısının uygulama gereksinimlerine göre belirlenmektedir (Li et al., 2018; Li et al., 2019). Temelde, Bir Komut Bloğu komutları, koşulları ve parametreleri tutarken, Bir Meta Veri Bloğu ek meta verileri taşımaktadır. Bu bloklar isteğe bağlıdır, çünkü uygulamalar birini kullanabilirken diğerini kullanmayabilir.
- **Komut ve Meta Veri Blok Alanları:** Komut ve Meta Veri Blok alanları, BPP Bloğunun en önemli bölümünü oluşturmaktadır. BPP

destekli ağ düğümlerinin, bir BPP paketini aldıklarında nasıl hareket edeceklerine karar vermeleri için gerekli bilgileri sağlamaktadır. Komut, Koşul, Eşik değeri ve Eylemler, Komut ve Meta Veri Blok alanları arasında yer almaktadır. Bu bloklar ayrıca mevcut paket içindeki parçalarla ilgili ek meta verileri taşımaktadır.

Şart alanı, bir paketten herhangi bir parçanın çıkarılıp çıkarılmayacağı konusunda bilgi oluşturmak için kullanılmaktadır. Ağ düğümüne, parçaların hangi koşullarda bir paketten çıkarılması gerektiği konusunda karar vermesi amacıyla sinyal vermek için sunucu bir Şart değeri ayarlayabilmektedir. Sunucu, video kodlama özelliklerine dayanarak sezgisel bir sayı hesaplayabilmekte ve bu değeri Şart değeri olarak kullanabilmektedir.

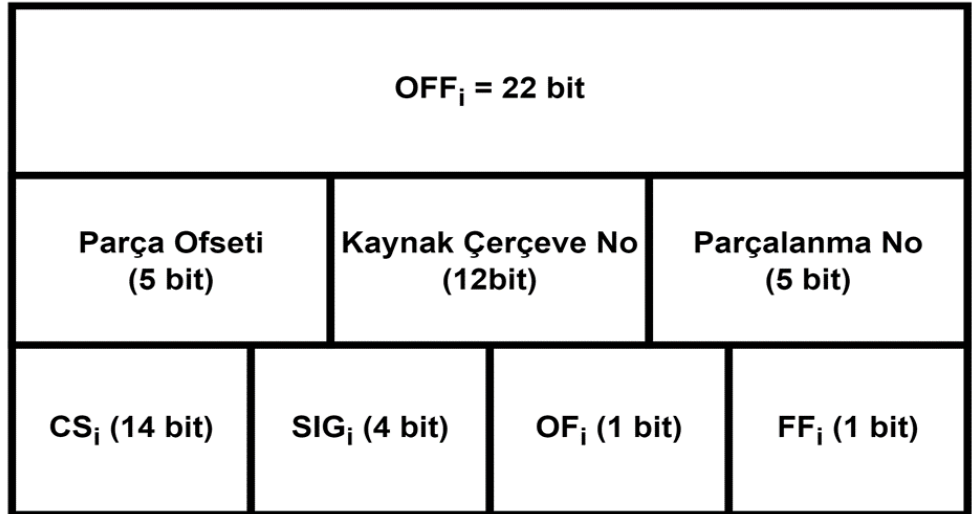
Eşik, hangi parçaların düşürülemeyeceğini belirleyen sınır olarak kullanılmaktadır. Parçaların daha fazla düşürülemeyeceği önem değerini temsil etmektedir. Bu değer, video çerçevelerinin ve katmanlarının önemi dikkate alınarak belirlenmektedir. Komut ve Meta Veri Blok alanlarında, paket içinde taşınan her parça için bilgi taşıyan alanlar da bulunmaktadır. OFFi ve SIGi, sırasıyla yük (payload) içindeki ofset konumu ve parçanın öncelik değerini temsil ederken, OFi ilgili parçanın düşürülüp düşürülmediğini göstermektedir. BPP, Paket Küçültme (Packet Wash) olarak bilinen, paket iletimi sırasında bazı parçaların düşürülmesini sağlayan çeşitli komutları desteklemektedir (Li et al., 2019). Bu yaklaşım, bant genişliği sınırlı olduğunda paketin tamamının düşürülmesini önlemeye yardımcı olmaktadır, böylece paket boyutunu mevcut bant genişliğine uygun şekilde azaltmaktadır. Paket içerisindeki öncelik değeri düşük olan parçalar atılabilmekte ve alıcı tarafına ulaşan verinin bazı kullanılabilir bilgiler içermesi sağlanmaktadır. Alıcı tarafında elde edilen veri sunucu tarafından gönderilen veriyle aynı olmasa da hiç veri gönderilememesinden veya gecikmiş veri gönderilmesinden daha iyi bir durum oluşmaktadır. Tez çalışmasında, akışkan video yönetimi için etkili bir teknik sağladığı için Paket Küçültme komutu BPP paket iletiminde doğrudan kullanılmıştır. Katmanlı video iletimini desteklemek için, BPP çerçevesinin Komut ve Meta Veri Blok alanları yeni alanları içerecek şekilde genişletilmiştir, ayrıntıları aşağıda verilecektir (Clayman et al., 2021b).

### **3.1.2. Video akışı için oluşturulan ek BPP blok alanları:**

Her ne kadar literatürde BPP' nin video uygulamaları için avantajlı olacağı belirtilmiş olsa da, paket içerisinde video iletimi için gerekli alanlar henüz düşünülmemiş ya da tartışılmamıştır. Tez çalışması içerisinde BPP paket yapısı ile

SVC kodlanmış video dosyalarının iletimi gerçekleştirileceği için, ilk olarak BPP başlık alanları video iletimini destekleyecek şekilde düzenlenerek eklemeler yapılmıştır. Özellikle I tipi kareler ve taban katmanlar boyut dolayısıyla birden fazla paket içerisinde taşınabilmektedir. Ayrıca paketleme tekniğine bağlı olarak düşük boyutlu bir katman bile birden fazla pakete dağıtılabilmektedir. Video çözücünün de ihtiyaçları göz önünde bulundurulduğunda, paket içerisinde taşınan parçaların her biri için kare numarası ve kare/katman parçalarına dair bilgilerin istemciye taşınması gerekmektedir. Bu bilgiler az sayıda bit ile ifade edilerek Şekil 3.4' te görüldüğü gibi Offi içinde taşınması sağlanmıştır. BPP alanları Ethernet paketi içerisinde gerçekleştirilmiştir. Video dosyası, SVC ile bir temel katman ve n adet genişleme katmanı içerecek şekilde oluşturularak BPP paketlerine yerleştirilmektedir. Ethernet MTU (Maximum Transmission Unit) 1500 Byte olduğu için video bu paketlere sığacak şekilde düşük bit hızıyla kodlanmıştır.

BPP Bloğu Başlığı alanlarının yanı sıra, paket içinde kaç adet parçanın bulunduğunu gösteren yeni bilgiler eklemesi işlemi gerçekleştirilmiştir. Bu alana Parça Sayısı (Chunk Count) adı verilmiştir. Paket iletimi sırasında paket değiştirildiğinde, bazı parçaların silinmesinden dolayı YTA denetleyicisinin ve istemcinin parça ofsetlerinin bitiş adresini ve yükün başlangıç adresini anlayabilmesini sağlamak için gerekmektedir. Parça Sayısı, Meta Veri ofset alanına yazılır ve bu alanın boyutu 5 bittir. Parça Sayısına ek olarak, (Li et al., 2019; Li et al., 2018) çalışmalarında teklif edilen alanlara tez çalışması kapsamında bazı ek alanlar eklenmiştir, böylece Komut ve Meta Veri Bloğundaki parçalar da video ile ilgili bilgilere sahip olabilmektedir.



Şekil 3.4 Parça (Chunk) Alanları (Clayman et al., 2021b)

BPP Bloğu içerisinde yer alan bölümler Şekil 3.4' te gösterilmektedir. Alanlardan bazıları kaynak makalelerden elde edilen bilgilerle oluşturulmuşken bazıları ise çalışmada ortaya çıkan ihtiyaçlara yönelik olarak oluşturulmuştur. Özellikle alanların boyutları ve görevleriyle ilgili olarak en baştan net tanımlamalara sahip olunmaması bu uygulamanın geliştirilmesi sürecinde yaşanan en büyük zorluk olmuştur. Çalışmada duyulan ihtiyaçlar doğrultusunda her bir parça için Şekil 3.4' te gösterilen alanlar oluşturulmuştur.

Eklenen ek alanlar, her bir parça için BPP Bloğuna toplamda 42 ek bit eklemektedir.

**Ofset (OFFi-22 bit):** Ofset alanı (Li et al., 2018) çalışmasında tanıtılmış olmasına rağmen, burada video akışı uygulamalarını desteklemek için parça Ofsetine 12 bit uzunluğundaki Kaynak Çerçeve Numarası (Source Frame No) ve 5 bit uzunluğundaki Parçalanma Numarası (fragmentation No) alanları eklenerek geliştirilmiştir. Tez çalışmasında kullanılan yeni parça Ofset yapısı olan OFFi' nin ayrıntıları Şekil 3.4' te gösterilmiştir ve her bir parça için kullanılan ek alanları ifade etmektedir. Bazı kareler boyutlarından dolayı birden fazla pakete bölünebilirler. Parçalanma numarası, parçaların istemci tarafında doğru bir şekilde birleştirilmesi için kullanılmaktadır.

Ofset alanı üç alt alana ayrılır:

- **Parça Ofseti (Chunk Offset -5 bit)** alanı, bir paket içindeki parça numarasını göstermektedir.
- **Kaynak Çerçeve No (12 bit)**, tam veya parçalı çerçevelerde taşınan çerçeve numarasını göstermektedir
- **Parçalanma No (5 bit)**, ilgili çerçeve bir pakete sığmayıp parçalanmışsa kaçınıcı parça olduğunun numarasını göstermektedir.

Genişletilmiş BPP paket yapısı aynı zamanda katmanlı video hakkında bilgi tutmak için aşağıdaki alanları da kullanır:

**Parça Büyüklüğü (Chunk Size-CSi (14 bit)):** Geçerli parçanın boyutunu göstermektedir. Aynı amaca yönelik benzer bir alan (Li et al., 2018) çalışmasında kullanılmaktadır.

**Öncelik Seviyesi (Significance-SIGi (4 bit)):** Parçanın öncelik değerini göstermektedir.

**Düşürülme Bayrağı (Is Dropped Flag-OFi (1 bit)):** Parça düşürüldüyse 1, düşürülmediyse 0 olarak ayarlanmaktadır.

**Parçalanma Bayrağı (Fragmentation Flag- FFi (1 bit)):** Geçerli parçanın son parça olup olmadığını göstermektedir.

Önerilen BPP başlık yapısı tarafından getirilen fazlalık, daha büyük çerçeveler (Jumbo Frame) kullanılsa bile sabit kalmaktadır (Clayman et al., 2023).

Alanlara karar verilmesi sonrasında matematiksel çözümler kullanılarak bit seviyesindeki işlemlerin doğru bir şekilde yapılması sağlanmıştır. Bir çerçeve içerisindeki varsayılan başlıklar düşüldükten sonra kalan alan için parça sayısına bağlı olarak aşağıdaki alanların oluşturulması gerekmektedir.

Tablo 3.1 BPP Paketine Ait Alanlar

PAKETE AİT ALANLAR		
ALAN ADI	UZUNLUK (bit)	AÇIKLAMA
BPP BAŞLIĞI (BPP_DEFAULT_HEADER)	32	4 Baytlık BPP Header kısmı boş alan bırakıldı sonradan kullanılacaktır. İlerideki farklı BPP işlemleri için ayrılmıştır.
PAKET UZUNLUĞU (PACKET_LENGTH_BYTE)	11	Paketin toplam uzunluğu Byte Cinsinden tutulmaktadır. İlk pakette Maksimum değer olarak 1500 Byte olarak yazılmaktadır.
PARÇA SAYISI (HEADER_CHUNK_COUNT)	5	Paket içerisindeki Parça (Chunk) sayısı tutulmaktadır. En çok 32 adet parça adreslenebilecek şekilde oluşturulmuştur.
KOMUT (COMMAND)	5	Komut türünü ifade etmek için kullanılmaktadır. 32 farklı komut yazılabilmektedir.
ŞART (CONDITION)	8	BPP için belirlenen şart değerini göstermektedir. 1-256 arasında bir değer olabilir.
EŞİK (THRESHOLD)	8	Paket için eşik değeri (Bozulma noktası) göstermektedir. 1-256 arasında bir değer olabilir.

Tablo 3.1 BPP Paketine Ait Alanlar (devam)

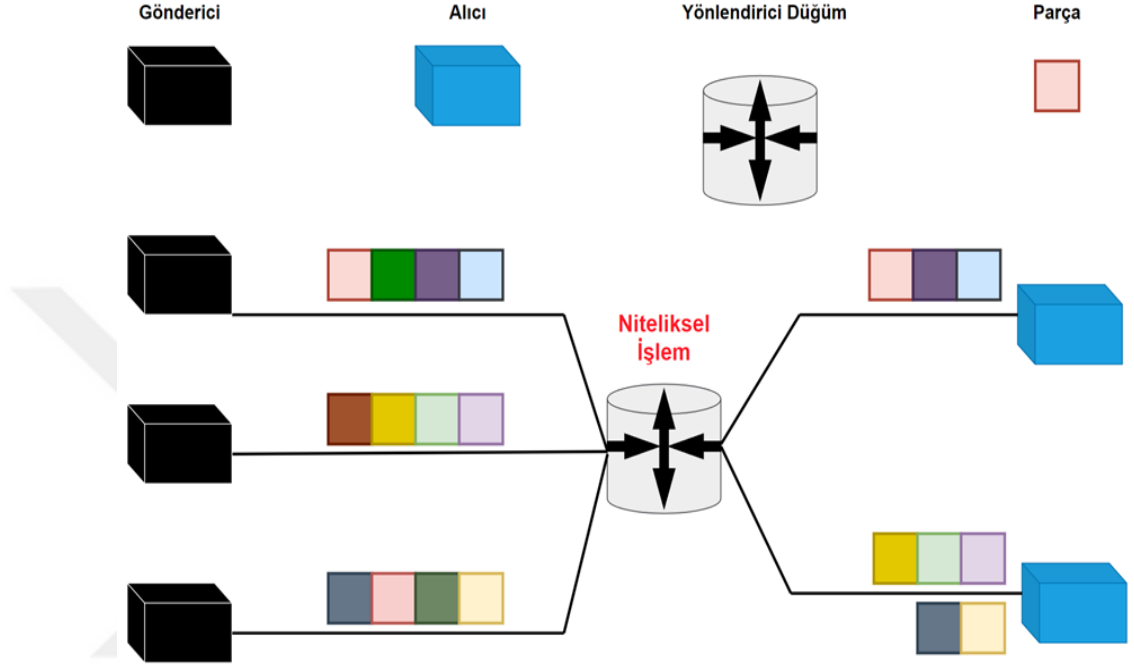
PARÇAYA(CHUNK) AİT ALANLAR		
PARÇA OFSETİ (CHUNK_OFFSETNo)	5	Parçanın (Chunk'ın) Paket içerisindeki kaçınıcı parça (Chunk) olduğunu göstermektedir.
KAYNAK ÇERÇEVE NUMARASI (SOURCE_FRAME_NO)	12	Chunkın ait olduğu çerçeve (Frame) numarasını göstermektedir
PARÇALANMA NUMARASI (FRAGMENTATION_NO)	5	Parçalanmış bir çerçeve (Frame)'e ait bir parçanın (Chunkın) ilgili çerçeve (Frame)'in kaçınıcı parçası olduğunu göstermektedir
DÜŞÜRÜLME BAYRAĞI (IS_DROPED)	1	Parça (Chunk) düşürüldüğünde (Drop edildiğinde) 1 değeri atanmaktadır. Varsayılan olarak 0 değerine sahiptir.
PARÇALANMA BAYRAĞI (FRAGMENTATION_FLAG)	1	Parça (Chunk) bölümlenmiş (fragmente edilmiş) bir çerçeveye (Frame'e) aitse ve son parça değilse 1 değeri atanmaktadır. İlgili çerçeve için bölümlemenin devam ettiğini göstermektedir.
ÖNCELİK DEĞERİ (SIGNIFICANCE)	4	Parçanın (Chunk'ın) önem/öncelik (significance) seviyesini göstermektedir.
PARÇA BOYUTU (CHUNK_SIZE)	14	Parçanın(Chunk'ın) bit cinsinden boyutunu ifade etmektedir.

Ayrıca bir sonraki bölümde detaylı açıklanacak olan ONOS içerisinde geliştirilen BPP\_APP isimli uygulama BPP paketini açarak gereken alanları okumaktadır. Paket üzerinde herhangi bir küçültme işlemi gerçekleştirildiğinde ilk olarak Paket Uzunluğu (Packet Length) ve sonrasında silinen parçanın Düşürülme Bayrağı (isDropped) alanı ayarlanmaktadır. Paket içerisinde bulunan alanların açıklamaları Tablo 3.1' de verilmiştir. Tez çalışması kapsamında oluşturulan ilk BPP paket yapısı bu tablodaki mantıkla gerçekleştirilmiş ve alanlar oluşturulmuştur. Tablo içerisinde BPP paket yapısı içerisinde bulunan tüm alanlar orijinal adları, uzunlukları ve açıklamaları ile yer almaktadır. Ayrıca bir parçaya ait olan alanlar da ayrıca belirtilmiştir. Video akışlandırmaya uygun BPP paket yapısının ilk oluşturulması kısmında bu tablo hazırlanmış ve sonrasında uygulanması gerçekleştirilmiştir. Açıklama kısmında ilgili alanların alabileceği değerlere dair de bilgilendirme bulunmaktadır.

### 3.1.3. Video akışında BPP bloğunda kullanılan paket küçültme komutu

Paket Küçültme komutu, video iletimi için Komut ve Meta Veri Bloklarının bir parçası olarak taşınmaktadır. Paket Küçültme Komutu, nitelikli iletişimlerde mümkün olduğunca çok bilgiyi korurken bir paketin boyutunu azaltan bir

temizleme işlemi olarak tanımlanmaktadır. Şekil 3.5’ te görüldüğü gibi Paket, yük (payload) içinde taşınan bilgiye bağlı olarak daha düşük öncelikli parçaları bırakarak çalışmaktadır ve ileten düğümün parçaların anlamını (veya ilişkisini) anlamasına yardımcı olmaktadır. Kaybedilen parçalar kurtarılmaz, ancak alıcı tarafında bazı bilgiler kullanılabilir hale gelmektedir.



Şekil 3.5 Paket Küçültme (Packet Wash) İşlemi (Li et al., 2019)

Şekil 3.5’ te görüldüğü gibi paket ağ üzerinde bir ara düğümden geçerken duruma bağlı olarak bazı parçaların silinmesi işlemi gerçekleştirilebilmektedir. Silme işlemi için paket üzerinde bir bozulma noktası eşik değeri belirlenmiştir. Bu noktaya ulaşılmış ise veya parçaların öncelik değerleri silinemez seviyede ise parça silme işlemi gerçekleştirilememektedir. Bu yaklaşım SVC yapısındaki katmanlar için oldukça uygundur. BPP için kullanılan ara düğümler de YTA denetleyici tarafından yönetilen OpenFlow özellikli anahtar cihazları olarak düşünülebilmektedir.

(Li et al.,2019) çalışmasında belirtildiği gibi Paket Küçültme (PacketWash) işlemi ile bazı parçalar paketten düşürülebilmektedir. Paket Yıkama işlemi, yük içeriğini daha az önemliden daha önemli parçalara doğru seçerek gerçekleştirilmektedir. Bu doğrultuda ileten düğüm, hangi paketin ne kadar kırılacağına ve hangi parçaların kırılacağına karar vermektedir. Tıkanıklık durumunda ileten bir düğüm, geliştirme katmanlarını içeren parçaların ne kadarının

kaldırılması gerektiğini bilerek işlemi gerçekleştirmektedir. BPP paket yapısı ve paket küçültme işlemleri ilk olarak (Li et al.,2019) çalışmasında önerilmiştir. Fakat herhangi bir uygulaması gösterilmemiştir. Tez çalışmasında bu çalışma referans alınarak BPP işleminin gerçekleştirilebilmesine uygun paketleme yapıları oluşturulmuştur. Ayrıca BPP işleminin gerçekleştirilmesini sağlayacak uygulamaların oluşturulması işlemi gerçekleştirilmiştir. Yapılan çalışmalar sonucunda çeşitli konferans ve dergi yayınları gerçekleştirilmiştir.

BPP' nin tanıtılması ve video iletimi için BPP' de yapılan geliştirmelerin açıklanmasının ardından, YTA denetleyicisinin videoyu nasıl işlediğini gösteren işlemler bir sonraki bölümde açıklanacaktır.

### **3.1.4. Video akışının BPP' ye eşlenmesi**

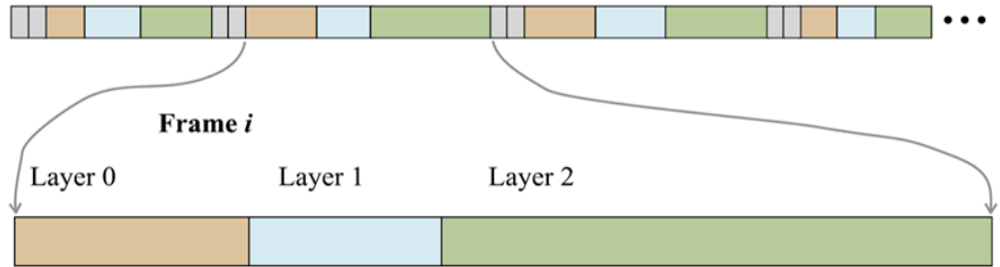
BPP' nin video taşımada etkili olması için, BPP, aynı kare için birden fazla kodlama yapabilen bir kodlayıcı ve dekodlayıcıyla birleştirilmelidir. Ayrıca BPP komutlarını işleyen bir ağ içi fonksiyon kullanılmaktadır (Clayman et al., 2021a). Video içeriği, BPP için uygun bir şekilde düzenlenip kodlanmalıdır. Göndericideki paket oluşturma yöntemi BPP iletim sürecinin farkında olmalı ve paketler, iletim sırasında video parçalarının bazılarının çıkarılmasına izin verecek şekilde oluşturulmalıdır. Bu yaklaşım, istemcilerin kabul edilebilir bir kalitede videoyu elde etmelerini sağlamaktadır.

BPP iletilmek istenen paketin tamamının düşürülmesi yerine, bir paketten bazı parçaları atabilmekte ve daha küçük bir paket teslim edebilmektedir. Bu durum, tüm paketin atılmasından farklıdır ve istemci tarafında daha iyi sonuçlar elde edilmesini sağlamaktadır. Ancak, video blokları hiçbir ön işlem yapılmadan BPP parçalarına konulursa ve herhangi bir kontrol yapılmadan bu parçalar iletim sırasında atılırsa, alıcı tarafta video veri kaybı görülecektir. Bu kayıp alıcı tarafından ele alınmalıdır ve bu, bir paketin kaybına çok benzer bir etkiye sahiptir. Bu şekilde kullanıldığında, BPP çok az fayda sağlayacaktır. Bir H264 video akışının yapısının detayları incelendiğinde ve akış içindeki segmentler, Ağ Soyutlama Katmanı (NAL) olarak adlandırılmaktadır. Farklı katmanların nasıl seçildiği ve bu katmanlar için verilerin BPP paketlerine nasıl haritalandığı tez içerisinde detaylandırılmıştır. Bu yaklaşım, diğer parçalar kaldırılırsa bile, her paket için taban katmana ait en az bir parçanın teslim edilebilmesini garanti etmektedir. Ayrıca şekiller, istemcinin BPP paketlerini nasıl aldığını ve geçerli bir H264 geri oynatma akışı oluşturmak için NAL' ları nasıl yeniden oluşturduğunu da göstermektedir.

### 3.1.5. BPP paketleri için oluşturulan SVC H264 yapısı

BPP üzerinden iletim için SVC' yi kullanan bir H264 dosyasının yapısı bu bölümde sunulmaktadır. Bir H264 dosyası, 2 ana tür veri tutan bir NAL dizisine sahiptir. VCL, Kodlanmış video verilerini içeren Video Kodlama Katmanlarını ifade ederken; NONVCL, geçerli bölüm video akışıyla ilgili meta verileri tutan Video Olmayan Kodlama Katmanlarını ifade etmektedir (Clayman and Sayit, 2023). Genel olarak bir SVC videosunun yapısı, her çerçeve türü için tek bir VCL NAL ile 2 tür NAL gördüğümüz normal bir H264 dosyasına benzemektedir. SVC video dosyalarında, bir çerçeve için birden fazla NAL görülebilir. Video, kavramsal olarak temel katman bilgisini sağlayan Katman 0 ve taban katmanına iyileştirmeler sağlanarak elde edilen Katman 1 ve Katman 2 olarak görüntülenebilmektedir. Genel olarak, çerçeve başına daha fazla NAL üretecek herhangi bir sayıda katmana sahip bir kodlama kullanılabilir. Bir H264 dosyasında, katmanlı yapı bir bayt akışına yerleştirilir ve birbiri ardına görünen her çerçeve için VCL NAL paketleriyle sıralı alt akışların bir dizisi olarak sunulmaktadır. Meta verilerden oluşan NONVCL NAL paketleri, bu video verisi için NAL paketlerinden önce gelmektedir.

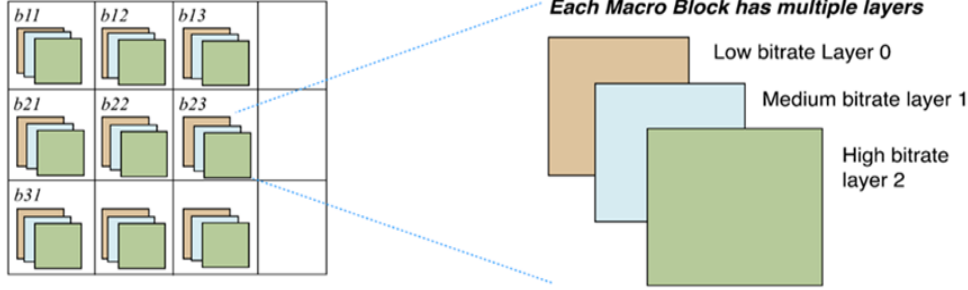
Şekil 3.6, bu diziyi ve NONVCL NAL paketlerin VCL NAL paketleri ile nasıl karıştırıldığını göstermektedir. Şekil 3.6' da görüldüğü gibi NAL paketleri katmanları temsil etmektedirler ve Kahverengi, Mavi, Yeşil renklerle gösterilmektedir. Ayrıca NON VCL NAL paketleri de gri renkle gösterilmektedir. Bu tür paketlerin boyutları küçük olduğu için de paket içerisinde az yer kaplamaktadır. Video verileri için 3 NAL görülmektedir. Yapılan çalışmada bu 3 NAL video dosyasından okunur ve iletim için işlenirler.



Şekil 3.6 Paket yapısı (Clayman et al., 2021b)

Şekil 3.7' de, video çerçevelerinin ana bileşenleri olan Makro Bloklar, gösterildiği gibi katmanlı bir kodlama ile kodlanmıştır. Şekilde 3 katman olduğunu

görülmektedir: En düşük bit hızlı veriye sahip Katman 0 (temel katman), orta bit hızlı veriye sahip Katman 1 ve en yüksek bit hızlı veriye sahip Katman 2' dir. Bu katmanlı bloklar, BPP paketlerinin parçaları olacaktır.



Şekil 3.7 Makro Bloklar (Clayman and Sayıt, 2023)

Şekil 3.7' de Paket yapısında gösterilen video akışı içinde, VCL NAL' lar arasına serpiştirilmiş bir dizi NONVCL NAL vardır. Bunlar genellikle küçüktür ve kolayca bir pakete yerleştirilebilir. Bu yaklaşımda, NONVCL NAL' lar tek bir BPP öbeği olarak gönderilmektedir. 3 katmanlı bir video için video akışındaki her video karesi için 3 VCL NAL görülmektedir. Neredeyse tüm videolar için çerçeve boyutu ve NAL boyutu, standart 1500 baytlık bir paketten çok daha büyüktür, bu nedenle, NAL dizisinin her çerçeve için toplanması ve ardından BPP paketlerine eşlenmesi gerekmektedir. Bu eşleme görevi, VCL' ler için 3 ana işleme aşaması kullanılarak yapılmaktadır:

1. Video akışından NAL dizisi toplanır ve numaralandırılır. Bu aşamada bir H264 akış işlemcisi girişten 3 VCL NAL toplamaktadır;
2. Her çerçeve için toplanan 3 NAL bölünür ve bunlar ChunkInfo nesnelere olarak adlandırılan bir dizi ara nesne içerisine yerleştirilir;
3. ChunkInfo nesnelere dizisi alınır ve bunlar iletme hazır BPP paketlerine dönüştürülür.

Aşamaların birinci adımında, girdi verilerinin bitlerini ve baytlarını döndürebilen ve her bir NAL paketini döndürebilen bir H264 akış okuyucusu ve işlemcisi gerekmektedir. Bir ara ChunkInfo formuna sahip olmak, NAL' ları paketler halinde eşleştirilmeden ve ağa gönderilmeden önce işlerken bir dereceye kadar esneklik ve uyarlanabilirlik sağlamaktadır. 1 veya daha fazla tam NAL (küçük oldukları için genellikle NONVCL' ler) veya parçalanmış 1 NAL (genellikle VCL) içerebilen her ChunkInfo nesnesi için aşağıdaki veriler tutulmaktadır:

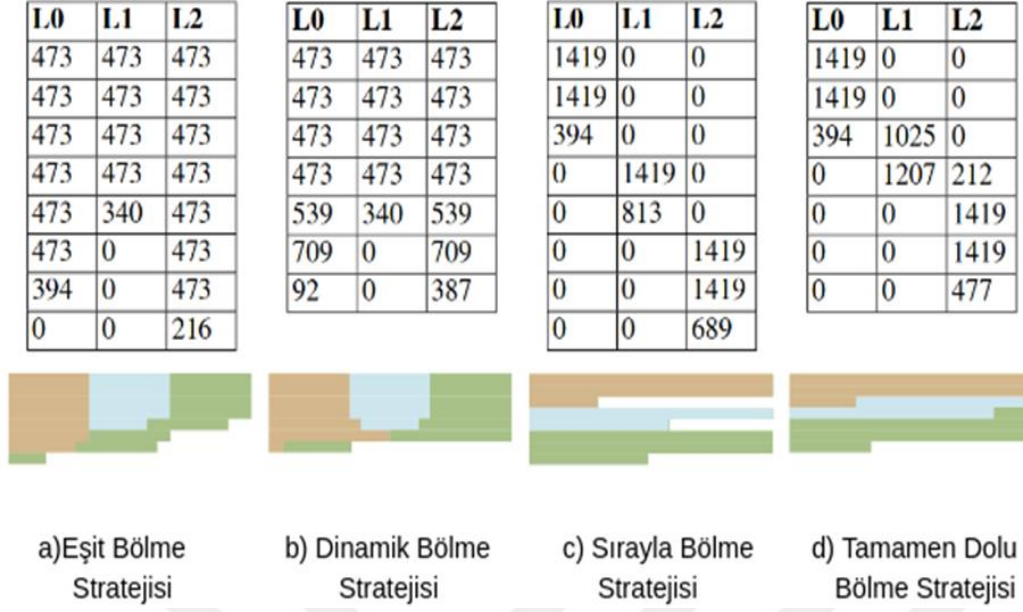
- NAL tipi: NONVCL veya VCL
- NAL numarası
- Akış içinde tutulan NAL sayısı
- Parçalanma sayısı
- Son Parça olup olmadığının kontrolü
- Parça boyutu (bayt cinsinden)
- Parçanın yük (payload) baytları
- Bir pakette kalan bayt sayısı

ChunkInfo nesnesi, akış üzerinde analiz yapılmasını sağlamaktadır. Bir pakete yerleştirilebilecek veri hacmi ve zamanlama hesaplamalarının yapılmasına izin vermektedir. Ayrıca, VCL NAL' ların BPP paketlerine konulabilmesinin farklı stratejileri olduğundan, paketler için bir dizi paketleme ve çoğullama stratejisini kolaylaştırmaktadır (Clayman and Sayit, 2023).

### 3.2. BPP Paketleme Stratejileri

Tez çalışmasının ikinci adımında, video verilerini VCL NAL' larından toplamak ve bu 3 alt akışı video verisinin ChunkInfo nesnelere ve ardından bir dizi BPP paketinin bölgelerine paketlemek için geliştirilen çeşitli paket doldurma stratejileri açıklanmaktadır. BPP paketlerinin oluşturulması için 4 farklı strateji kullanılmaktadır. Her stratejinin farklı özellikleri vardır ve farklı QoE parametreleri arasında bir denge sağlamaktadır. Oluşturulan stratejilere ait parça boyutları doluluk oranları Şekil 3.8' de gösterilmiştir. Verilen boyutlar 1500 Bayt büyüklüğünde bir Ethernet Çerçevesi içerisindeki yerleştirme durumu için ele alınmaktadır.

Bu çerçevede BPP başlıkları için kullanılan alan düşüldükten sonra yük taşıma için kullanılabilir boş alan parçalara ayrılarak video dosyasının katmanlarının yerleşimleri sayısal olarak Bayt cinsinden gösterilmektedir. Şekil 3.8' de sunulan veriler, farklı stratejilerin aynı video dosyası üzerinde farklı paket yapıları ve farklı boyutlarda parçalar ürettiğini göstermektedir. 1500 baytlık paket boyutu kullanılmaktadır ve paketlerin başlık boyutlarının çıkarılmasıyla 1419 bayt içerik boyutu sağlanmaktadır. Bu alan, BPP başlığı ve ilgili BPP meta verilerini sağlamaktadır ve parçalar içerisine video verileri tahsis edilmektedir. Şekil 3.8' deki örnekte, 3 adet VCL NAL için orijinal boyutlar şu şekildedir: Katman 0: 3232 bayt, Katman 1: 2232 bayt ve Katman 2: 3527 bayt.



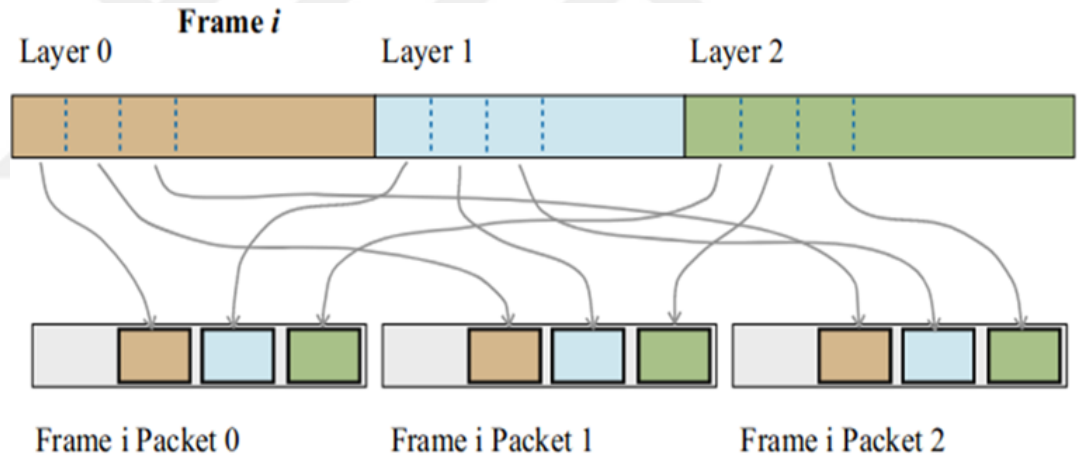
Şekil 3.8 Paketleme Stratejilerine Göre Oluşan Parça (Chunk) Boyutları (Clayman and Sayit, 2023)

Şekil 3.8’ de her paketleme stratejisi için bir tablo ve bir grafik bulunmaktadır. Tabloda her paketin bölgelerinin boyutları gösterilirken, grafikte farklı katmanların parçalarının orantılı boyutu ilgili renk ile gösterilmektedir. Katmanlar sırası ile L0: Kahverengi, L1 Mavi, L2: Yeşil renklerle gösterilmektedir.

Oluşturulan paket stratejileri ve özellikleri şu şekildedir:

**Eşit Bölme (Pe):** Video dosyasından gelen veriyi eşit boyutlu bölgelere eşit bir şekilde bölerek paketler oluşturulmaktadır. Testlerde 3 katmandan oluşan video dosyaları kullanılmaktadır. Her pakette çerçeveye (Frame’ e) ait katmanların üçünden de parça oluşturulmaktadır. Eğer çerçeveye ait katmanın gönderimi tamamlanmışsa ilgili parça boş bırakılmaktadır. Gelen veriler eşit boyutta parçalara bölünmektedir. 3 NAL’ ımız olduğunda, her bölgeye aynı miktarda veri alanı tahsis edilmektedir. Bu yaklaşımla tüm paketlerde en az bir çıkarılabilir parça bulunur ve genellikle bu parça küçük boyuttadır. Bu yaklaşım, teslim edilen pakette videonun kesintisiz bir şekilde oynatılabilmesi için hala gerekli parçalara sahipken paketlerin küçültülmesine izin vermektedir. Bu strateji, kalite uyarlamasında ince ayar yetenekleri sağlamaktadır. Şekil 3.9’ da video dosyasına ait 3 katmanı (L0-L1-L2) sunucu tarafında oluşturulan paketlere eşleyen ve paketlerin her birinin bu 3 katmandan bazı verileri nasıl içerdiğini gösteren şema sunulmaktadır.

**Dinamik Bölme (Pd):** Gelen veri miktarlarına dayanarak hangi bölgelerin veri tutması gerektiğini belirlemektedir. Bu strateji, video dosyasına NAL' lardan gelen girdi miktarlarına bağlı olarak kaç parçanın veri tutması gerektiğini belirlemektedir. İlk olarak bir önceki strateji olan eşit bölmeyi uygulamakta ve eşit olarak dağıtım dolduğunda bitirilmektedir. Sonrasında kalan verilerle NAL' lara tahsisi bayt cinsinden artırarak parça alanlarını uygun şekilde yeniden tahsis etmektedir. Eşit Bölme stratejisine benzer şekilde, tüm paketlerde en az bir çıkarılabilir parça vardır. Bu yaklaşımın farkı, oluşturulan paket sayısının daha az olmasıdır. Bu özellik, istemcilere daha kısa sürede paket teslimatı sağlamak ve bant genişliği yeterli olduğunda daha kısa duraksamalara neden olmaktadır. Oluşturulan paketler Şekil 3.9' da gösterilen şemayla aynı şekilde veri içermektedirler. Aralarındaki fark paketlerin içerisindeki parçaların boyutlarındaki değişimdir. Bu stratejide parçaların boyutları ihtiyaç durumunda artırılmaktadır. Bu durum da Şekil 3.8' den boyutları da aynı şekilde bulunan a ve b tablolarından anlaşılabilir.

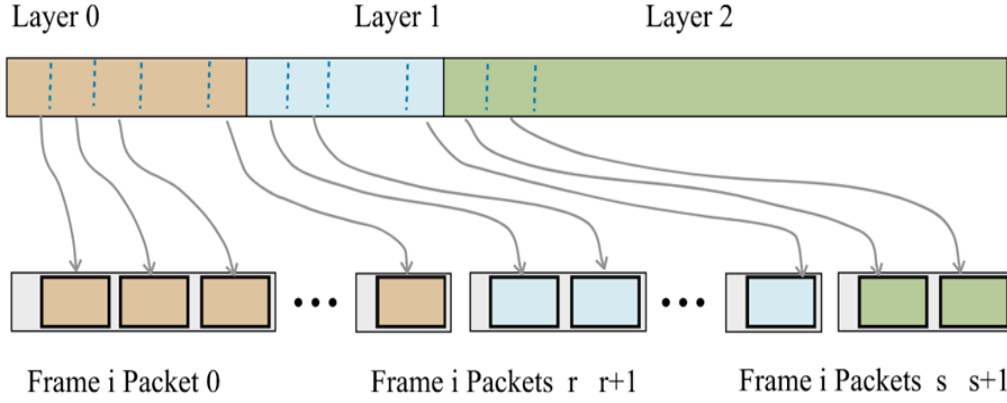


Şekil 3.9 Eşit Bölme ve Dinamik Bölme Paket Stratejisi (Clayman et al., 2021b)

**Sırayla Bölme (Pi):** BPP paket stratejileri arasında ilk olarak oluşturulan yaklaşımdır. Daha sonra bu yaklaşım baz alınıp geliştirilerek diğer stratejiler oluşturulmuştur.

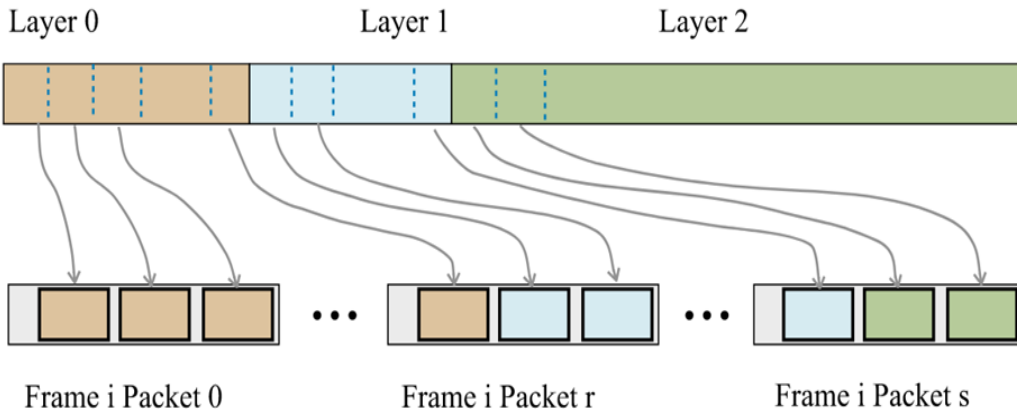
Şekil 3.10' da ilk paket modelinde gösterildiği gibi ilgili video çerçevesi taban katmanından (L0) başlanarak pakete yerleştirilmektedir. Bu yüzden bir paketin tamamı L0 içeren bir parçadan oluşabilmektedir. Paket içerisindeki parçalar video dosyasına ait NAL' lardan geldikleri sırayla oluşturulmaktadır. Şekil 3.8c' de görüldüğü gibi yalnızca bir NAL' dan gelen veriler, daha küçük boyutta olabilecek son parçaya kadar paketi doldurmaktadır. Diğer katmanlara ait tüm parçalar boş

olarak bırakılmaktadır. Bu stratejide, ağ düğümü bir parçayı kaldırırsa, paket istemciye hiçbir video içerik parçası olmadan teslim edilebilmektedir. Bu durum, video verilerini beklerken gecikmelere neden olabilmektedir.



Şekil 3.10 Sırayla Bölme Paket Stratejisi (Clayman and Sayıt, 2023)

**Tamamen Dolu Bölme (Pf):** Bu strateji Sırayla Bölme yaklaşımına benzer bir yapıdadır. Ancak bir NAL'ın son bölgesinde küçük miktarda veri kaldığında, kalan alan daha sonraki NAL'a tahsis edilmektedir. Şekil: 3.8d' de görüldüğü gibi bir paket içerisinde her iki katmana ait parça bulunabilmektedir. Çerçeve boyutunun Sırayla Bölme stratejisine göre daha etkin kullanılması amaçlanmaktadır. Bu stratejinin kullandığı kalite benimsenmesindeki esneklik sayesinde, sırayla bölme yaklaşımı ile diğer stratejiler arasında yer almaktadır. Şekil 3.11' de görüldüğü gibi oluşturulan paketler, büyük boyutlu bir parçaya sahip olabilecekleri gibi çıkarılabilir küçük boyutlu bir parçaya da sahip olabilirler veya çıkarılabilir hiçbir parçaya sahip olmayabilirler.



Şekil 3.11 Tamamen Dolu Bölme Paket Stratejisi (Clayman and Sayıt, 2023)

Yukarıda sunulan stratejilerin farklı özellikleri, QoE parametreleri açısından farklı performans sonuçlarına yol açmaktadır. Sırayla ve Tamamen Dolu stratejileri ile temel katmanlı video, ilk paket kümesine konulmaktadır. Bu nedenle, bant genişliği sınırlı ve paketler gecikmişse, istemciler paketleri temel katmanla birlikte aldıklarında hemen videoyu oynatabilirler, bu da kısa süreli duraksamalar sağlamaktadır. Öte yandan Eşit Bölme ve Dinamik Bölme yaklaşımlarında katmanlar paketlere dağıtıldığı için ağ içi kalite adaptasyonu daha ince ayarlanabilir ve daha iyi yapılabilir durumda olmaktadır. Kullanılan stratejilerde iletilecek video dosyasına ait NAL paketi, strateji tarafından ayrılan bölgelerden daha büyükse, ayrılmış olan bölge boyutu kadar bayt NAL' dan alınıp bölgelere konmaktadır. Sonrasında kalan NAL verisi bir sonraki bölge için saklanmaktadır. Kalan NAL verisi ayrılmış olan bölge alanından daha küçükse, bölge yalnızca bu baytları içermektedir. Bu şekilde bölünen tüm NAL' lar için bir parçalanma numarası oluşturulmaktadır ve ilgili parçalar bu numarayla etiketlenmektedir. Böylelikle, istemcide güvenilir bir şekilde video dosyasına ait çerçevelerin yeniden oluşturulması sağlanmaktadır. Eğer NAL' lardan biri daha fazla veri içermiyorsa, tüm veri bölgelere ayrıldığı için, yine de bir bölge tahsis edilmektedir ancak bölgenin boyutu 0 bayt olarak ayarlanmaktadır.

Bu paket stratejilerinin kendi içlerinde performans karşılaştırmaları yapılarak ilgili sonuçlar tezin bir sonraki kısmında gösterilmiştir.

#### 4. YTA ORTAMINDA BPP ÜZERİNDEN VIDEO İLETİMİ İÇİN GELİŞTİRİLEN UYGULAMALAR

Bu bölümde, alternatif bir taşıma katmanı protokolü olan BPP' nin video iletimi için kullanılmasına daha ayrıntılı bir bakış sağlanmaktadır. BPP için çok fazla davranışsal veri bulunmamaktadır, çünkü tez çalışması öncesinde farklı senaryolarda değerlendirilmemiştir. Mevcut MPEGDASH standardı, TCP taşımasının üzerinde çalışan HTTP ile HTTP Adaptif Akışlandırma (HAS) kullanır. DASH, istemci zekasıyla bir uygulama katmanı mekanizmasıdır. Kullandıkları ABR (Adaptif Bit Oranı) algoritmaları ve HTTP' nin avantajları, örneğin web önbellekleri ve kolay güvenlik duvarı geçişi sayesinde, HAS sistemleri bugünkü ağda en başarılı ve hâkim video akış sistemleridir. DASH sistemleri bu ek bileşenlere ve mekanizmalara ihtiyaç duyar ve yalnızca yerleşik TCP davranışına güvenemez. TCP'nin tıkanıklık kontrol algoritması, yavaş başlangıç mekanizması, yarışan TCP akışları, güvenilirlik için zorunlu gecikme gibi özellikleri HAS' in QoE' si üzerinde olumsuz etkiye neden olmaktadır. Bu nedenle, TCP'nin akış uygulamalarındaki olumsuz etkilerini ortadan kaldıran başka bir taşıma katmanı protokolü kullanıldığında, bu uygulamaların performansı artırılabilir ve bazı ek mekanizmalara artık ihtiyaç duyulmayabilir.

Videoyu iletmek için BPP kullanılmasının, video gönderen ve alan uygulamalar üzerinde olumlu yönde bir etkisi vardır. Yalnızca BPP' yi kullanmak, geliştirilmiş davranış veya performansı garanti etmemektedir. BPP en basit şekilde kullanılırsa, bir paketten bazı BPP parçaları düşebilir, bu da tüm paketin düşürülmesinden farklıdır. Eğer video blokları BPP parçalarına konulursa ve bu parçalar iletim sırasında atılırsa, alıcıda elde edilen video verisinde kayıp olacaktır. Bu kayıp, alıcı tarafından ele alınmalıdır ve bu, bir paketin kaybına çok benzer bir etkiye sahiptir. Eğer MPEG-DASH kavramları kullanılır ve farklı video segmentleri BPP paketlerinin parçalarına yerleştirilirse, her segment farklı bit hızları için veri içerebilir. Maalesef, DASH segmentleri genellikle 2 saniye uzunluğundadır ve bu, bir pakete sığdırılması gereken çok fazla veri olmasını gerektirir. DASH segment uzunluğu milisaniyelere indirilse bile, farklı bit hızlarında kodlamaların bir pakete eşleştirilmesi BPP iletimi için anlamlı olmayacaktır.

Video dekoderi, her video karesi için farklı bit hızı kodlamalarını almamaktadır. Bit hızı değişikliklerinin daha yavaş meydana geldiği ardışık çok saniyelik dizileri oynatmak için tasarlanmıştır. BPP, iletim sırasında bazı parçaları

paketlerden çıkarmak için özellikle tasarlanmıştır. BPP' nin aksine, tüm TCP tabanlı şemalar, verilerin bazı gecikmelerle ve yeniden iletimle garanti edilmesi için tasarlanmıştır. İşlevsel olarak, video verisi içeren BPP paketleri, ağ üzerinden iletimleri sırasında bir YTA denetleyicisi tarafından işlenir.

BPP paket küçültme işleminin uygulanabilmesi için ağ içerisinde BPP özelliğine sahip anahtarlama cihazlarının kullanılması gerekmektedir. Şu an BPP etkinleştirilmiş anahtarlar halihazırda mevcut olmadığı için tez çalışmasında ONOS YTA denetleyici ve OpenFlow etkin anahtarlar ile BPP işlemi gerçekleştirilmiştir. Çalışmada, YTA denetleyicisi BPP paketlerinin güncellemelerini / değişikliklerini belirlemiştir ve anahtarlar daha sonra denetleyici tarafından işlenen BPP paketlerini iletilmişlerdir. Bu işlemin gerçekleştirilmesi amacıyla ONOS içerisinde BPP\_APP isimli bir uygulama gerçekleştirilmiştir. Bu uygulama ONOS denetleyicisinin altyapısında mevcut bir uygulama olan Reactive Forwarding uygulamasından esinlenerek oluşturulmuştur. İlk geliştirilen yöntemde OpenFlow özellikli anahtarlama cihazına gelen tüm paketlerin ONOS denetleyiciye gönderilerek BPP işlemlerinin gerçekleştirilmesinden sorumlu olmuştur. Reactive Forwarding her akış için kalıcı (akış devam ettiği sürece) bir akış ataması gerçekleştirirken, oluşturulan bu uygulamada anahtara gelen her paket ONOS denetleyiciye yönlendirilmiş ve anlık bant genişliği değerinin kontrolü sonucunda paketin ne kadarının gönderileceğine karar verilmiştir (Clayman and Sayit., 2023)

Kodlanmış video dosyası bilgileri, hangi verilerin parçalar içinde taşınacağına ve parçaların öncelik değerlerinin (SIGi) nasıl belirleneceğine karar vermek için kullanılmaktadır. Katmanlı video iletilirken, her katmanın ve çerçeve türünün önemi, anlam değerlerinin belirlenmesinde önemli bir rol oynamaktadır. Bazı parçalar, kesinlikle silinmemesi gereken I tipi çerçeveleri gibi kritik unsurları taşımaktadırlar. Bu nedenle, en önemli parçalar, I çerçevelerinin temel katmanını(L0) taşıyan parçalardır. Öte yandan, P çerçevelerinin en yüksek katmanını (L2) taşıyan bir parça, alınan kaliteyi önemli ölçüde etkilemeden atılabilmektedir. Bu özellikler aynı zamanda paketten çıkarılmaması gereken parçaları temsil eden bir eşik değerinin belirlenmesinde de kullanılmaktadır (Clayman et al., 2021b).

Tez çalışması içerisinde kullanılan video dosyalarına ait video katmanları, çerçeve tipleri, video zaman katmanları hakkındaki bilgiler ve ilişkilendirilmiş öncelik değeri (SIG) Tablo 4.1' de verilmiştir. Test senaryolarında, çerçeve tipinin ve katmanın önemini dikkate alırken, her biri için bir öncelik değeri

ayarlanmaktadır. BPP paketlerindeki öncelik değerleri, her bir parçanın paket içindeki önemini göstermektedir. Bu değer, ağ düğümüne herhangi bir BPP işleminde parçayı tutma ya da çıkarma konusunda gereken bilgiyi sağlamaktadır.

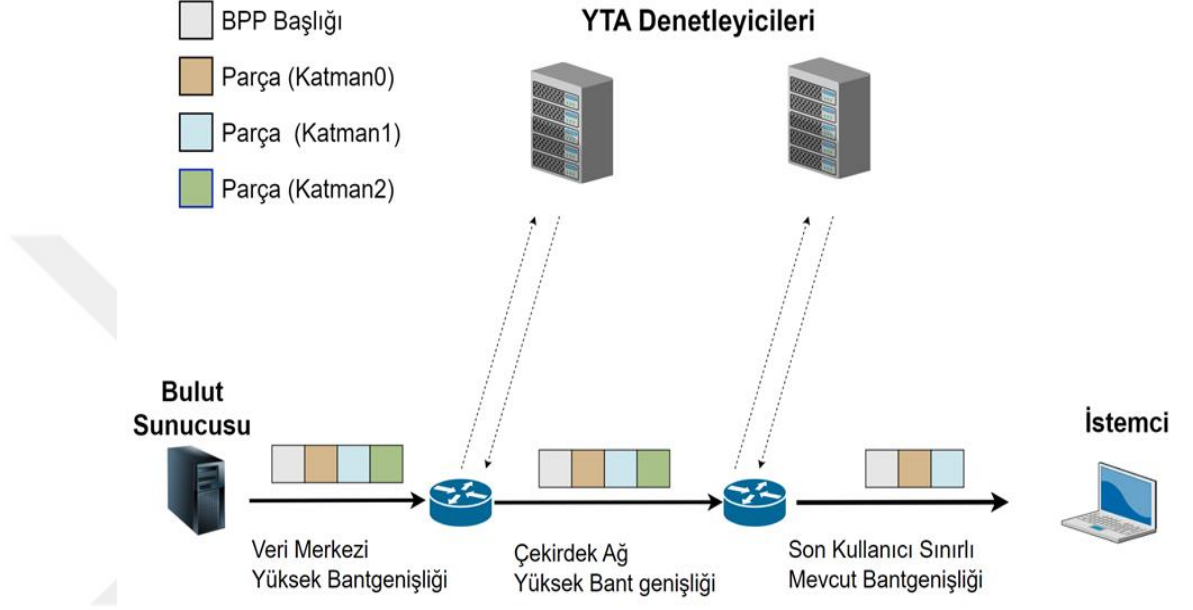
Tablo 4.1 Öncelik Değerleri Tablosu (Tuker et al., 2023)

Video Katmanı	Çerçeve Tipi	Zamansal Katman	Öncelik Değeri (SIGi)
L0	I	T0	1
L1	I	T0	2
L2	I	T0	3
L0	P	T1	2
L1	P	T1	7
L2	P	T1	12
L0	P	T2	3
L1	P	T2	8
L2	P	T2	13
L0	P	T3	4
L1	P	T3	9
L2	P	T3	14
L0	P	T4	5
L1	P	T4	10
L2	P	T4	15

Tablonun 4. sütununda verilen öncelik değeri, önceki bölümde Şekil 3.4' te verilmiş olan alanlar içerisinde SIGi alanına sunucu tarafından yerleştirilir ve i. parça için belirlenir. I tipi çerçeveleri (SIG 1, 2 ve 3 ile) ve P tipi çerçevelerin temel katmanları (L0) (SIG 1-5 ile) kesintisiz olarak videoyu oynatabilmek için istemciler tarafından her zaman alınmalıdır, bu nedenle Eşik Değeri (Threshold) 5 olarak ayarlanmıştır. Bu, mevcut bant genişliği yetersiz olduğunda ağ fonksiyonunun öncelik değeri 5' in üzerinde olan parçaları kırmasına izin vermektedir.

Paket Küçültme sürecini kullanarak iletim sırasında paketleri kırma amacıyla BPP kullanıldığı için, parça çıkarılma politikası hem parçanın video içeriğine olan önemini hem de mevcut bant genişliğini dikkate alarak yapılmaktadır.

#### 4.1. YTA İle Ağ İçi Video Akış Sistemi Uygulaması

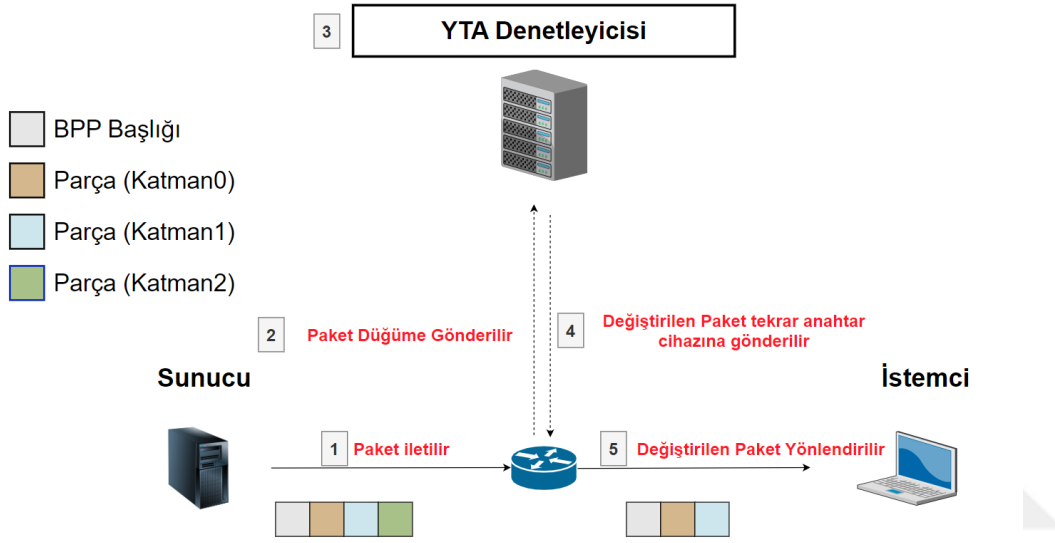


Şekil 4.1 YTA Denetleyicileri İle BPP İşlemi Yapılması (Clayman et al., 2021a)

BPP paketlerinin oluşturulması için sunucu tarafında, şekilde gösterilen paket doldurma yaklaşımı kullanılmaktadır. Şekil 4.1' deki gibi parçaların her biri 0, 1 ve 2 katmanlarından bir parça ile doldurulmaktadır ve paketlerin iletimi sırasında parça çıkarılmasına olanak tanınmaktadır. Bu amaçla, sunucu her bir çerçevenin her katmanını bölmekte ve bu bölümleri parçalar olarak kullanmaktadır. Her paket, her katmandan bölümleri taşımakta ve bazı katmanlar diğerlerinden daha büyük olabileceğinden, paketler içindeki parçalar yalnızca bir katmanın alt kümesini taşıyabilmektedir. Bu nedenle sunucudan gönderilen her bir pakette 3 parça bulunmaktadır. Paket Küçültme kararı verildiğinde paketten çıkarılabilecek bazı parçalar olmakta ve böylelikle işlemin gerçekleştirilebilmesi mümkün olmaktadır. Sunucu, belirtilen BPP paket alanlarının öncelik değerlerine dayalı olarak BPP paketlerini oluşturmaktadır. Koşul ve Eşik değerleri paket içerisinde ilgili alanlara konulmaktadır. Koşul ve paket uzunluğu değerleri paket başlığında tanımlanmış olsa da paket işleme adımları ve bu alanın semantiği (Li et al., 2019)' daki

çalışmadan farklıdır, çünkü bu çalışmadaki katmanlı video iletimi gereksinimlerinde farklılıklar bulunmaktadır.

BPP paket işlemini gerçekleştirmek için BPP yetenekli anahtarlar önerilmiştir (Li et al., 2018), fakat pratikte bu cihazlar mevcut değildir. Tez çalışmasında BPP yetenekli anahtarların görevini yerine getirebilmek için ilk geliştirilen yöntemde YTA denetleyicileri ve OpenFlow yetenekli anahtarlar kullanılmıştır.



Şekil 4.2 BPP İşleminin Gerçekleştirilmesi (Tuker et al., 2023)

Şekil 4.2, Paket Küçültme işleminin ağ düğümünde nasıl gerçekleştiğini göstermektedir. Belirli bir zaman diliminde transfer edilen toplam veri miktarı mevcut bant genişliğini aşarsa parçaların silinmesi işlemi gerçekleştirilmektedir.

1. adımda paketler, BPP başlığı ve bir dizi parça ile oluşturulmakta ve ağ boyunca iletilmektedir.

2. adımda paket bir ağ düğümüne geldiğinde, işlem için YTA denetleyicisine gönderilmektedir. YTA denetleyicisi, paketlerin boyutunu kullanarak her bir zaman diliminde transfer edilen bit sayısını hesaplamaktadır. Bu değer mevcut bant genişliğini aşarsa, paket küçültme işleminin yapılması gerektiğini belirlemektedir.

3. adımda YTA denetleyicisi, mevcut bant genişliği ölçümlerine göre paketleri değerlendirmek ve hangi parçaların paketten kırılması gerektiğini belirlemek için her bir paketi kontrol etmektedir. Parçalar, her bir parçanın anlam

değeri alanı dikkate alınarak sırayla işleme alınır, böylece bir paketin boyutu belirtilen sınırdan daha düşük olamaz. Önem değeri eşliğinden daha düşük olan parçalar herhangi bir durumda silinmemektedir, bu nedenle paket boyutu bazı durumlarda gerektiği kadar azaltılamayabilir. Bu durum, sonraki paketlerin daha fazla küçültülmesi gerekeceği anlamına gelmektedir.

4. adımda değiştirilmiş veya değiştirilmemiş paket anahtara geri gönderilmektedir.

5. adımda paket bir sonraki bağlantıya yönlendirilmektedir. İstemci, parçaları paketlerden toplama ve geçerli bir veri akışını yeniden oluşturma işlemini yerine getirmektedir.

OpenFlow uyumlu anahtarlama cihazlarının iletim tabloları başlangıçta boş olmaktadır. Yeni bir akış başladığında, OpenFlow uyumlu anahtarlama cihazı kendisine ulaşan ilk paketle birlikte ilgili iletim kuralını öğrenmek üzere YTA denetleyicisine mesaj atmaktadır. YTA denetleyicisi, düzenli olarak trafik ölçümleri gerçekleştirmekte ve yolların kullanılabilir bant genişliğini hesaplamaktadır. Sunucudan istemciye kullanılacak yol (maksimum bant genişliğine sahip yol kullanılacaktır) üzerinde bulunan anahtarlama cihazlarının çıkış hat kapasitesi ve saniyede gelen paket sayısına bakılarak (Paketin koşul alanında yer alan değer), her paketin paket boyutu için üst sınır hesaplanmaktadır. Paketler içerisinden parçaların silinme kararını denetleyici vermektedir. Bu amaçla, paket istenen boyuta düşürülene kadar parçaların önem sırasına göre sırayla silme işlemi gerçekleştirilmektedir.

Bir akış için ilk defa iletim kuralları belirlendiğinde yol üzerindeki cihazlara iletim kuralları gönderilmektedir. Eğer ağ üzerinde paket boyutlarının azaltılması gereken noktalar varsa, bu anahtarlama cihazlarına akış kuralı gönderilmemektedir. Böylece, bu anahtarlama cihazları ilgili akış için paket aldıklarında bu paketi kontrolcüye iletmekte ve paket değiştirildikten sonra iletimi gerçekleştirilmektedir. Paketlerin modifikasyon için YTA kontrolcüsüne gönderilmesi gecikmeye sebep olmaktadır, ancak BPP uyumlu cihazlar kullanıldığında bu işleme gerek kalmayacaktır. Kontrolcüde paket modifikasyonu yapılması sadece BPP' yi simüle etmek amaçlı gerçekleştirilmektedir.

Bu amaçla ONOS denetleyicisi içerisinde yeni bir uygulama yazılımı gerçekleştirilerek, BPP işlemlerinin denetleyici üzerinde gerçekleştirilmesi

sağlanmıştır. Video paketleri anahtarlar tarafından alındığında ilk olarak işlenmesi için YTA denetleyicisine gönderilir. YTA denetleyicisi, (i) paketin küçültülmesi gerekip gerekmediğini ve (ii) hangi parçaların paketten çıkarılması gerektiğini belirlemektedir. Değiştirilmiş paket geri anahtara gönderilir ve ardından anahtarın çıkış bağlantısına yönlendirilmektedir. YTA denetleyicisinin çeşitli modülleri bulunmaktadır, bunlar arasında bağlantılardaki etkin trafik miktarını ölçme ve yolun mevcut bant genişliğini hesaplama bulunmaktadır. Mevcut bant genişliği değerleri, anahtarlar tarafından periyodik olarak elde edilen trafik miktarı değerlerinin düzeltilmesiyle hesaplanmaktadır. YTA denetleyicisi, mevcut bant genişliği hakkındaki bilgilerini kullanarak paketleri değiştirmeye karar vermektedir. YTA denetleyicisi, belirli bir zaman diliminde aktarılan toplam veri miktarı mevcut bant genişliğini aştığında, bir paketin içindeki bazı parçaları atmaktadır. Bu zaman dilimi, Koşul alanındaki sunucunun sağladığı bilgiyle uyumlu olacak şekilde 1 saniye olarak seçilmektedir.

$$\text{Şart} = \frac{\text{Toplam Paket Sayısı}}{\text{Video Uzunluğu}} \quad (1)$$

Formül-1: Paket Küçültme İçin Gereken Şart

$$L > \frac{\text{Kullanılabilir Bant Genişliği}}{\text{Şart}} \quad (2)$$

Formül-2: Paket Küçültme İçin Gereken Boyut Kontrolü

Formül-1 de gösterilen Şart alanı bir saniyede iletilmesi gereken toplam paket sayısını ifade etmektedir. Bu değer iletilmesi gereken toplam paket sayısının, videonun saniye cinsinden toplam uzunluğuna bölünmesiyle elde edilmektedir. Denetleyici, her bir paket için Formül-2' de verilen koşulu kontrol etmektedir. Burada L paket uzunluğunu temsil etmektedir ve Kullanılabilir Bant Genişliği denetleyici tarafından ölçülen mevcut bant genişliği değeridir. Koşul gerçekleşirse, denetleyici tarafından paket küçültme işlemi gerçekleştirilmektedir.

Bir sonraki adım, paketten kaç parçanın ve hangi parçaların çıkarılması gerektiğini belirlemektir. Bir paketten çıkarılacak parça sayısı, her bir parçanın boyutuna ve öncelik değerine bağlı olmaktadır. Denetleyici, geçerli paketin boyutu için bir üst sınır değeri hesaplamakta ve ayrıca hangi parçaların silineceğini belirlemek için her bir paketi kontrol etmektedir. Bu üst sınır, denklem (2) deki L değerine eşittir. Paketten çıkarılacak parçaların sırası öncelik (SIG) değerlerine

dayanmaktadır. Bu amaçla, denetleyici, öncelik değerlerine göre parçaları azalan sırada liste halinde sıralamaktadır. Ardından, bu liste içerisinde en üstten başlanarak paket boyutu, sunucu tarafından belirlenen üst sınırların altına inene kadar bölgeleri çıkartılmaktadır. Öncelik değeri eşik değerinden daha düşük olan parçaların hiçbir şekilde silinmemesine dikkat edilerek bu işlem gerçekleştirilmektedir. Bu paket işleme adımları, katmanlı video iletimi gereksinimlerindeki farklılıklar nedeniyle (Li et al., 2018) çalışmasından biraz daha farklı şekilde gerçekleştirilmektedir.

## 4.2. Sanallaştırılmış BPP İşlemlerinin Kenar Ağda Uygulanması

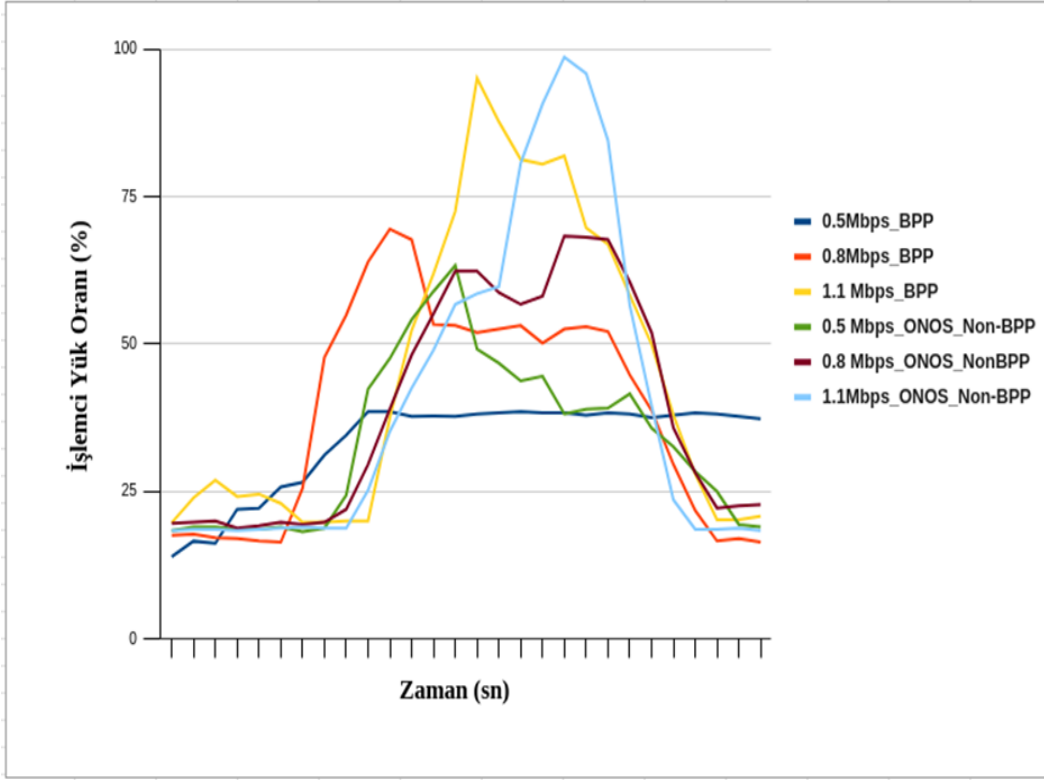
Gerçekleştirilen tez çalışmasında ağ operatörü ve video yayın şirketinin iş birliği yaptığı varsayılmaktadır. Böylelikle ONOS denetleyici sunucudan iletilecek videonun en yüksek video bit hızı hakkında bilgi almaktadır. Bu bilgi, denetleyicinin mevcut bant genişliğinin en yüksek kalitede videoyu iletmek için yeterli olup olmadığını belirlemesine yardımcı olmaktadır. Bir bağlantının mevcut bant genişliği, en yüksek kalitede videoyu iletmek için yeterli değilse, paketlerin o noktada küçültülmeleri gerekmektedir. Video akışlandırması sırasında, mevcut bant genişliği periyodik olarak ölçülmektedir. Bu işlemin gerçekleştirilmesi için de Halil Arasan'ın tez çalışmasında hazırlanmış olan "Akış Yöneticisi" isimli uygulamadan yararlanılmaktadır (Arasan, H., 2022). ONOS denetleyici ve Akış Yöneticisi uygulamaları REST API üzerinden birbirleriyle haberleşmektedirler. Tüm paketlerin ONOS'a gönderilmesi paket işleme sürelerinin artmasına neden olmuştur. Ağda yalnızca bir istemci bulunuyorsa, YTA denetleyicisi akışın her paketi için BPP işlemini yönetebilmektedir. Fakat istemci sayısı artırıldığında, denetleyici yüksek bir CPU kaynağı tüketmeye başlamakta ve sorunlar ortaya çıkmaktadır.

### 4.2.1. CPU yükünün ölçülmesi

Tez çalışmaları içerisinde BPP işlemlerinin denetleyici üzerindeki CPU'ya bindirdiği yükü görmek için CPULoad isimli bir uygulama oluşturulmuştur. CPU'ya binen yükü ölçmek için ONOS'un ilgili Process ID'si bulunarak programa parametre olarak verilmektedir. CPU yükü mümkün olduğunca hassas ölçülmeye çalışılmıştır fakat sistem içerisindeki periyodik olarak gönderilen LLDP vb paketlerin bile belli bir ölçüde CPU yükü üzerine etkileri bulunmaktadır.

Şekil 4.3'te çeşitli test senaryolarında BPP'nin devrede olması ve olmaması durumuna göre CPU Load değerleri görüntülenmiştir. Bunun için ilk olarak

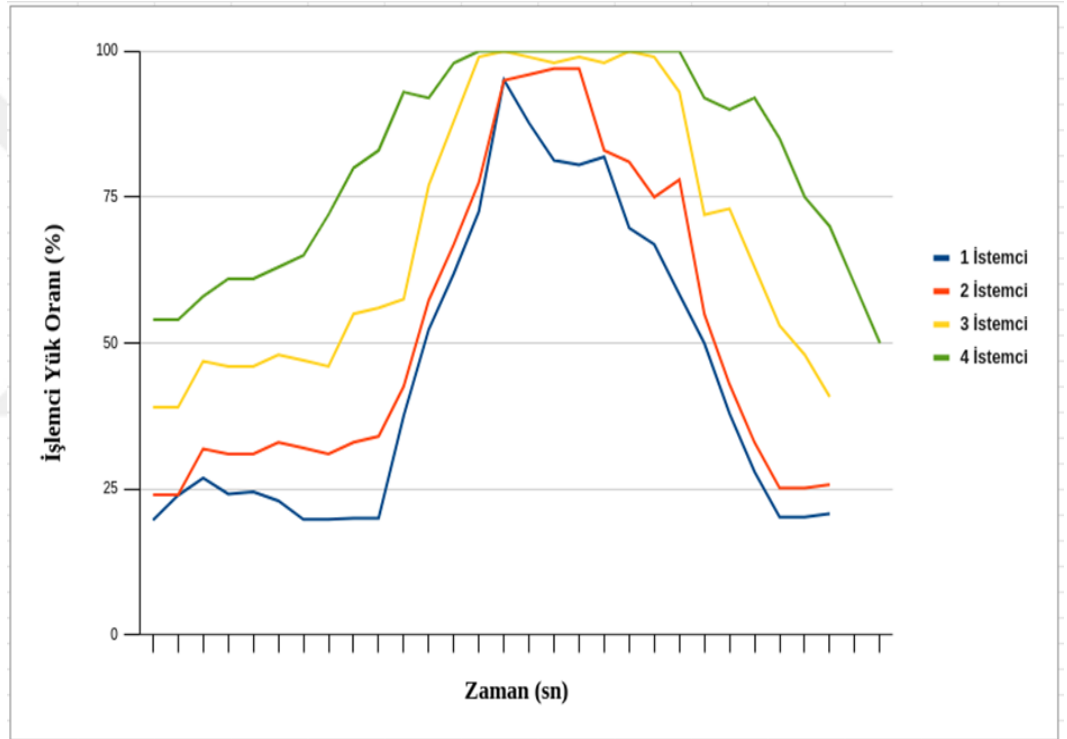
hazırlanan BPP\_APP uygulaması devredeyken testler gerçekleştirilmiştir. Bu testlerde BPP tam olarak çalışmakta bant genişliği durumuna göre de paket küçültme işlemleri gerçekleştirilmektedir. Bant genişliği artırıldıkça birim zamanda iletilen paket sayısı da arttığı için CPU üzerindeki yük hızlı bir şekilde yükselmektedir. Bu durum grafik üzerinde de açıkça görülmektedir.



Şekil 4.3 BPP İşlemlerinin İşlemci Üzerindeki Etkisi

İkinci olarak uygulama sınırlı anlamda devredeyken testler gerçekleştirilmiştir. Burada gelen paketlerin ONOS' a yönlendirilip akış ataması gerçekleştirilmekte fakat BPP işlemi devreye alınmamaktadır. Paketler sadece anahtar cihazından YTA denetleyiciye gönderilmekte ve geri alınmaktadır. Burada her paketin ONOS' a gidip gelme işleminin CPU açısından maliyeti görüntülenmektedir. Önceki senaryodan daha fazla yük çıkmaması BPP işleminin CPU üzerinde çok maliyet oluşturmadığını göstermektedir. Her iki durumda yaklaşık aynı üst seviyede CPU yükü oluşmasının sebebi bu yaklaşımda her paketin denetleyiciye gönderiliyor olmasından kaynaklanmaktadır. Şekil 4.3' teki grafikten de anlaşılacağı gibi tüm paketlerin YTA denetleyiciye gönderilmesi ve geri alınması işlemci açısından ek bir yük oluşturmaktadır.

Şekil 4.3' te gösterilen testlerde tek bir istemci ve farklı bant genişliği değerleri kullanılmaktadır. Ayrıca sabit bir bant genişliği belirlenerek birden fazla sayıda istemcinin işlemci üzerinde oluşturduğu yük de ölçümlenmiştir. Şekil 4.4' te sırasıyla 2, 3 ve 4 istemcinin BPP işlemlerinin YTA denetleyici üzerindeki işlemciye bindirdiği yük ölçümlenmiştir. İstemci sayısının artmasıyla işlemciye binen yük de artmaktadır. Kullanılan yöntemle 3 istemciye kadar iletim gerçekleştirilebilmektedir. 4 istemci için eş zamanlı iletim yapılmak istendiğinde işlemci maksimum noktaya ulaşmakta ve bu yüzden paket kayıpları ile hatalar oluşmaktadır. Yapılan deneylerde 4 ve daha fazla istemci için sağlıklı sonuçlar elde edilememektedir.



Şekil 4.4 Çok Sayıda İstemciye Ait BPP İşlemlerinin İşlemci Üzerindeki Etkisi

Ayrıca, bir anahtardan ONOS denetleyiciye tüm paketleri göndermek, denetleyici ile anahtar arasındaki iletim gecikmesinin uzamasına neden olarak uçtan uca gecikmenin artmasına yol açmaktadır. Bu noktada ONOS üzerindeki hesaplama karmaşıklığını sınırlamak ve uçtan uca gecikmeyi azaltmak amacıyla, her paketi göndermek yerine yalnızca bir BPP işlemi gerektiğinde denetleyiciye paket gönderecek bir yaklaşım geliştirilmiştir. Bu işlemi desteklemek için, bant genişliği sınırlı hale geldiğinde ONOS denetleyici, anahtarlama cihazı üzerindeki ilgili video akışı ile ilgili akış kuralını kaldırmaktadır. Bir anahtar cihazı bir paketi

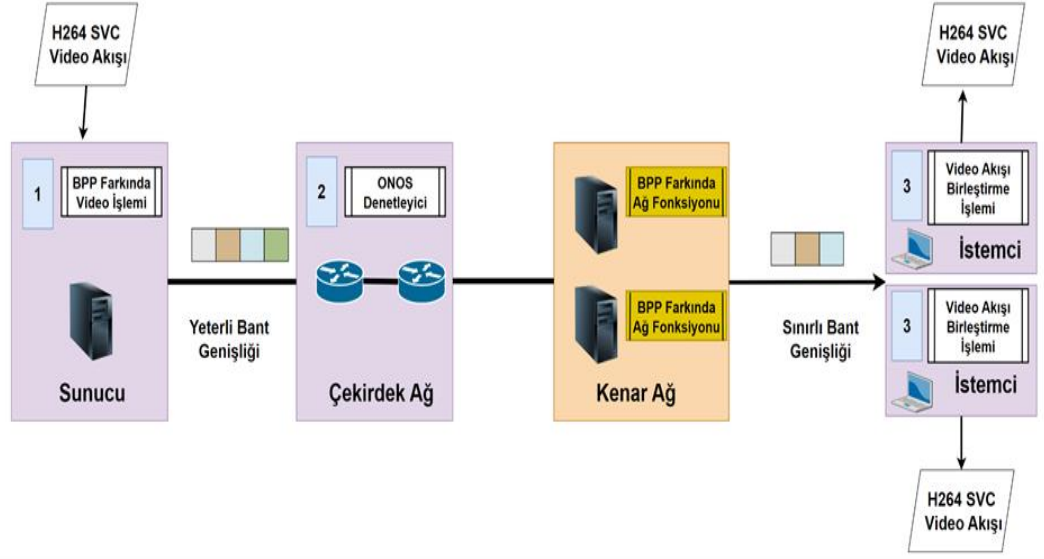
aldığında ve akış kuralları yoksa, video akışıyla ilgili akış bilgilerini istemek için ONOS denetleyiciye bir mesaj göndermektedir. ONOS denetleyici de yanıt olarak, küçültülmüş paketi ve çıkış bağlantı noktası bilgilerini içeren bir yanıt göndermektedir. Bu yaklaşım uygulanarak Mininet üzerinde çeşitli senaryolarla deneyler gerçekleştirilmiştir. Ancak deneyler, bu yaklaşımın da ölçeklenebilirliğinin oldukça sınırlı olduğunu göstermiştir. Çünkü veriyi işleme ve başlıkları kırpma işlemi Derin Paket İncelemesi (Deep Packet Inspection - DPI) işlemleri gerektirmektedir.

Denetleyicide BPP akışlarını işlemeye ilişkin tüm bu sorunlar göz önüne alındığında, bu sorunları ele almak için başka bir yöntem kullanmanın daha verimli olacağı görülmüştür. Yeni bir yaklaşım olarak BPP işlevlerini sanal bir ağ işlevi olarak uygulayıp ağ içi kalite adaptasyonu sağlamak verimliliği artırmaktadır. ONOS denetleyicisi ağ yönetimi için sistemde yine de yer almaktadır, fakat BPP işlemleri sunucu ve istemci arasına BPP yerleştirilen BPP-Server olarak isimlendirdiğimiz birim tarafından ağın kenar kısmında gerçekleştirilmektedir. BPP işlemleri için sanallaştırılmış BPP işlevi kullanılmaktadır. Bu yaklaşım daha önceki yaklaşımla karşılaştırma testlerine tabi tutularak elde edilen sonuçlar tez içerisinde verilmiştir ve ayrıca bir makale (Tuker et.al.,2023) yayını da gerçekleştirilmiştir.

#### **4.2.2 Sanallaştırılmış BPP işlemi**

Ağ ortamında genellikle bant genişliğinde daralma yaşanan noktalar, istemcilerin ağa bağlandığı kısımlarda yer almaktadır. Şekil-4.5' te görüldüğü gibi oluşturulan sistemde, sanallaştırılmış BPP farkındalığına sahip ağ işlevleri kenar anahtarların bulunduğu noktalara yerleştirilmiştir. Oluşturulan sistemde kenar ağ, ağı yönetme sorumluluğunu taşıyan bir YTA denetleyicisinin bulunduğu bir YTA etki alanını temsil etmektedir. YTA denetleyici, istemcilerin ağa bağlanmak için kullandığı bağlantıların mevcut bant genişliğini periyodik olarak ölçmektedir. Ölçülen mevcut bant genişliği bilgisi, BPP işlevine iletilmektedir.

Şekil 4.5' te gösterilen bu sistemde, sunucu videoyu en yüksek kalitede gönderirken, farklı bant genişliği koşullarına, cihaz çözünürlüklerine ve işleme yeteneklerine sahip farklı özelliklere sahip birçok istemcinin bulunduğu düşünülmektedir.



Şekil 4.5 Sanallaştırılmış BPP Yapısı (Tuker et al., 2023)

Videoyu taşıyan tüm katmanları içeren paketler, ağ üzerinden kenara kadar iletilmektedir. Kenarda, bu paketler sanallaştırılmış BPP işlevine gönderilmektedir. Burada, kenar anahtarlama cihazı ile istemci arasındaki geçerli bağlantı bant genişliği gibi parametrelere göre paketler üzerinde Paket Küçültme işlemi uygulanmaktadır. Paketler BPP işleminden geçirildikten sonra istemciye iletilmektedir. Sanallaştırılmış BPP işlevi bir paket aldığı anda, paketin küçültülmesi gerekip gerekmediğini ve eğer küçültülecekse paketten kaç birimlik alanın çıkarılması gerektiğini belirlemektedir. Sanallaştırılmış BPP işlevi, denetleyiciden alınan bilgilere dayanarak her istemcinin mevcut bant genişliği bilgisini kullanmaktadır. Paket Küçültme işlemi, önceki yöntemden farklı olarak Paket Küçültme Algoritmasında verilen basit bir hacim kontrol mekanizması kullanılarak yapılmaktadır.

#### 4.2.3. Paket küçültme için kullanılan algoritma

"Paket Küçültme Algoritması" adlı işlev her paket için, mevcut saniyede gönderilen bayt sayısının kullanılabilir bant genişliğinden büyük ya da küçük olup olmadığını kontrol etmektedir. Eğer gönderilen bayt sayısı kullanılabilir bant genişliğinin altındaysa yani hattın mevcut durumu paketin tamamının iletimi için uygun durumdaysa, paket olduğu gibi iletilmektedir. Ancak, gönderilen bayt sayısı kullanılabilir bant genişliğinden büyük olursa yani hattın bant genişliğinde bir daralma varsa, o zaman paketten belirli miktarda baytların çıkartılması gerektiği sonucuna varılmaktadır. Bu küçültme işlemi, paketten bazı parçaların kaldırılması yoluyla gerçekleştirilmektedir.

---

**Algorithm 1** Paket Küçültme Algoritması
 

---

```

// Şuan ki zamanın milisaniye cinsinden ayarlanması
1 baslangicZamani ← ZamanSayaci()
2 alinanBaytSayisi ← 0; gonderilenBaytSayisi ← 0

3 foreach paket almiyor do
4     ▷ Zamanlama Bilgisi Elde Etme
    // Milisaniye cinsindeki zamanın alınması
5     suankiZaman ← ZamanSayaci()
    // Şu anki süre ile başlangıç zamanı arasındaki fark
6     zamansalFark ← suankiZaman – baslangicZamani
    // Bu saniyedeki zamansal fark değerinin saniye cinsinden değeri
7     saniyeCinsindenFark ← zamansalFark / 1000

8     ▷ Paketin Ön İşleme
9     paketUzunlugu = uzunluk(paket)
10    alinanBaytSayisi += paketUzunlugu

    // Saniye cinsinden fark ile gönderilebilecek bayt sayısı
11    idealGonderilenBaytSayisi ← mevcutBantGenisligi * saniyeCinsindenFark
    // Alt değer ideal değerden ne kadar uzaktadır
12    altDeger ← idealGonderilenBaytSayisi – gonderilenBaytSayisi

13    ▷ Saniye Sınırlarını Kontrol Etme
14    if (zamansalFark >= 1000) then
    // Saniye sınırı ile karşılaşıldığında, yeni saniye için
    // değişkenlerin yeniden ayarlanması
15    baslangicZamani ← suankiZaman
16    saniyeCinsindenFark ← 0
17    alinanBaytSayisi ← 0; gonderilenBaytSayisi ← 0
18    end

19    ▷ Karar Verme ve İletme
20    if (altDeger > 0) then
    // Alt değer idealin altındadır, böylece kesintisiz iletim
    // kapasitesi olduğu anlamına gelir
21    alinanBaytSayisi += paketUzunlugu
22    ilet(paket)
23    else
    // Paketten birşeyler çıkarmak gerekir, bu sebeple içerikten
    // ayıklanır
24    (yeniPaket, cikarilanMiktar) = kucult(paket, idealGonderilenBaytSayisi, altDeger)
    // Çıkarılan miktar, gerçekte ne kadar içeriğin ayıklandığıdır
25    gonderilenBaytSayisi += paketUzunlugu – cikarilanMiktar
26    ilet(yeniPaket)
27    end
28 end

```

---

Şekil 4.6: Paket Küçültme Algoritması (Tuker et al., 2023)

Bu işlem daha önceki bölümde paketlerin öncelik sıralarında belirtildiği gibi, parçaların önem değerine göre yapılmaktadır. Çıkartılacak parçaların boyutu çıkartılması gereken bayt miktarından farklı olabileceğinden, süreç içerisinde tam

bir eşitlik yakalanması her zaman mümkün olmamaktadır. Eğer parçalar büyükse daha fazla bayt çıkartılabilir veya doğru bant genişliği seviyesine uygun hale gelmek için bir dizi paketin çıkartılması gerekebilmektedir. "BPP işlevi", video iletimi sırasında her saniye için aşağıdaki algoritmayı çalıştırmaktadır. Şekil 4.6' da görüldüğü gibi algoritmanın dört ana aşaması vardır ve algoritmanın temel aşamaları şu şekildedir:

**Zaman Bilgisi Elde Etme:** Her bir paketin varışı sırasında, algoritma paketlerin ne zaman alındığını takip etmek için zaman bilgisini elde eder. Bu zaman bilgisi belirli bir saniye içinde gönderilen bayt sayısını hesaplamak için kullanılmaktadır.

**Paket Ön İşleme:** Her gelen paket için algoritma, paketin boyutunu belirlemektedir ve bulunulan bir saniye içinde gönderilmesi gereken ideal bayt sayısını belirlemek için ön işlem yapmaktadır.

**Saniye Sınırlarını Kontrol Etme:** Algoritma, mevcut saniyeyi izleyerek bir saniye sınırının geçilip geçilmediğini belirlemektedir. Bu, algoritmanın her saniye için bant genişliği kısıtlamalarına uygun olduğundan emin olmak için önemlidir.

**Karar Verme ve İletme:** Son aşama, paketin olduğu gibi iletilip ileilmeyeceğine veya mevcut bant genişliği içine sığması için kırılıp kırılmayacağına karar vermeyle ilgilidir. Belirli bir saniye içinde belirli bir ofset için gönderilecek bayt sayısının mevcut bant genişliğinin altında olması durumunda paket değiştirilmeden iletilir. Ancak, paketi iletmek bant genişliğini aşacaksa, algoritma önem değerine göre hangi paket parçalarının kırılacağına karar verir. Bu kırpmalar, paketin bant genişliği kısıtlamalarına uygun hale getirilmesini sağlar.

Genel olarak, bu algoritma, kısıtlı bir bant genişliği içinde video paketlerinin iletimini yönetmek için dinamik bir yaklaşım kullanmaktadır. Zaman bilgisi, paket ön işleme, saniye sınırı kontrolü ve akıllı karar verme gibi unsurları bir araya getirerek mevcut kaynakların verimli bir şekilde kullanılmasını amaçlamaktadır. Eğer mevcut saniyede iletilen paketlerin hacmi, yani ulaşılan değer 0' dan büyük veya bant genişliğine eşitse, o zaman alınan paket, paket kırpmaya uygulanmadan istemciye iletilmektedir. Aksi takdirde paketten bazı parçaların düşürülmesi gerekecektir, bu yüzden kırpmaya işlemi gerçekleştirilmektedir. Trim () fonksiyonu, mevcut ideal gönderim hacmini, parçaları ve önem değerlerini dikkate almaktadır

ve bir paket ile düşürülen veri miktarını döndürmektedir. Bu işlem sonrasında değiştirilmiş veya değiştirilmemiş olan paketler istemcilere iletilmektedir.

#### 4.2.4. QoE hesaplama formülü

Uçtan uca video iletiminin gerçekleştirilmesinde farklı parametreler gözlemlenmekte ve ölçümlenmektedir. Ölçümlenen bu değerler arasında en önemli parametrelerden biri olarak istemci tarafında oynatılan videonun ortalama bit hızı (average bit rate) ele alınmaktadır. Video iletiminde TCP kullanılması durumunda, TCP istemcisi daima en yüksek kalitede ve tüm çerçeveleri içeren kayıpsız bir video dosyası elde etmektedir. Fakat, bu durum bir diğer kalite deneyim parametresi olan duraksama zamanının üzerinde negatif bir etki yaratmaktadır. İletilemeyen video paketlerinin tekrar iletilmesi sebebiyle yüksek duraksama sürelerinin oluşmasıyla sonuçlanmaktadır. Bir diğer kalite ölçümleme parametresi kalite değişim sayıları olarak değerlendirilebilir. Video iletiminde belirli bir an içerisinde gerçekleşen bir önceki çerçeve ile bir sonraki çerçeve arasında gerçekleşen katman kalitesi değişim sayılarının fazla olması QoE üzerinde negatif etki yaratmaktadır. Bunların dışında bir diğer önemli kalite ölçüm parametresi PSNR (Peak Signal To Noise Ratio) olmaktadır. PSNR bir sinyalin olası maksimum sinyal gücü ile, o sinyali etkileyen bozucu gürültünün gücü arasındaki oran olarak ifade edilmektedir.

$$QoE_1^K = \alpha \sum_{k=1}^K Q_k - \beta \sum_{k=1}^{K-1} |l_{k+1} - l_k| - \delta \sum_{k=1}^K dp_k$$

Formül 3: QoE Hesaplama (Tuker et al., 2023)

Tez çalışması içerisinde yapılan testlerde farklı kalite deneyim parametreleri ölçümlenmektedir. Elde edilen bu parametrelerin değerlendirilmesinde, algılanan kalite hakkında daha iyi bir fikir elde edebilmek için toplam QoE değeri isimli genelleştirilmiş bir değer daha oluşturulmaktadır. Bu değer hesaplanmasında testlerde elde edilen parametrelere ağırlıklar verilerek ve aşağıdaki formül kullanılarak sonuçlara ulaşılmaktadır. Toplam QoE değerini hesaplamak için, Formül-3 tarafından tasarlanan bir işleve dayanan doğrusal bir fonksiyon olarak tanımlanmaktadır (Tuker et al., 2023). Farklı QoE parametrelerinin önemi, algılanan kalite üzerindeki etkilere dayanarak farklı terimlere farklı ağırlıklar atanarak belirtilmektedir. Formül-3'de verildiği şekilde  $k$  ve  $K$  arasındaki bütün segmentler için toplam QoE değeri hesaplanması gerçekleştirilmektedir. Denklem içerisinde yer alan  $\alpha$ ,  $\beta$ ,  $\delta$  değerleri QoE parametreleri için kullanılan ağırlıklardır.

$Q_k$   $k'$  inci segmente ait video bit hızı cinsinden video kalitesini ya da PSNR değerini temsil etmektedir.  $l_k$   $k'$  inci segmentin katmanını ifade etmektedir. Bu sebeple, denklemde verilen 2. Terim art arda iki segment arasındaki kalite değişim seviyesini ifade etmektedir.

$$dp_k = \begin{cases} ts_k - p_k & ts_k \geq p_k \\ 0 & \text{aksi halde} \end{cases}$$

Formül 4: Duraksama Süreleri Hesaplama (Tuker et al., 2023)

Formül-4'deki denklemde hesaplanan  $dp_k$  değeri ise  $k'$  inci segmentin oynatılması sırasında oluşan duraksamaların toplam süresini ifade etmektedir. Ayrıca  $ts_k$  ve  $p_k$ ;  $k'$  inci segmentin alındığı zamanın zaman damgası değeri ile oynatma zamanlarını ifade etmektedir.

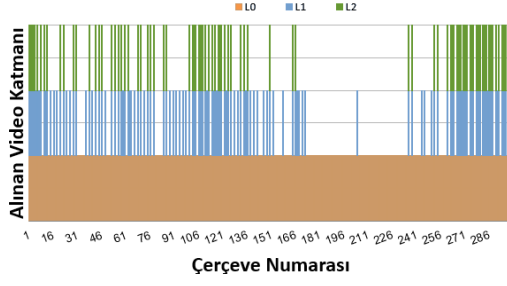
Tez çalışması içerisinde gerçekleştirilen deneyler için hesaplanan genel QoE değerleri bir sonraki bölümde gösterilecektir. QoE hesaplamasında, PSNR değeri Formül-3' teki  $Q_k$  değerleri olarak kullanılmaktadır. Denklemin parametre ağırlıkları  $\alpha$ ,  $\beta$  ve  $\gamma$  için sırasıyla 2, 0,01 ve 0,5 olarak belirlenmiştir. Bu sayılar, QoE parametrelerinin olumlu ve olumsuz etkilerine dayanarak seçilmiştir. (Seufert et al., 2015)' de algılanan kalite üzerine yapılan çalışmaya göre; videonun ortalama bit oranı değerinin QoE' yi etkileyen en önemli parametre olduğu ve kullanıcıların duraklama süresinden ziyade kalite değişikliklerinin daha az sayıda olmasının tercih edildiği gösterilmektedir. Bu bilgiler ışığında oluşturulmuş olan QoE hesaplama denklemi tez çalışması için yapılan ölçümlerde kullanılarak sonuçlarda gösterilmektedir.

## 5. DENEYSEL SONUÇLAR

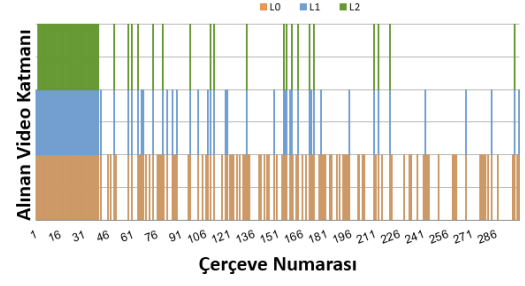
Tez çalışması içerisinde geliştirilen BPP Paket yapıları ve BPP işlem türleri için çeşitli karşılaştırma testleri gerçekleştirilmiştir. İlk olarak ONOS denetleyici içerisinde dahili bir uygulama olarak gerçekleştirilen BPP\_APP isimli uygulamanın UDP ve TCP ile karşılaştırılmaları yapılmıştır. Bu konuda yapılan çalışmalar ve testler ile ilgili olarak 2 adet Uluslararası Konferans Bildirisi yayınlanmıştır. Daha sonra BPP işleminin kenar ağa alınmasıyla ilgili testler gerçekleştirilmiştir. Bu konuda öncelikle iki yaklaşımın birbirleriyle karşılaştırılması yapılmıştır. Sonrasında ise geliştirilen yeni yaklaşımın diğer iletim protokolleri olan BPP, UDP ve DASH ile karşılaştırmaları yapılmıştır. Bu konularla ilgili olarak 3 adet Uluslararası Konferans Bildirisi yapılmıştır. Ayrıca tüm karşılaştırma ve geliştirmeleri içeren 1 adet Uluslararası Dergi Makalesi yayınlanmak üzere kabul almıştır. BPP paketleme stratejilerinin karşılaştırmalarını içeren bir adet Uluslararası Konferans Bildirisi de son olarak gönderilmiştir ve değerlendirme süreci beklenmektedir. Tezin bu bölümünde yapılan tüm testler açıklanacak ve sonuçları gösteren tablo grafikler ile desteklenecektir. Tez çalışması içerisinde yapılan performans değerlendirme testleri için, Yazılım Tanımlı Ağların oluşturulmasını ve test edilmesini sağlayan açık kaynaklı bir emülatör yazılımı olan Mininet platformu kullanılmıştır. Çalışmada YTA denetleyici olarak ONOS kullanılmıştır. ONOS, hizmet sağlayıcılar, cihaz üreticileri, internet operatörleri ve ONF (Open Network Foundation- Açık Ağ Vakfı) tarafından geliştirilen açık kaynaklı bir Ağ İşletim Sistemidir.

### 5.1 Ağ İçi Kalite Uyarlaması Kullanılarak Yapılan Karşılaştırma Testleri

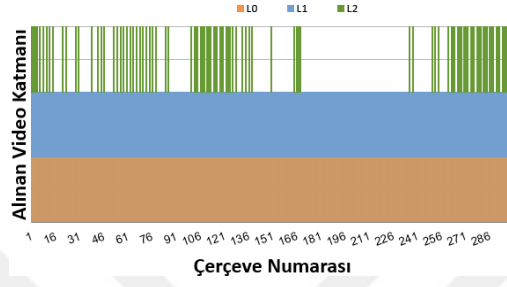
Bu kısımda yapılan testlerde BPP işlemleri için ilk geliştirdiğimiz yöntem kullanılmıştır ve (Clayman et al, 2021a; 2021b) yayınlarında yer almaktadır. ONOS içerisinde geliştirilmiş bir uygulama olan BPP\_APP ile paket küçültme işlemleri gerçekleştirilmiştir. ONOS' ta bulunan modülleri kullanarak, mevcut bant genişliği ölçümleri, paket akışlarının iletimi ve trafik istatistikleri toplanmıştır. BPP paketleri, BPP' nin Jumbo çerçeveleri taşıyabilmesine rağmen, standart 1500 Byte Ethernet paketlerine sığacak şekilde oluşturulmuştur. Testlerde kullanılan SVC ile kodlanmış video dosyası, 10 saniye uzunluğundadır ve 300 kare içeren "foreman" isimli video dosyası kullanılmıştır. Kodlanmış video dosyasında bir temel katman (L0) ve iki geliştirme katmanı (L1 ve L2) bulunmaktadır. Kullanılan video dosyasının katmanlarına ait bit akış hızı değerleri şu şekildedir: Temel katman (L0) 204 Kbps, L1 katmanı 488 Kbps, L2 katmanı 1094 Kbps' dir. BPP' nin gerçek bir uygulamasında, daha yüksek bit hızlarına sahip video dosyalarını iletmek mümkündür.



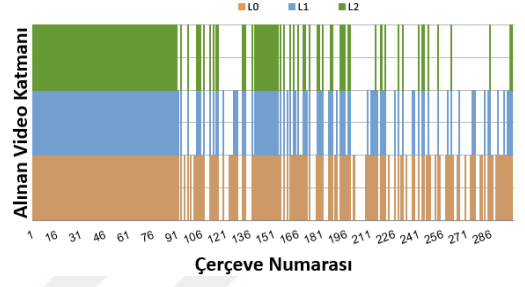
a) 0,5 Mbps BPP Alınan Video Katmanları



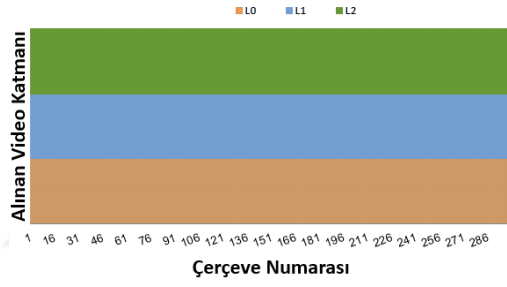
b) 0,5 Mbps UDP Alınan Video Katmanları



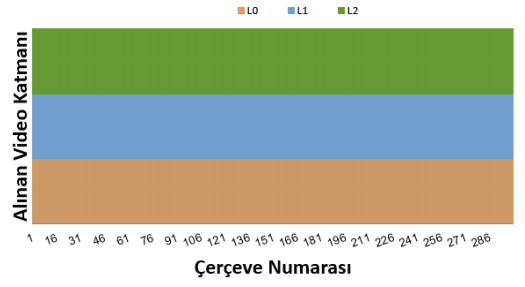
c) 0,8 Mbps BPP Alınan Video Katmanları



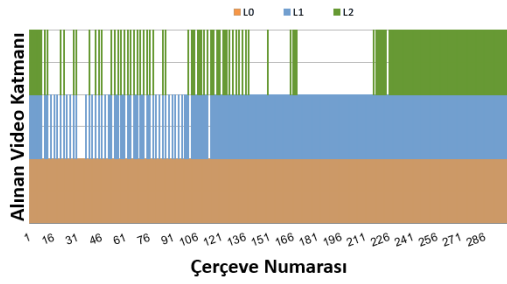
d) 0,8 Mbps UDP Alınan Video Katmanları



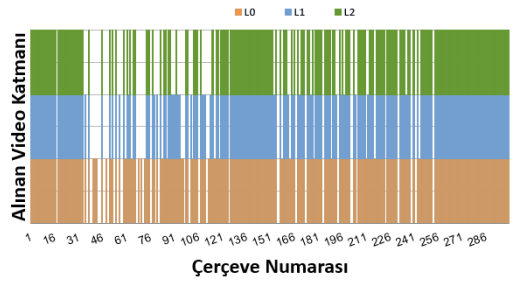
e) 1,5 Mbps BPP Alınan Video Katmanları



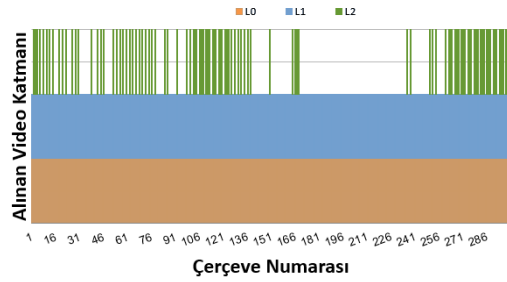
f) 1,5 Mbps UDP Alınan Video Katmanları



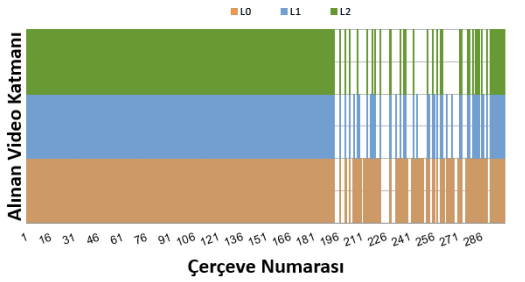
g) Artan Bw BPP Alınan Katmanlar



h) Artan Bw UDP Alınan Katmanlar



i) Azalan Bw BPP Alınan Katmanlar



j) Azalan Bw UDP Alınan Katmanlar

Şekil 5.1 Değişik Bant Genişliği Değerlerinde Alınan Çerçeveler (Clayman et al, 2021a)

Performans sonuçları istemciler tarafından alınan ham paket verileri ölçülerek elde edilmiştir. Ayrıca, katmanlı video paketlerini iletmek için UDP ve TCP' yi kullanarak testler gerçekleştirilmiştir. BPP, UDP ve TCP' nin elde edilen performans sonuçları karşılaştırılmıştır. İlk test serisinde, ağ üzerindeki hatların kullanılabilir bant genişliği sabit tutulmuştur. İkinci test setinde, iperf ile karşı trafik oluşturularak kullanılabilir bant genişliği dinamik olarak değiştirilmiştir. UDP ve TCP paketleri oluşturulurken, paketin taşıyabileceği maksimum katman sayısı ile paketler doldurulmuştur. Katmanlar, temel katmandan başlayarak sırayla paketlere yerleştirilmiştir. BPP paketlerinin yapısı ve yükü önceki bölümde tanımlandığı şekilde gerçekleştirilmiştir. İstemci tarafında alınan ham verinin yanı sıra, PSNR (dB cinsinden) ve toplam kesinti süresi (ms cinsinden) olmak üzere iki önemli QoE parametresinin daha ölçümü gerçekleştirilmiştir.

Şekil 5.1' de verilen grafikler, üç farklı sabit bant genişliği değeriyle (0,5 Mbps, 0,8 Mbps ve 1,5 Mbps) ve artan-azalan bant genişliği değerleriyle istemci tarafında alınan katmanları, çerçeve numarasına göre sunmaktadır. Artan ve azalan bant genişliği için yapılan testlerde 0,5 Mbps ile 1,5 Mbps arasındaki değerler kullanılmıştır. BPP istemcileri tarafından alınan katmanlar tüm grafiklerde sol tarafta, UDP istemcileri tarafından alınanlar ise tüm grafiklerde sağ tarafta verilmektedir. Güvenilir mekanizması nedeniyle, TCP istemcileri her zaman tüm katmanları almaktadır, bu nedenle burada TCP ile ilgili grafikler çizilmemiştir. BPP ve UDP' nin sonuçlarını karşılaştırmalı olarak incelediğimizde, 0,5 Mbps, 0,8 Mbps, Artan ve Azalan bant genişliği ayarları için, BPP her zaman temel katman paketlerini iletmekteyken, UDP sınırlı bir süre için en yüksek kalitede video ile yayını başlatmakta ve ardından kaliteyi düşürmektedir, böylece birçok çerçevenin temel katmanı istemciye ulaşmamaktadır. İki protokol de, bant genişliği sınırlı olsa bile, zaman zaman zenginleştirme katmanı paketlerini iletmektedir, çünkü kullanılabilir bant genişliği bulunmaktadır ve bazı zenginleştirme katmanlarının boyutları bant genişliğine uygun boyutta olmaktadır. Bant genişliği 1,5 Mbps değerine eşit olduğunda, her iki protokol de tüm çerçevelerin tüm katmanlarını iletmektedir, çünkü ağın kapasitesi tüm videoları en yüksek kaliteyle iletmek için yeterli bir bant genişliğine sahip olmaktadır. Artan ve azalan bant genişliği testlerinde de bu durum açıkça görülmektedir. Bant genişliği 1,5 Mbps civarında iken her iki tür istemci de tüm katmanlardan paket alımı gerçekleştirmektedir. BPP kontrolcü yaklaşımı ile UDP' ye göre daha stabil bir sonuç elde etmekte ve katmanları sırasıyla almaktadır.

Şekil 5.1' de sunulan grafikler, istemci tarafından alınan ham katmanları temsil eder, ancak video çözücüyü beslemek için tamamı işlenemez. SVC videosu, ardışık çerçeveler arasındaki benzerlikleri kullanarak kodlanmaktadır, aynı

zamanda aynı çerçevelerin katmanları arasındaki bağımlılığı kullanarak da kodlanmaktadır. Dolayısıyla farklı çerçevelerin aynı katmanları arasında bir bağımlılık bulunmaktadır. Birçok çerçeve, diğer çerçeveleri referans almaktadır ve düzgün bir çözümleme için bir çerçevenin tüm referans çerçevelerinin istemciler tarafından alınması gerekmektedir. Bu nedenle, bu ham katmanlar oynatma kabiliyetini tam olarak yansıtmamaktadır.

Tablo 5.1 Ağ İçi BPP QoE Metriklerinin Karşılaştırılması (Clayman et al, 2021a)

Bw	BPP	UDP	TCP	Bw	BPP	UDP	TCP	Bw	BPP	UDP	TCP
0,5 Mbps	173	5470	14396	0,5 Mbps	40	29	44	0,5 Mbps	10	2,4	10
0,8 Mbps	205	2634	5697	0,8 Mbps	40	36	44	0,8 Mbps	10	4,2	10
1,5 Mbps	0	0	584	1,5 Mbps	44	44	44	1,5 Mbps	10	10	10
Artan	0	916	2554	Artan	39	18	44	Artan	10	1,7	10
Azalan	0	920	4691	Azalan	40	37	44	Azalan	10	6,7	10

a) Kesintilerin süresi (ms)

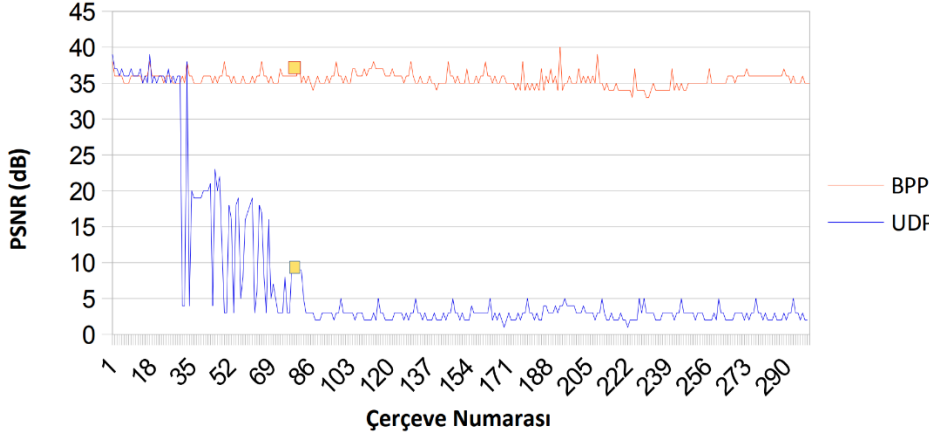
b) Ortalama PSNR değerleri (dB)

c) İstemci tarafında videonun oynatılma süresi (sn)

Tablo 5.1a' da, üç protokol için farklı bant genişliği ayarlarında gözlemlenen duraksama süreleri sunulmaktadır. Duraksama süresi, videonun algısal kalitesini etkileyen en önemli QoE ölçütlerinden biri olarak kabul edilmektedir. Tablodaki duraksama sürelerine ait değerler incelendiğinde, BPP kullanan istemcilerin önemli bir duraksama yaşamadığı görülmektedir, en yüksek duraksama 0,2 saniye (200 milisaniye) dir. Diğer yandan, bant genişliği 0,5 Mbps iken UDP ile 5,47 saniyeye kadar, TCP ile 14,4 saniyeye kadar duraksama gözlemlenmektedir. Videonun süresinin sadece 10 saniye olması göz önünde bulundurulduğunda, bu kadar uzun duraksama sürelerinin algısal kalite üzerinde yüksek bir olumsuz etkisi bulunacaktır. UDP istemcileri ayrıca 0,8 Mbps ile 2,6 saniye, dinamik bant genişliği senaryoları ile 0,9 saniye duraksama yaşamaktadır. Ancak en yüksek duraksama değerleri TCP testlerinde gözlemlenmektedir. Bunun iki ana nedeni bulunmaktadır: tüm katmanlar istemcilere gönderilmekte ve tüm kaybolan paketler yeniden iletilmektedir. Bant genişliği 1,5 Mbps iken bile TCP testinde 0,5 saniye kesinti gözlemlenmektedir.

Orijinal kodlanmış video dosyasının PSNR değeri 44 dB' dir ve alınan ortalama PSNR değerleri Tablo 5.1b' de gösterilmektedir. Bant genişliği 1,5 Mbps olduğunda, tüm protokoller için veri kaybı olmamasından dolayı istemcilerin videoyu en yüksek olası kaliteyle oynattığı görülmektedir. Diğer bant genişliği ayarları için, BPP istemcileri, videoyu her zaman 39 dB' den yüksek bir PSNR

değeriyle oynattığı için videonun kalitesi her zaman yüksek olmaktadır. UDP, tüm senaryolarda tutarlı ve iyi bir kalite sağlayamamaktadır. Artan bant genişliği senaryosunda, UDP' nin ortalama PSNR değerinin 18 dB olduğu gözlemlenmektedir. Bu da video kalitesinin kabul edilebilir bir seviyede olmadığını göstermektedir. TCP istemcileri her zaman en yüksek PSNR değeriyle videoları büyük kesintilerle oynatmaktadır. Tablo 5.1c, istemci tarafında oynanan videonun toplam süresini göstermektedir. BPP ve TCP istemcileri tüm video dizisini oynatabilmektedir. Şekil 5.1a ve 5.1c için grafiklere bakıldığında, istemcilerin her ikisinin de akış oturumunun sonuna kadar video katmanları almaya devam ettikleri görülmektedir. Ancak UDP istemcisinin birkaç saniye sonra videoyu oynatamadığı görülmektedir.



Şekil 5.2 BPP ve UDP İçin PSNR Değerleri (Clayman et al, 2021b)



(a) Çerçeve 81: BPP İstemcine ait videonun ekran görüntüsü

(b) Çerçeve 81: UDP İstemcine ait videonun ekran görüntüsü

Şekil 5.3 BPP ve UDP Oluşan Video Dosyası Örneği (Clayman et al, 2021b)

SVC video, ardışık çerçeveler arasındaki benzerlikleri kullanarak kodlanmıştır, aynı zamanda aynı çerçevelerin katmanları arasında bir bağımlılık bulunmaktadır. Ayrıca farklı çerçevelerin aynı katmanları arasında da bir bağımlılık bulunmaktadır. Çoğu çerçeve diğer çerçevelere referans olmaktadır ve doğru çözümleme için bir çerçevenin tüm referans çerçeveleri istemciler tarafından alınmalıdır. Şekil 5.2' de, bant genişliği 0,5 Mbps olduğunda hem BPP hem de UDP için PSNR- Çerçeve Numarası grafiği verilmiştir. Görüldüğü üzere, BPP istemcisi tüm çerçeveler için videoyu sabit bir kalitede oynatırken, UDP istemcisi sadece 81. çerçeveye kadar videoyu çözümleyip oynatabilmektedir. Ayrıca, önceki çerçevelerin PSNR değerleri çok değişken bir akış sergilemektedir. Bu da kesintisiz bir oynatma gerçekleştirilemediğini göstermektedir.

Şekil 5.3a ve 5.3b görsellerinde, BPP ve UDP istemcilerinin oynattığı aynı çerçeveler gösterilmektedir. Bu çerçevelerin video akışındaki konumları Şekil 5.2' de sarı bir dikdörtgenle gösterilmektedir. Eksik referans çerçevelerinin etkisi, UDP istemcisinin ekran görüntüsünden görülebilmektedir. Bu testte, UDP istemcisi 4 saniye boyunca video gösterebilmekte ve Şekil 5.3b 'deki kare video sonuna kadar ekranda kalmakta ve videonun kalanı oynatılmamaktadır (Clayman et al, 2021b). Oynatılmama nedeni, bant genişliği sınırlamaları nedeniyle belirli bir noktadan sonra diğer çerçevelere referans olan tüm ana çerçevelerin kaybedilmesidir. Bu durum sonucunda UDP için, istemcilere ulaşan çoğu katmanın iletiminde kullanılan ağ kaynaklarının israf edildiği görülmektedir. BPP ve YTA' yı birlikte kullanarak ağ yönetmenin, iletim sırasında kaliteyi ayarlayarak ve yeterli kapasite olduğunda sadece üst katmanları göndererek daha yüksek etkili bant genişliği kullanımı sağladığı sonucuna varılmaktadır (Clayman et al, 2021a).

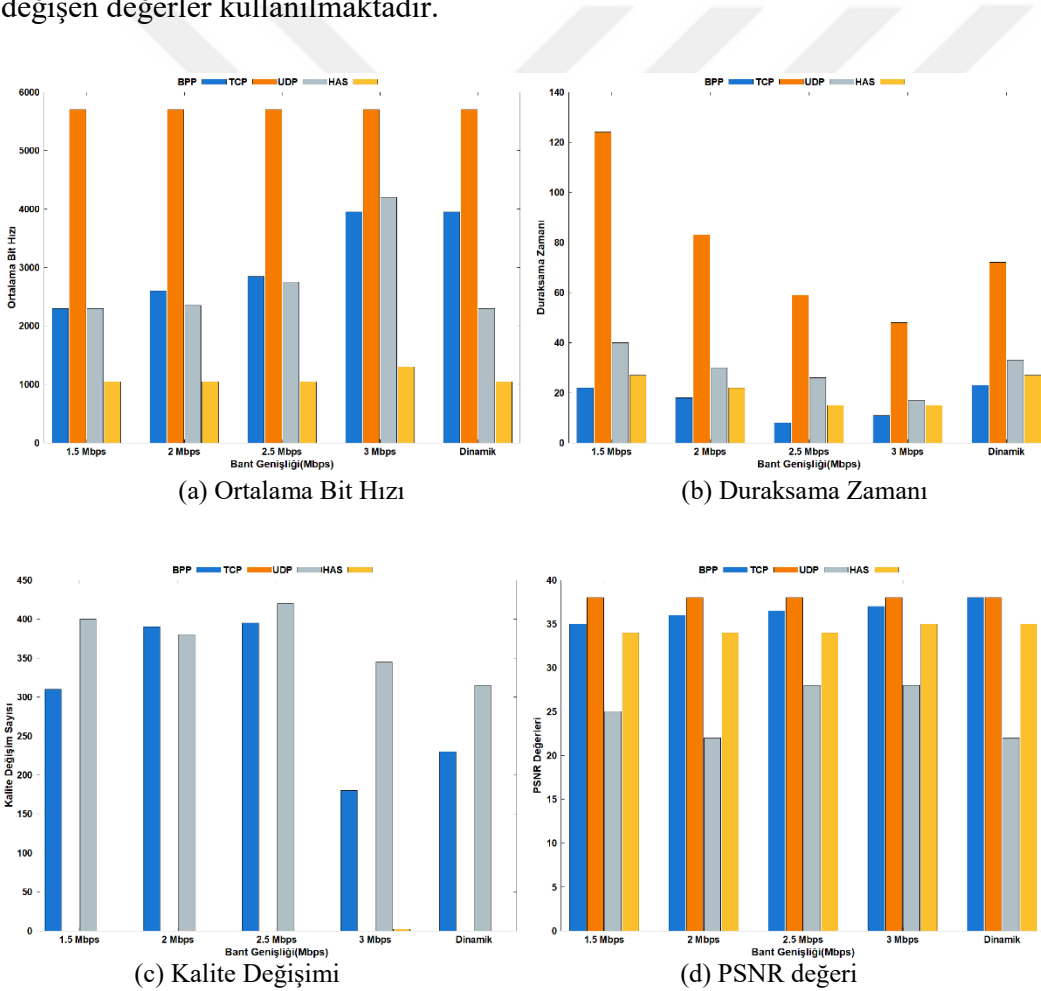
## 5.2 Ağ İçi Kalite Uyarlaması ile HAS Karşılaştırması

Ağ içi kalite uyarlamasının potansiyel avantajlarının gösterilmesi için video akışlandırmada popüler bir yöntem olan HTTP Adaptif Akışı (HAS) ile de karşılaştırma testleri gerçekleştirilmiştir (Tuker et al, 2023). Böylelikle önerilen BPP yaklaşımının beklenen faydalara sahip olup olmadığının belirlenmesi hedeflenmiştir. İlk olarak düşünülen BPP yaklaşımı, (Clayman et al, 2021a; Clayman et al, 2021b) çalışmalarında belirtildiği gibi, BPP paket küçültme sürecini uygulamak için bir ONOS YTA denetleyici kullanılmakta ve bu fonksiyon ONOS denetleyicisi içerisinde bir uygulama üzerinde gömülü olarak çalıştırılmaktadır.

Tüm deneylerde kullanılan video, Big Buck Bunny' dir. Video kaliteleri için bit hızı dağılımı, kodlama yönlerinden dolayı HAS ve diğer protokoller için biraz farklıdır. HAS deneyleri için, (Lederer et al., 2012) içinde verilen HAS sistemleri

için paketlenmiş video dosyası kullanılmıştır. HAS ile kullanılan paketlenmiş kodlanmış videonun kalite bit hızları 1 Mbps, 2 Mbps ve 4,2 Mbps' dir. BPP, UDP ve TCP kullanılarak gerçekleştirilen diğer deneylerde, SVC kodlanmış video katmanlarının bit hızları sırasıyla L0: 1,1 Mbps, L1: 1,9 Mbps ve L2: 5,6 Mbps' dir. Ancak, bu bit hızları paketleme sürecinden sonra bir miktar daha artmaktadır. HAS deneylerinde adil bir karşılaştırma sağlamak için paketlenmiş videonun bit hızına uygun olarak bant genişliği değerleri de azaltılmıştır.

Gerçekleştirilen deneylerde, çekirdek ağ ile istemci arasındaki yolun bant genişliği için sabit ve dinamik değerler belirlenmiştir. Sabit bant genişliği testleri için sırasıyla 1,5 Mbps, 2 Mbps, 2,5 Mbps ve 3 Mbps değerleri kullanılmaktadır. Dinamik bant genişliği testleri için ise zamana göre 1 Mbps ile 3 Mbps arasında değişen değerler kullanılmaktadır.



Şekil 5.4 ONOS Kontrolör Üzerinden Gerçekleştirilen Test Sonuçları (Tuker et al, 2023)

Şekil 5.4' te, farklı protokoller tarafından gözlemlenen farklı QoE parametreleri sunulmaktadır. Grafiklerde BPP için mavi, TCP için turuncu, UDP

için gri ve HAS için sarı renk kullanılmıştır. Şekil 5.4a’ da, istemci tarafında oynatılan videonun ortalama bit hızı verilmektedir. Bu QoE parametresi açısından, TCP her zaman en yüksek kalitede video aldığı için diğer yaklaşımlardan daha iyi performans gösterir. Ancak Şekil 5.4b’ de görüldüğü gibi bu davranışın yüksek düzeyde duraksama süresi gibi QoE üzerinde büyük olumsuz bir etkisi olan özelliklerle önemli maliyetleri vardır. Tasarımı nedeniyle, HAS istemcisi duraksama süresini minimumda tutmak için her zaman en düşük kalitede video talep eden korumacı bir yaklaşım uygulamaktadır. TCP ve HAS uygulamaları zamanla kaliteyi değiştirmedikleri için Şekil 5.4c’ de gördüğümüz gibi, bu deneylerde neredeyse hiç gözlemlenebilir kalite değişikliği yoktur, oysa BPP ve UDP’ nin birçok değişikliği vardır. Şekil 5.4d’ de her protokol için gözlemlenen PSNR değerleri verilmiştir. TCP’ nin tüm oturum boyunca sunucunun en yüksek kalitede video göndermesi nedeniyle en yüksek PSNR değerine sahip olduğunu görüyoruz. BPP ve HAS için PSNR değerleri benzerdir, burada BPP ile elde edilen PSNR değerleri HAS’ ten biraz daha yüksektir.

Tablo 5.2 Ağ İçi BPP QoE Değerlerinin Karşılaştırılması (Tuker et al, 2023)

	<b>Bant Genişliği</b>				
	<b>1,5 Mbps</b>	<b>2 Mbps</b>	<b>2,5 Mbps</b>	<b>3 Mbps</b>	<b>Dinamik</b>
<b>BPP</b>	57,7	59,5	63,4	68,8	63,7
<b>TCP</b>	14,8	35,8	48,3	53,8	41,8
<b>UDP</b>	23,5	34,7	41,2	46,4	24,1
<b>HAS</b>	56,9	59	62,7	64,9	56,9

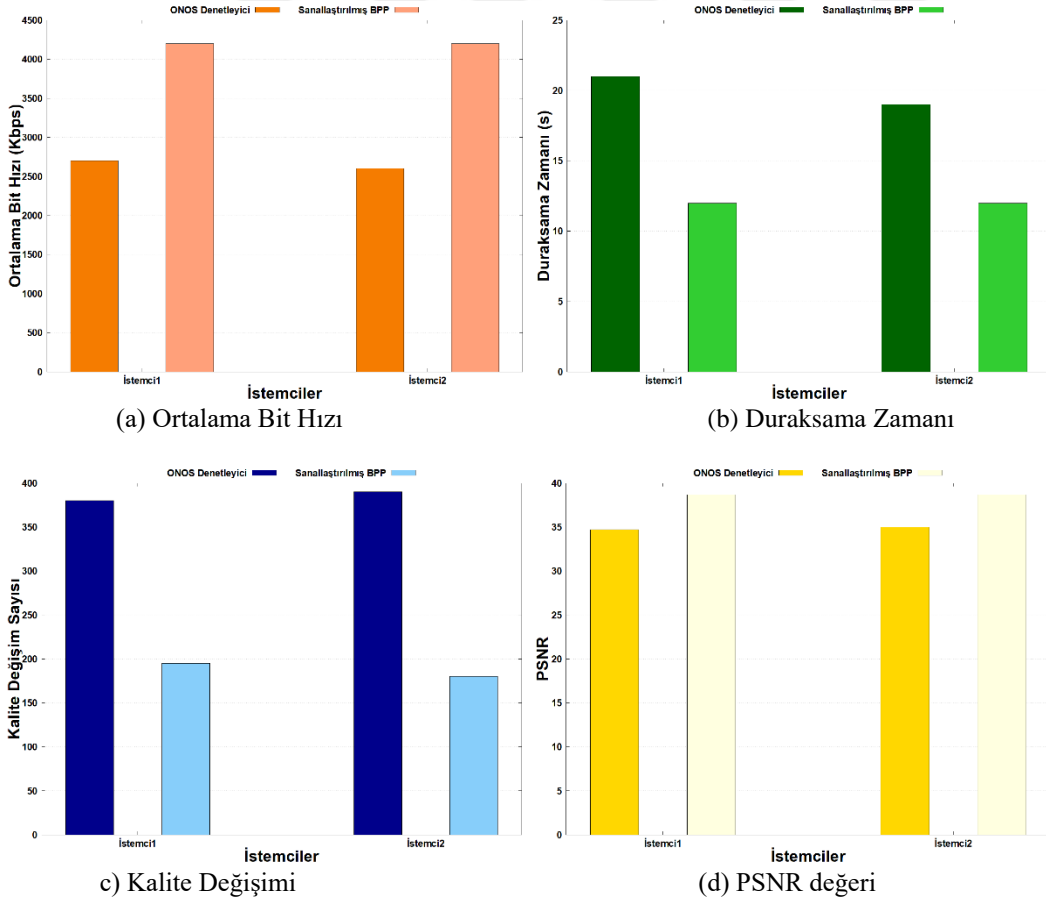
Tablo 5.2, sabit bant genişliği deneylerinde en yüksek genel QoE değerlerinin BPP ile gözlemlendiğini, ancak HAS değerlerinin BPP değerlerine çok yakın olduğunu göstermektedir. Dinamik bant genişliği koşullarında, BPP’ nin diğer yaklaşımları geride bıraktığı ve ağ içi kalite adaptasyonunun istemci tabanlı adaptasyona kıyasla ne kadar faydalı olabileceği konusunda bir fikir verdiği gözlemlenmektedir.

### 5.3. Sanallaştırılmış BPP Yöntemi ile Ağ İçi BPP Yönteminin Karşılaştırılması

Şekil 5.5’ te, Paket Küçültme işlemini gerçekleştirmek için iki farklı yaklaşım arasındaki karşılaştırma sunulmaktadır. Bu amaçla, video sunucusu ile 2 adet istemci arasında video akışlandırma gerçekleştirilmekte ve sonuçlar elde

edilmektedir. Gerçekleştirilen testler sonucunda istemci tarafında ölçülen QoE parametreleri toplanmaktadır. Oluşturulan grafiklerde "ONOS Denetleyici" olarak adlandırılan koyu renkli çubuklar, ilk yöntem olarak geliştirilmiş olan yaklaşıma aittir. BPP işlemini yürüten bir modülü çalıştıran bir ONOS denetleyici kullanıldığında gözlemlenen sonuçları temsil etmektedir.

"Sanallaştırılmış BPP sunucusu" olarak adlandırılan daha açık renkli çubuklarda, ikinci yöntem olarak geliştirilmiş olan yaklaşıma aittir. BPP işlemi ağır kenar kısmında, istemciye yakın noktada sanallaştırılmış bir işlem tarafından gerçekleştirilmektedir. Bu testlerde, 2 istemci farklı kenar bağlantıları üzerinden ağa bağlıdır ve bu bağlantılardaki kullanılabilir bant genişliği 2,5 Mbps' dir. Paket Küçültme işlemi sanallaştırılmış BPP işlevi tarafından gerçekleştirildiğinde, tüm QoE parametreleri, Şekil 5.5' te görüldüğü gibi, ONOS denetleyici ile yapılan testlere kıyasla daha iyi değerlere sahip olmaktadır.



Şekil 5.5 Ağ İçi ve Sanallaştırılmış BPP Yöntemlerinin Karşılaştırılması (Tuker et al, 2023)

Buna ek olarak, istemciler videoyu daha yüksek kalitede almasına rağmen, sanallaştırılmış BPP sunucusuyla gözlenen kesilmelerin süresi, Şekil 5.5b' de

görüldüğü gibi denetleyici tarafındaki BPP işlemi tarafından eklenen gecikme nedeniyle hala ONOS denetleyicisiyle kıyaslandığında daha azdır. Ayrıca Şekil 5.5b’de görüldüğü gibi istemcilerin daha yüksek kalitede video almasına rağmen sanallaştırılmış BPP sunucusuyla yaşanan duraksamaların süresi ONOS denetleyicisiyle kıyaslandığında daha kısa olmaktadır. Bunun sebebi denetleyici tarafındaki BPP işlemi tarafından eklenen gecikme değerleridir. Genel olarak, kenarda paketleri küçülterek elde edilen QoE, ONOS denetleyicisi kullanılarak elde edilen QoE’ ye kıyasla daha iyi olmaktadır. Çünkü kenarda sanallaştırılmış BPP işlevlerini kullanmak, ONOS denetleyicisiyle elde edilenlere göre daha yüksek QoE parametreleri sunmaktadır (Tuker et al, 2023).

#### 5.4. Sanallaştırılmış BPP Yöntemi Performans Testleri

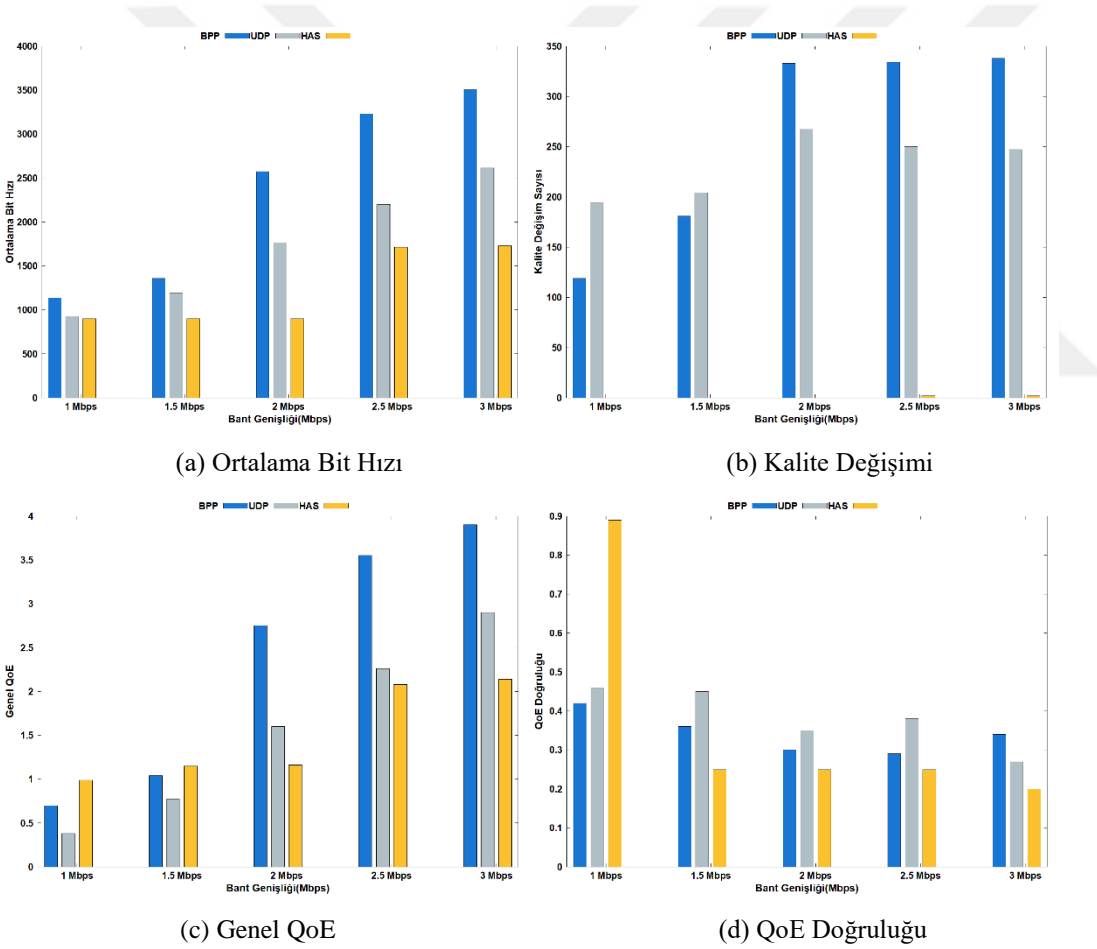
Bu bölümde, sanallaştırılmış BPP işlevi ile diğer protokoller olan UDP ve HAS karşılaştırılarak, performans değerlendirmesi yapılmaktadır. BPP ve UDP deneylerinde sunucu videoyu en yüksek bit hızında göndermektedir.

Bu kısımdaki testler için kullanılan video dosyası, daha önce kullanılan parametrelerden daha farklı bir şekilde kodlanmıştır. Temel katmanın (L0) bit hızı: 0,9 Mbps, ilk artırılmış katmanın bit hızı (L1): 1,9 Mbps ve ikinci artırılmış katmanın bit hızı (L2): 4,4 Mbps’ dir. BPP kullanılarak oluşturulan paketlerdeki fazladan eklenen veri yükü göz önüne alınarak HAS deneyleri için de benzer bit hızında video dosyaları kullanılmaktadır. Daha önce belirtildiği gibi, mevcut H264 SVC kodlayıcı/çözücüler hala deneyseldir, bu nedenle her zaman tam video çözme işlemi yapamayabilmektedir ve dolayısıyla video akışını her zaman yeniden oluşturamayabilmektedir. Sahip olunan araçlar bazı bilgileri sağlamaktadır fakat bu sınırlı olmaktadır. Bunun sonucunda, VMAF, SSIM veya ITU-T önerisi P.1203 gibi daha yeni QoE video ölçütleri değerlendirilememektedir.

Yapılan testlerde, sabit bant genişliği için 1 Mbps ile 3 Mbps arasında değişen bir sabit bant genişliği değeri kullanılmaktadır. Ayrıca sunucuya 6 adet istemci bağlanmaktadır. Aynı ağ koşulları kullanılarak HAS ve UDP için her test yeniden yapılandırılmıştır. HAS doğal olarak uyarlamalı bir yaklaşıma sahipken, UDP ile yapılan deneylerde kalite adaptasyonu yoktur. İlk ön bellekleme süresi düşük gecikme ile iletim sağlamak amacıyla tüm yaklaşımlar için 600 ms olarak ayarlanmıştır. Toplam video süresi 42 saniyedir. Dinamik senaryo 1’ de, bant genişliği değeri 3 Mbps ile başlar, ardından 2 Mbps’ ye düşer ve sonra 1 Mbps’ ye düşer. Dinamik Senaryo 2’ de, bant genişliği değeri tekrar 3 Mbps ile başlar, ancak sonra 1 Mbps’ ye düşer ve 2 Mbps’ ye yükselir. Her bant genişliği değeri değişikliği 14 saniyelik bir aralıkla gerçekleşir. Bu deneylerde gözlenen performans sonuçları

aşağıda sıralanmıştır; elde edilen parametreler ortalanmakta grafiklerde sunulmaktadır.

Bu deney setinde QoE parametreleri her protokol için toplanıp ortalama alınmıştır. Şekil 5.6a’ da, her protokol için ortalama alınan video bit hızı değerleri sunulmaktadır. UDP istemcileri kaliteyi uyarlamadığı için UDP istemcilerinin bit hızı değerleri HAS istemcilerinden daha yüksektir, ancak bu UDP istemcilerinin uzun süreli duraksamalar yaşamasına neden olmaktadır. BPP kullanan istemciler, UDP ve HAS istemcilerine göre daha yüksek kalitede video oynatmaktadır. HAS istemcileri en düşük bit hızına sahip olmalarına rağmen, HAS istemcileri ile gözlenen diğer QoE parametreleri, ABR algoritmasının duraksama süresini ve kalite değişikliklerinin sayısını minimumda tutmayı hedeflediğini göstermektedir.



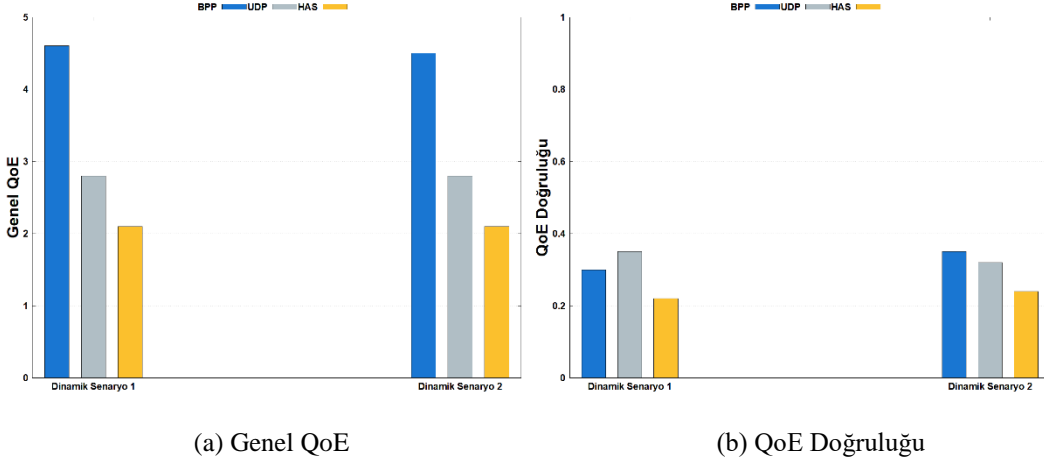
Şekil 5.6 Sanallaştırılmış BPP Yönteminin Sabit Bant Genişliği ile Kalite Parametreleri (Tuker et al, 2023)

Şekil 5.6b’ de görüldüğü gibi BPP kullanıldığında kalite değişikliklerinin sayısı yüksek çıkmaktadır. Çünkü kalite, HAS’ deki gibi 2 saniyelik bir segmentle sınırlı olmak yerine çerçeve bazında değişebilmektedir. 42 saniyelik akışlandırma

için HAS 2 saniye uzunluğunda 21 adet segment kullanılmaktadır, BPP 1000 adet çerçeve kullanılmaktadır. Bu farklılıktan dolayı her 2 saniyede bir, büyük sayıda paket iletimi gerçekleştirilmektedir.

Şekil 5.6c' de görülen Genel QoE değeri, daha önceki bölümde belirtilmiş olan QoE Hesaplama Denklemine değiştirilmiş bir sürümü kullanılarak hesaplanmaktadır. Hesaplama sırasında, duraklama süresinin doğrudan bir nedeni olan kesinti sayısına dayalı yeni bir QoE parametresi eklenmektedir. Bu, QoE' yi olumsuz etkileyen bir parametre olduğundan, formülde kesinti sayısı değeri toplam değerden çıkarılmaktadır. QoE değerinin hesaplanmasında ortalama alınan video bit hızı  $Q_k$  parametresi olarak kullanılır. Bit hızı birimi Kbps cinsinden olduğundan, diğer parametrelerden yani kalite değişikliklerinin sayısı ve duraklama süresinden daha yüksektir. Bu nedenle tüm QoE parametreleri normalize edilmiştir. Bu QoE hesaplamalarında, video bit hızının önemini gösteren  $\alpha$  parametresi 5 olarak ayarlanmıştır. Ayrıca, kalite değişikliklerinin sayısı, kesinti sayısı ve duraklama süresi için parametre ağırlıkları sırasıyla 0,01, 0,5 ve 0,5 olarak kullanılmıştır. Şekil 5.6c' de görüldüğü gibi genel QoE değerlerine bakıldığında, sadece bant genişliğinin 1 Mbps' ye eşit olduğu durumlar haricinde BPP' nin sabit bant genişliği deneylerinde diğer yaklaşımları geride bıraktığı görülmektedir. Bu çok sınırlı bant genişliği testinde HAS istemcileri her zaman kaliteyi en düşük düzeyde tutarak akışlandırma gerçekleştirmektedir. Bu yaklaşım HAS' in daha yüksek QoE değeri elde etmesine yardımcı olmaktadır.

Ayrıca, Şekil 5.6d' de görüldüğü gibi QoE Doğruluğu değeri de ölçülmektedir. En iyi QoE Doğruluğu değeri, bant genişliğinin 1 Mbps olduğu durumda HAS ile elde edilmiştir. Bu test grubunda, tüm istemciler tüm segmentler için en düşük kaliteyi talep etmekte ve çok benzer QoE değerleri elde etmektedirler. Diğer bant genişliği ayarları için, BPP ve UDP' nin HAS' ten biraz daha yüksek QoE adil değeri sağladığı görülmektedir, QoE değerleri Şekil 5.6c' de görüldüğü gibi BPP ile UDP arasında daha yüksek olmaktadır.



Şekil 5.7 Sanallaştırılmış BPP Yönteminin Dinamik Bant Genişliği ile Kalite Parametreleri (Tuker et al, 2023)

Şekil 5.7a ve Şekil 5.7b’ de, dinamik ortamda gerçekleştirilen testler için QoE ile ilgili sonuçlar sunulmaktadır. BPP için gözlenen QoE değerleri diğer yaklaşımlara göre yüksektir. Sabit bant genişliği testlerinde verilenlere benzer bir QoE Doğruluğu davranışı gözlemlenmektedir. Bu sonuçlar, BPP’ nin farklı ağ koşullarında diğer yaklaşımlardan daha yüksek QoE değerleri sağlayarak benzer şekilde davrandığını göstermektedir. Ancak halen, kalite değişikliklerinin sayısı ve kesinti sayısı gibi bazı QoE parametreleri açısından BPP’ nin performansını artırma amaçlı çalışma alanı bulunmaktadır.

## 5.5. Paket Stratejileri Karşılaştırma Testleri

Bu bölümde, BPP paketleme stratejilerinin performanslarının karşılaştırılması için testler gerçekleştirilmiştir. Farklı paketleme stratejileri arasında karşılaştırmalar yapabilmek için video akışlandırması 1 ila 4,5 Mbps arasında değişen sabit bir bant genişliği ve 1 ila 3 Mbps arasında değişen dinamik bir bant genişliği olmak üzere iki farklı bant genişliği koşulu altında gerçekleştirilmiştir. Artan senaryoda, bant genişliği 1 Mbps’ de başlamakta, sonra 2 Mbps’ ye yükselmekte ve ardından 3 Mbps’ ye çıkmaktadır. Dinamik senaryoda ise ilk olarak 3 Mbps’ de başlanmakta, sonra 1 Mbps’ ye düşmekte ve sonra 2 Mbps’ ye yükselmektedir. Azalan senaryoda, bant genişliği 3 Mbps’ de başlamakta, 2 Mbps’ ye inmekte ve daha sonra 1 Mbps’ ye düşmektedir. Her bant genişliği değeri değişikliği 14 saniyelik bir aralıkla gerçekleştirilmektedir. Test için kullanılan videonun toplam süresi 42 saniyedir. SVC kodlu videonun detayları Tablo 5.3’ te gösterilmektedir.

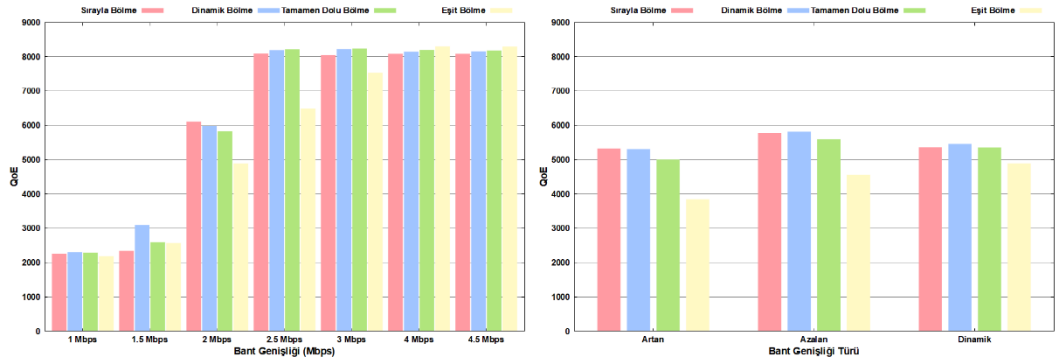
Tablo 5.3 Testlerde Kullanılan Video Dosyasının Özellikleri

<b>Video Tipi</b>	H.264	<b>Çerçeve Hızı</b>	24 fps
<b>Çözünürlük</b>	480 x 360	<b>L0 Bit Hızı</b>	897.66 Kbps
<b>Çerçeve Sayısı</b>	1000	<b>L0+L1 Bit Hızı</b>	1927.29 Kbps
<b>Video Uzunluğu</b>	42 sn	<b>L0+L1+L2 Bit Hızı</b>	4384.19 Kbps

Kullanılan paketleme stratejilerine dayalı olarak üretilen paketlerin sayısı Tablo 5.4' te gösterilmektedir.

Tablo 5.4 Üretilen Paket Sayıları

<b>Paketleme Stratejisi Türü</b>	<b>Üretilen Paket Sayısı</b>
Eşit Bölme ( <b>Pe</b> )	28.560
Dinamik Bölme ( <b>Pd</b> )	17.638
Sırayla Bölme ( <b>Pi</b> )	18.626
Tamamen Dolu Bölme ( <b>Pf</b> )	17.624



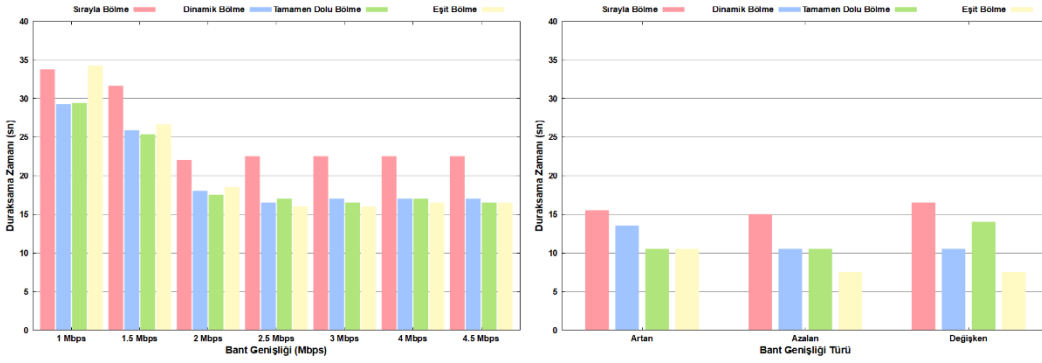
(a) Sabit Bant Genişliği Genel QoE

(b) Dinamik Bant Genişliği Genel QoE

Şekil 5.8 Paketleme Stratejilerine Göre Genel QoE Sonuçları

Test sonuçlarının yorumlanmasında paket adlarının kısaltılmaları kullanılmaktadır. Test sonuçları incelendiğinde, Pd paketleme stratejisinin, Şekil 5.8a' da gösterildiği gibi düşük bant genişliğiyle (1-1,5 Mbps) yapılan testlerde ve Şekil 5.8b' deki bant genişliğinin azaldığı testlerde diğer paketleme stratejilerine göre daha iyi Genel QoE sonuçları elde edildiği görülmektedir. Pe' ye kıyasla Pd

paketleme stratejisiyle daha az paket üretmenin avantajı ortaya çıkmaktadır. Şekil 5.9a ve Şekil 5.9b’ de gösterildiği gibi, Pd paketleme stratejisinin toplam bekleme süresi açısından düşük bir değeri vardır. Ancak Şekil 5.10a ve Şekil 5.10b’ ye dayanarak toplam bit hızı açısından tüm paketleme stratejilerine göre en iyi değere sahiptir. Pd paketleme stratejisi, 4 Mbps bant genişliğine kadar yapılan tüm testlerde Pe paketleme stratejisinden daha yüksek bir bit hızı ve QoE değerine sahip olmaktadır.

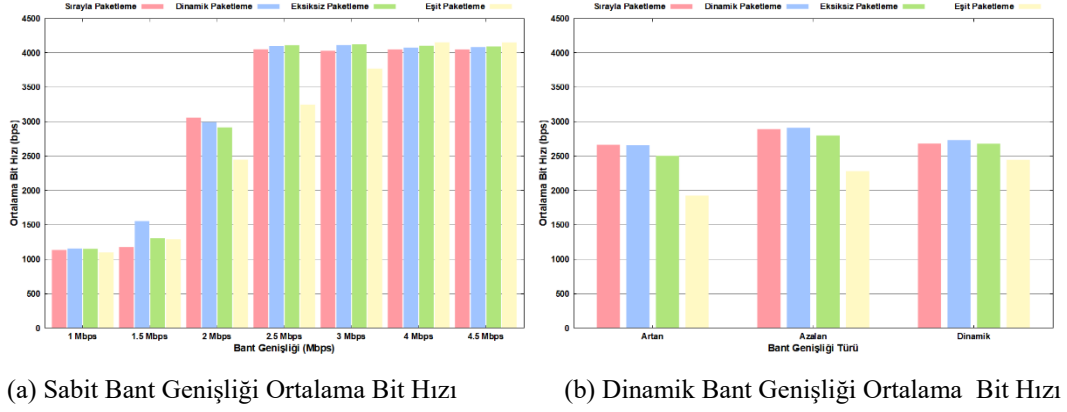


(a) Sabit Bant Genişliği Duraksama Süresi

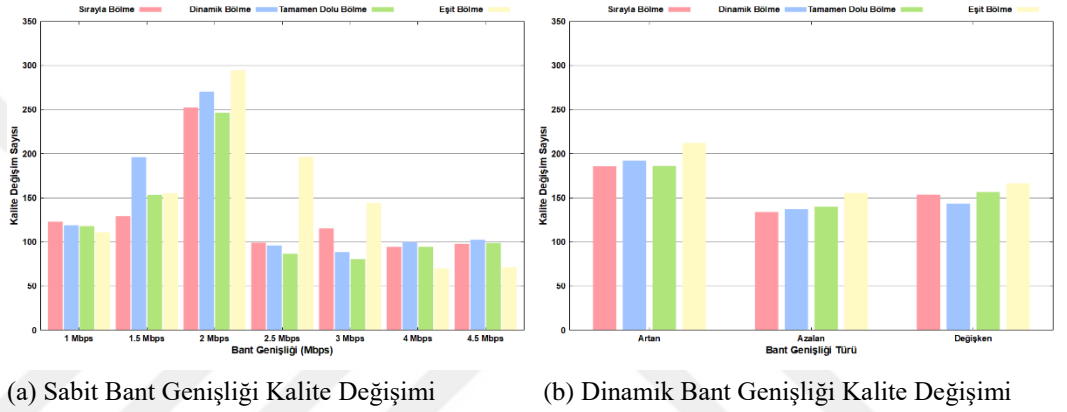
(b) Dinamik Bant Genişliği Duraksama Süresi

Şekil 5.9 Paketleme Stratejilerine Göre Toplam Duraksama Süreleri

Tablo 5.4’ ten görülebileceği gibi, Pe’ nin paket sayısındaki artış, düşük bant genişliğiyle yapılan iletimlerde bir dezavantaj haline gelen bir artışa yol açmaktadır. Örneğin, testler için kullanılan video dosyasının L1 Bitrate değeri yaklaşık 2 Mbps’ dir. Ancak 2 Mbps testinin sonuçları incelendiğinde, tüm paketleme stratejilerinin Kalite Değişikliği değerlerinin yüksek olduğu Şekil 5.11 a’ dan görülmektedir. Paket yapısı sırasında meydana gelen fazlalık nedeniyle L1 parça alımı yeterli şekilde gerçekleştirilememiştir. Oluşturulan fazlalık, videonun ilgili katmanının (L1) bit hızı değerine eklendiğinde tüm L1 çerçevelerini taşımak için yetersiz hale gelir. Bu nedenle, büyük miktarda L0 çerçevesi oluşur ve L0-L1 Çerçeveleri arasındaki geçiş sayısındaki artışla birlikte büyük miktarda kalite değişikliği gözlemlenir. Şekil 5.11 a’ ya göre, 2,5 Mbps bant genişliği kullanıldığında tüm paketleme stratejileri için kalite değişikliği değeri azalmaktadır. En az azalma gösteren strateji, büyük paket sayısı nedeniyle Pe paket stratejisidir. Çünkü, büyük paket sayısı ekstra fazlalığa neden olmaktadır. Özellikle 4 ve 4,5 Mbps bant genişliği kullanılarak yapılan testlerde, tüm paketleme türleri için kalite değişikliklerinin sayısı minimize edilmiştir.



Şekil 5.10 Paketleme Stratejilerine Göre Ortalama Bit Hızı



Şekil 5.11 Paketleme Stratejilerine Göre Toplam Kalite Değişimi

Pe paketleme stratejisinin yarattığı büyük paket sayısına rağmen, birçok test için (2 Mbps ve üstü testler) toplam gecikme süresi açısından en düşük değere sahiptir. Bant genişliği 4 Mbps ve üzerine çıkarıldığında, Pe paketleme stratejisi Şekil 5.8a' da görüldüğü gibi QoE açısından en yüksek değerlere ulaşmaktadır. Bu paketleme stratejisi yüksek bant genişliği değerlerinde daha iyi sonuçlar vermektedir.

Şekil 5.9a ve Şekil 5.9b' de görüldüğü gibi, Pi paketleme stratejisi, tüm bant genişliği değerleriyle yapılan testlerde en yüksek toplam bekleme süresine sahip olup, bu durum QoE açısından olumsuz sonuçlara yol açmaktadır. Pi paketleme stratejisi, 2 Mbps ile yapılan iletimde en yüksek bit hızı değerine sahip olduğundan, en iyi QoE sonucuna ulaşmakta ve bu bant genişliği değerleri için en uygun paketleme stratejisi haline gelmektedir. Tüm testler açısından, Pi paketleme stratejisi yüksek bant genişliği kullanılarak yapılan testlerde en kötü sonuçlara sahiptir.

Düşük toplam duraksama süresi değeri ve yüksek bit hızı değerine sahip olan Pf paketleme stratejisi, tüm test türlerinde ortalama bir konumdadır. Hiçbir test türünde minimum QoE değerine sahip olmamıştır. Diğer paketleme türlerini büyük bir farkla geçmemekte, ancak uygun durumlarda iyi sonuçlar alabilmektedir. Pf paketleme stratejisi, Şekil 5.11a' da görüldüğü gibi en düşük Kalite Değişikliği değerine ve Şekil 5.10a' da görüldüğü gibi 2,5 ve 3 Mbps testlerinde en yüksek bit hızı değeriyle Şekil 5.8a' da görüldüğü gibi en iyi QoE sonucuna sahip olmaktadır.

Artan bant genişliği testinde, Şekil 5.11b' de belirtildiği gibi, düşük Toplam Kalite Değişiklik sayısı nedeniyle Pi paketleme stratejisi, Toplam Duraksama Süresi açısından en yüksek değere sahiptir. Bununla birlikte, sadece küçük bir farkla Genel QoE' de en iyi sonucu elde etmektedir.

## 6. SONUÇLAR

Gerçekleştirilen tez çalışmasında; YTA ortamında sanallaştırılmış ağ işlevlerini kullanarak video akışı uygulamalarının performansını artırmak hedeflenmiştir. Bu amaçla video akışı uygulamaları için düşük gecikme ile yüksek QoE sunmak üzere yazılım tanımlı ağ ortamında paket küçültme ve BPP protokolünü kullanan bir mimari önerilmektedir. Bu tezde önerilen çalışmalar üç sınıfa ayrılabilir:

İlk olarak video akışlandırmaya uygun yapıda BPP Paket yapılarının oluşturulması için gerçekleştirilen çalışmalar bulunmaktadır. BPP paket yapısının ilk önerildiği yayınlarda çok sınırlı bilgi bulunması ve gerçekleştirilmiş bir uygulama bulunmaması dolayısıyla tez çalışması içerisinde kullanılan paket yapısı ilk kez oluşturulmuştur ve geliştirilmeye devam edilmektedir. Paket yapısı içerisinde bulunan alanlar video akışlandırma için gereken özellikleri taşımaktadırlar; fakat paket yapısında veya paketleme stratejilerinde ek geliştirmeler yapılarak daha iyi sonuçlar elde edilebileceği düşünülmektedir. Ayrıca tez içerisinde gözlemlenen paketleme stratejilerinin karşılaştırma sonuçları da ilerideki geliştirmeler için bir temel oluşturmaktadır.

İkinci olarak katmanlı video aktarımı için Büyük Paket Protokolü (BPP) kullanarak, bir YTA denetleyicisinin yönetimindeki bir ağda, video iletim sistemi oluşturulması için gerçekleştirilen çalışmalar bulunmaktadır. Bir ONOS denetleyicisinin, ağ yönetimi sorumluluklarının yanı sıra, ağ içi kalite adaptasyonunu uygulamak için bir çözüm olarak nasıl kullanıldığı gösterilmiştir. Bu amaçla, katmanlı video iletimini desteklemek için BPP geliştirmesi önerilmiş ve detaylı olarak sunulmuştur. Gerçek zamanlı ağ durumu bilgisi ile video kodlama özelliklerini kullanan bir YTA denetleyicisi tasarlanmıştır. Bu denetleyici, video kalitesini ve BPP veri parçalarını dikkate alarak video iletimini yönetmiştir. İletim sırasında YTA denetleyicisi, video karakteristikleri ve mevcut ağ koşullarına dair bilgisini kullanarak paketleri değiştirebilmekte ve bazı veri parçalarını çıkarabilmektedir. Karşılaştırma testlerinde kullanılmış olan HAS istemcisi ağı ve bireysel bant genişliği tahminlerini kullanarak video kalitesini ayarlarken, geliştirilen yaklaşım kaliteyi doğrudan ağ içinde adapte etmektedir. Tez çalışmasında oluşturulan yapı, YTA ve BPP özelliklerini birleştirerek elde edilen bir ağ içi kalite adaptasyonu sunmaktadır. Gerçek bant genişliği değerlerini kullanarak ağ içinde kalite adaptasyonu yapmak, geçmiş verilere dayalı tahminleri

kullanan istemci tarafındaki kalite tahminlerinden daha iyi performans sağlamaktadır.

Üçüncü olarak BPP işlemlerinin YTA Denetleyici üzerinden gerçekleştirilmesinden kaynaklanan ve tez içerisinde gösterilmiş olan ölçeklenebilirlik problemlerine karşın kenar ağda oluşturulan uygulamalar yer almaktadır. Kenar ağda sanallaştırılmış BPP işlevlerini video iletimi için kullanan bir mimari sunulmuştur. Deneysel sonuçlar, kenarda ağ içi kalite adaptasyonunun uygulanmasının ve sanallaştırılmış BPP işlevinin kullanılmasının ölçeklenebilirlik ve QoE' de bir iyileşme sağladığını göstermiştir. Kenarda paketleri küçültme yöntemi kullanılarak elde edilen QoE' nin ONOS denetleyicisi kullanmaktan daha iyi olduğunu gösteren sonuçlar elde edilmiş ve gösterilmiştir. Sanallaştırılmış BPP işlevleri kullanılarak gerçekleştirilen testlerde elde edilen ortalama video bit hızı değerinin, ONOS denetleyicisi kullanılarak gerçekleştirilen testlere göre daha yüksek olduğu görülmektedir. Tez içerisinde yapılan testler ve sonuçlardan görüldüğü kadarıyla BPP' nin katmanlı video iletiminde kullanılmasının, sınırlı bant genişlikleriyle bile istemciye kesintisiz oynatılabilir bir video akışı ve düşük kesinti düzeyleri sunduğu görülmektedir.

Video akış uygulamalarında, çerçeveler, oynatma zamanından daha sonra istemciye ulaşırlarsa QoE kötü yönde etkilenmektedir. BPP ile yapılan tüm testlerde, taban katmana ait tüm çerçeveler zamanında ya da düşük gecikme süresine ulaşmış ve sadece bazı geliştirme katmanı veri parçaları silinmiştir. UDP kullanıldığında, bazı çerçeveler tamamen kaybolmuş ve diğerleri gecikmeli iletilmiştir. TCP kullanıldığında, paket kaybı olmamış ancak kabul edilemeyecek seviyede yüksek gecikme süreleri ortaya çıkmıştır. Sonuçlar, BPP' nin kaliteyi etkili bir şekilde adapte ettiğini, böylece HAS ile karşılaştırıldığında daha düşük toplam duraksama süresi ve daha yüksek ortalama alınan bit hızı elde edilebileceğini göstermektedir. BPP' nin, video için yüksek hassasiyetli zamanlama açısından faydalar sağladığı seçilen tekniklerin video transferi için iyi çalıştığı açıkça görülmektedir. Ayrıca sonuçlarımız, BPP' nin ve katmanlı video akışının birbirini tamamladığını ve bu mekanizmanın online oyun, Artırılmış Gerçeklik ve Sanal Gerçeklik gibi düşük gecikme, yüksek bant genişliği gerektiren diğer alanlarda da uygun olabileceğini göstermektedir.

Tez çalışmasından elde edilen içgörüler, ağ içi video kalite adaptasyonunun gelişmiş QoE sağlayabileceğini göstermektedir. Ancak HAS ile yapılan karşılaştırma testlerine bakıldığında, BPP' ye kıyasla kalite değişikliklerinin

sayısının daha düşük olduğu görülmektedir. Genel olarak BPP için, kalite değişikliklerinin sayısını dikkate alan yaklaşımların geliştirilmesi durumunda QoE değerinin daha da yüksek olabileceği görülmektedir. Gelecekte yapılacak olan çalışmalarda bir dizi iyileştirme gerçekleştirilmesi planlanmaktadır. Paketlerden veri parçası çıkaran algoritmanın farklı parametrelere göre de adapte edilmesi düşünülmektedir. Şu anda yalnızca mevcut bant genişliği ve kullanılan bant genişliğine dikkat edilerek ve kalite değişikliklerinin sayısı hesaba katılmadan işlemler gerçekleştirilmektedir. Bu sayı testlerde oldukça yüksektir, ancak daha düzgün bir iletim için kalite değişikliklerinin sayısını azaltmak gerekmektedir. Paketlerden veri parçası çıkaran algoritmanın zaman karmaşıklığı değeri paket içerisindeki parça sayısı, bu parçaların değerlendirilme adımlarına ve istemci sayısına göre değişmektedir. Algoritmanın zaman karmaşıklığı değerinin düşürülmesi konusunda çalışmalar yapılması planlanmaktadır. Ayrıca, gereken altyapının oluşması durumunda daha yüksek verimlilik sağlamak için uygun donanım sistemlerini kullanarak paket küçültme işleminin gerçekleştirilmesinin performansını ölçmek planlanmaktadır. Tez çalışması içerisinde gerçekleştirilen test sonuçlarına göre BPP Paketi oluşturmak için kullanılan stratejinin de sonuçlara etkisi olabilmektedir. Bunun için ağıın anlık durumuna göre en uygun paketleme stratejisinin seçilmesini sağlayan bir yapı oluşturulması düşünülmektedir. Bu amaçla yürütülmekte olan TÜBİTAK Proje çalışması içerisinde Derin Pekiştirme Algoritması geliştirilerek bu konuda bir çözüm üretilmesi planlanmaktadır.

## KAYNAKLAR DİZİNİ

- Adrian, P., Dragos, D., Mark, H., Georgios, N., Jeongkeun, L., and Costin, R.,** 2022, Implementing packet trimming support in hardware (arXiv:2207.04967). arXiv. <http://arxiv.org/abs/2207.04967>.
- Ahmad, A., Floris, A., and Atzori, L.,** 2020, Timber: An SDN-Based Emulation Platform for Experimental Research on Video Streaming. *IEEE Journal on Selected Areas in Communications*, 38(7), pp. 1374-1387. <https://doi.org/10.1109/JSAC.2020.2999683>.
- Albalawi, A., Yousefi, H., Westphal, C., Makhijani, K., and Garcia-Luna-Aceves, J. J.,** 2020, Enhancing End-to-End Transport with Packet Trimming. *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 1-7. <https://doi.org/10.1109/GLOBECOM42002.2020.9322506>.
- Al-Issa, A. E., Bentaleb, A., Barakabitze, A. A., Zinner, T., and Ghita, B.,** 2019, Bandwidth Prediction Schemes for Defining Bitrate Levels in SDN-enabled Adaptive Streaming. *2019 15th International Conference on Network and Service Management (CNSM)*, pp. 1-7. <https://doi.org/10.23919/CNSM46954.2019.9012713>.
- Alsmadi, I., and Xu, D.,** 2015, Security of Software Defined Networks: A survey. *Computers & Security*, 53, 79-108. <https://doi.org/10.1016/j.cose.2015.05.006>.
- Arasan, H.,** 2022, Mec Kaynak Yönetimi İçin Bir YTA Kontrolör Modülünün Geliştirilmesi Ve İmplementasyonu, Yüksek Lisans Tezi, Ege Üniversitesi Uluslararası Bilgisayar Enstitüsü, 32s
- Belshe, M., Peon, R., and Thomson, M.,** 2015, Mayıs, RFC7540. IETF HTTP Working Group Specifications. <https://httpwg.org/specs/rfc7540.html>.
- Ben Azzouz, L., and Jamai, I.,** 2019, SDN, slicing, and NFV paradigms for a smart home: A comprehensive survey. *Transactions on Emerging Telecommunications Technologies*, 30(10). <https://doi.org/10.1002/ett.3744>.

## KAYNAKLAR DİZİNİ (devam)

- Ben Letaifa, A.**, 2017, Adaptive QoE monitoring architecture in SDN networks: Video streaming services case. 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1383-1388. <https://doi.org/10.1109/IWCMC.2017.7986486>.
- Bentaleb, A., Taani, B., Begen, A. C., Timmerer, C., and Zimmermann, R.**, 2019a, A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP. IEEE Communications Surveys & Tutorials, 21(1), pp. 562-585. <https://doi.org/10.1109/COMST.2018.2862938>.
- Binsahaq, A., Sheltami, T. R., and Salah, K.**, 2019, A Survey on Autonomic Provisioning and Management of QoS in SDN Networks. IEEE Access, 7, pp. 73384-73435. <https://doi.org/10.1109/ACCESS.2019.2919957>.
- Bishop, M.**, 2022, HTTP/3 (Request for Comments RFC 9114). Internet Engineering Task Force. <https://doi.org/10.17487/RFC9114>.
- Bitmovin., 2023**, "Low Latency Streaming: What is it and How can it be solved?" <https://bitmovin.com/cmaf-low-latency-streaming/> (Erişim Tarihi 16 Ağustos 2023).
- Cao, K., Liu, Y., Meng, G., and Sun, Q.**, 2020, An Overview on Edge Computing Research. IEEE Access, 8, pp. 85714-85728. <https://doi.org/10.1109/ACCESS.2020.2991734>.
- Clark, D. D., and Tennenhouse, D. L.**, 1990. "Architectural Considerations for a New Generation of Protocols", In SIGCOMM '90.
- Clayman, S.**, 2019, The Inter-Dependence of Network Transport and Application Behaviour. ITU Network, 2030.
- Clayman, S., and Sayit, M.**, 2023, Low Latency Low Loss Media Delivery Utilizing In-Network Packet Wash. Journal of Network and Systems Management, 31(1), 29. <https://doi.org/10.1007/> pp. 10922-022-09712-1.
- Clayman, S., and Sayit, M.**, 2021, In-Network Scalable Video Adaption Using Big Packet Protocol. Proceedings of the 12th ACM Multimedia Systems Conference, pp. 363-368. <https://doi.org/10.1145/3458305.3478440>.

## KAYNAKLAR DİZİNİ (devam)

- Clayman, S., and Sayit, M.,** 2022, The Effects of Packet Wash on SVC Video in Limited Bandwidth Environments. 2022 IEEE 23rd International Conference on High Performance Switching and Routing (HPSR), pp. 17-22. <https://doi.org/10.1109/HPSR54439.2022.9831279>.
- Clayman S, Toker M, Arasan H, Sayit M.,** 2021a Managing Video Processing and Delivery using Big Packet Protocol with SDN Controllers. In: 2021 IEEE 7th International Conference on Network Softwarization (NetSoft) [Internet]. Tokyo, Japan: IEEE; 2021 [cited 2023 Aug 9]. p. 196–200. Available from: <https://ieeexplore.ieee.org/document/9492574/>.
- Clayman, S., Toker, M., Arasan, H., and Sayit, M.,** 2021b, The Future of Media Streaming Systems: Transferring Video over New IP. 2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR), Paris, France 1-6. <https://doi.org/10.1109/HPSR52026.2021.9481835>.
- Clayman S, Toker M, Karakış E, Sayit M.,** 2023. In-Network Video Quality Adaption using Packet Trimming at the Edge. In: 2023 26th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN) [Internet]. Paris, France: IEEE; [cited 2023 Aug 9]. p. 161–8. Available from: <https://ieeexplore.ieee.org/document/10073496/>.
- Cotroneo, D., De Simone, L., Iannillo, A. K., Lanzaro, A., Natella, R., Fan, J., and Ping, W.,** 2014, Network Function Virtualization: Challenges and Directions for Reliability Assurance. 2014 IEEE International Symposium on Software Reliability Engineering Workshops, pp. 37-42. <https://doi.org/10.1109/ISSREW.2014.48>.
- Crowcroft, J.,** 1999, Internetworking Multimedia. CRC Press.
- Dong, L., and Li, R.,** 2019a, “Big Packet Protocol: Advances the Internet with InNetwork Services and Functions,” MMTc Communications - Frontiers, 14(5), pp. 23–29.

**KAYNAKLAR DİZİNİ (devam)**

- Dong, L., and Li, R.,** 2019b, Distributed Mechanism for Computation Offloading Task Routing in Mobile Edge Cloud Network. 2019 International Conference on Computing, Networking and Communications (ICNC), pp. 630-636. <https://doi.org/10.1109/ICCNC.2019.8685537>.
- Dong, L. and Li, R.,** 2019c. In-packet network coding for effective packet wash and packet enrichment, 1–6 (2019).
- Erfanian, A., Amirpour, H., Tashtarian, F., Timmerer, C., and Hellwagner, H.,** 2021a, LwTE-Live: Light-weight Transcoding at the Edge for Live Streaming. Proceedings of the Workshop on Design, Deployment, and Evaluation of Network-Assisted Video Streaming, pp. 22-28. <https://doi.org/10.1145/3488662.3493829>.
- Farahani, R., Tashtarian, F., Amirpour, H., Timmerer, C., Ghanbari, M., and Hellwagner, H.,** 2021a, CSDN: CDN-Aware QoE Optimization in SDN-Assisted HTTP Adaptive Video Streaming. 2021 IEEE 46th Conference on Local Computer Networks (LCN), pp. 525-532. <https://doi.org/10.1109/LCN52139.2021.9524970>.
- Farahani, R., Tashtarian, F., Erfanian, A., Timmerer, C., Ghanbari, M., and Hellwagner, H.,** 2021b, ES-HAS: An edge- and SDN-assisted framework for HTTP adaptive video streaming. Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, pp. 50-57. <https://doi.org/10.1145/3458306.3460997>.
- Farahani, R., Tashtarian, F., Timmerer, C., Ghanbar, M., and Hellwagner, H.,** 2022, LEADER: A Collaborative Edge- and SDN-Assisted Framework for HTTP Adaptive Video Streaming. ICC 2022 - IEEE International Conference on Communications, pp. 745-750. <https://doi.org/10.1109/ICC45855.2022.9838949>.

**KAYNAKLAR DİZİNİ (devam)**

- Georgopoulos, P., Broadbent, M., Plattner, B., and Race, N.,** 2014, Cache as a service: Leveraging SDN to efficiently and transparently support video-on-demand on the last mile. 2014 23rd International Conference on Computer Communication and Networks (ICCCN), pp. 1-9. <https://doi.org/10.1109/ICCCN.2014.6911775>.
- Grafl, M., Timmerer, C., Hellwagner, H., Cherif, W., and Ksentini, A.,** 2013, Evaluation of hybrid Scalable Video Coding for HTTP-based adaptive media streaming with high-definition content. 2013 IEEE 14th International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM), pp. 1-7. <https://doi.org/10.1109/WoWMoM.2013.6583506>
- GRPC.,** 2023, (GRPC) general-purpose RPC, <https://grpc.io/> (Erişim Tarihi 28 Ağustos 2023).
- Handley, M., Raiciu, C., Agache, A., Voinescu, A., Moore, A. W., Antichi, G., and Wójcik, M.,** 20 Handley17, Re-architecting datacenter networks and stacks for low latency and high performance. Proceedings of the Conference of the ACM Special Interest Group on Data Communication, pp. 29-42. <https://doi.org/10.1145/3098822.3098825>.
- Henni, D. E., Ghomari, A., and Hadjadj-Aoul, Y.,** 2020, A consistent QoS routing strategy for video streaming services in SDN networks. International Journal of Communication Systems, 33(10), e4177. <https://doi.org/10.1002/dac.4177>.
- Islam S, Muslim N, Atwood JW.,** 2018. A Survey on Multicasting in Software-Defined Networking. IEEE Commun Surv Tutorials. 20(1) pp. 355–87.
- Joshi, K., and Benson, T.,** 2016, Network Function Virtualization. IEEE Internet Computing, 20(6), pp. 7-9. <https://doi.org/10.1109/MIC.2016.112>.

**KAYNAKLAR DİZİNİ (devam)**

- Karakus, M., and Durresi, A.,** 2017, Quality of Service (QoS) in Software Defined Networking (SDN): A survey. *Journal of Network and Computer Applications*, 80, pp. 200-218. <https://doi.org/10.1016/j.jnca.2016.12.019>.
- Karimzadeh-Farshbafan, M., Shah-Mansouri, V., and Niyato, D.,** 2020, A Dynamic Reliability-Aware Service Placement for Network Function Virtualization (NFV). *IEEE Journal on Selected Areas in Communications*, 38(2), pp. 318-333. <https://doi.org/10.1109/JSAC.2019.2959196>.
- Kleinrouweler, J. W., Cabrero, S., and Cesar, P.,** 2016, Delivering stable high-quality video: An SDN architecture with DASH assisting network elements. *Proceedings of the 7th International Conference on Multimedia Systems*, 1-10. <https://doi.org/10.1145/2910017.2910599>.
- Lantz, B., and O'Connor, B.,** 2015, A Mininet-based Virtual Testbed for Distributed SDN Development. *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pp. 365-366. <https://doi.org/10.1145/2785956.2790030>.
- Lederer, S., Müller, C. and Timmerer, C.,** 2012. Dynamic adaptive streaming over http dataset, *MMSys '12*, pp. 89–94 (Association for Computing Machinery, New York, USA, 2012).
- Li, R., Clemm, A., Chunduri, U., Dong, L., and Makhijani, K.,** 2018, A New Framework and Protocol for Future Networking Applications. *Proceedings of the 2018 Workshop on Networking for Emerging Applications and Technologies*, pp. 21-26. <https://doi.org/10.1145/3229574.3229576>.
- Li, R., Makhijani, K., Yousefi, H., Westphal, C., Dong, L., Wauters, T., and De Turck, F.,** 2019, A Framework for Qualitative Communications Using Big Packet Protocol. *Proceedings of the ACM SIGCOMM 2019 Workshop on Networking for Emerging Applications and Technologies*, pp. 22-28. <https://doi.org/10.1145/3341558.3342201>.

## KAYNAKLAR DİZİNİ (devam)

- Li, Y., and Chen, M.,** 2015. Software-Defined Network Function Virtualization: A Survey. IEEE Access. (3) pp.42–53.
- Lim, M., Akcay, M. N., Bentaleb, A., Begen, A. C., and Zimmermann, R.,** 2020, When they go high, we go low: Low-latency live streaming in dash.js with LoL. Proceedings of the 11th ACM Multimedia Systems Conference, pp. 321-326. <https://doi.org/10.1145/3339825.3397043>.
- Makhijani, K., Li, R., and Boukary, H. E.,** 2019, Using Big Packet Protocol Framework to Support Low Latency based Large Scale Networks.
- Michel, O., and Keller, E.,** 2017, SDN in Wide-Area Networks: A Survey. 2017 Fourth International Conference on Software Defined Systems (SDS), Valencia, Spain, pp. 37-42, doi: 10.1109/SDS.2017.7939138.
- MPEG., 2023., MPEG-DASH,** <https://mpeg.chiariglione.org/standards/mpeg-dash>, (Erişim Tarihi:16 Ağustos 2023).
- Mu, M., Broadbent, M., Farshad, A., Hart, N., Hutchison, D., Ni, Q., and Race, N.,** 2016, A Scalable User Fairness Model for Adaptive Video Streaming Over SDN-Assisted Future Networks. IEEE Journal on Selected Areas in Communications, 34(8), pp. 2168-2184. <https://doi.org/10.1109/JSAC.2016.2577318>.
- Nam, H., Kim, K.-H., Kim, J. Y., and Schulzrinne, H.,** 2014, Towards QoE-aware video streaming using SDN. 2014 IEEE Global Communications Conference, pp. 1317-1322. <https://doi.org/10.1109/GLOCOM.2014.7036990>.
- Nguyen, M., Amirpour, H., Timmerer, C., and Hellwagner, H.,** 2020, Scalable High Efficiency Video Coding based HTTP Adaptive Streaming over QUIC. Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC, pp. 28-34. <https://doi.org/10.1145/3405796.3405829>.

**KAYNAKLAR DİZİNİ (devam)**

- ONOS (Open Network Operating System)**, 2023, (ONOS) SDN Controller for SDN/NFV Solutions, <https://opennetworking.org/onos> (Erişim Tarihi: 24.08.2023).
- Ozcan, S. G., Kivilcim, T., Cetinkaya, C., and Sayit, M.**, 2017, Rate adaptation algorithm with backward quality increasing property for SVC-DASH. 2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), pp. 24-28. <https://doi.org/10.1109/ICCE-Berlin.2017.8210580>.
- Paliwal, M., Shrimankar, D., and Tembhurne, O.**, 2018, Controllers in SDN: A Review Report. IEEE Access, 6, 36256-36270. <https://doi.org/10.1109/ACCESS.2018.2846236>.
- Palmer, M., Krüger, T., Chandrasekaran, B., and Feldmann, A.**, 2018, The QUIC Fix for Optimal Video Streaming. Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC, pp. 43-49. <https://doi.org/10.1145/3284850.3284857>.
- Park, G. S., and Song, H.**, 2019, Cooperative Base Station Caching and X2 Link Traffic Offloading System for Video Streaming Over SDN-Enabled 5G Networks. IEEE Transactions on Mobile Computing, 18(9), pp. 2005-2019. <https://doi.org/10.1109/TMC.2018.2869756>.
- Prabha, C., Goel, A., and Singh, J.**, 2022, A Survey on SDN Controller Evolution: A Brief Review. 2022 7th International Conference on Communication and Electronics Systems (ICCES), pp. 569-575. <https://doi.org/10.1109/ICCES54183.2022.9835810>.
- Rao, A., Lanphier, R., and Schulzrinne, H.**, 1998, Real Time Streaming Protocol (RTSP) (Request for Comments RFC 2326). Internet Engineering Task Force. <https://doi.org/10.17487/RFC2326>.

## KAYNAKLAR DİZİNİ (devam)

- Sanvito, D., Moro, D., Gulli, M., Filippini, I., Capone, A., and Campanella, A.** 2018, ONOS Intent Monitor and Reroute service: Enabling plug & play routing logic. 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal pp. 272-276. <https://doi.org/10.1109/NETSOFT.2018.8460064>.
- Schulzrinne, H., Casner, S. L., Frederick, R., and Jacobson, V.,** 2003, RTP: A Transport Protocol for Real-Time Applications (Request for Comments RFC 3550). Internet Engineering Task Force. <https://doi.org/10.17487/RFC3550>.
- Schwarz, H., Marpe, D., and Wiegand, T.,** 2007, Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. IEEE Transactions on Circuits and Systems for Video Technology, 17(9), pp. 1103-1120. <https://doi.org/10.1109/TCSVT.2007.905532>.
- SDN (Software Defined Networking),** 2023, (SDN) Software-Defined Networking (SDN) Definition, <https://opennetworking.org/sdn-definition/> (Erişim Tarihi: 24.08.2023).
- Seufert, M., Schatz, R., Wehner, N., and Casas, P.,** 2019, QUICker or not? -An Empirical Analysis of QUIC vs TCP for Video Streaming QoE Provisioning. 2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), pp. 7-12. <https://doi.org/10.1109/ICIN.2019.8685913>.
- Seufert, M., Egger, S., Slanina, M., Zinner, T., Hoßfeld, T., and Trangia, P.,** 2015, “A Survey on Quality of Experience of HTTP Adaptive Streaming.” IEEE Communication Surveys Tutorials, 17(1), pp. 469–492.
- Shen, S.-H.,** 2019,. Efficient SVC Multicast Streaming for Video Conferencing With SDN Control. IEEE Transactions on Network and Service Management, 16(2), pp. 403-416. <https://doi.org/10.1109/TNSM.2019.2911086>.

## KAYNAKLAR DİZİNİ (devam)

- Sullivan, G.**, 2020, Versatile Video Coding (VVC) Arrives. 2020 IEEE International Conference on Visual Communications and Image Processing (VCIP), 1-1. <https://doi.org/10.1109/VCIP49819.2020.9301847>.
- Taraghi, B., Hellwagner, H., and Timmerer, C.**, 2023, LLL-CAdViSE: Live Low-Latency Cloud-Based Adaptive Video Streaming Evaluation Framework. IEEE Access, 11, pp. 25723-25734. <https://doi.org/10.1109/ACCESS.2023.3257099>.
- Yu, T., Wang, K., and Hsu, Y.**, 2015, Adaptive routing for video streaming with QoS support over SDN networks. 2015 International Conference on Information Networking (ICOIN), pp. 318-323. <https://doi.org/10.1109/ICOIN.2015.7057904>.
- Tuker, M., Karakış, E., Sayıt, M., and Clayman, S.**, 2023. “Using Packet Trimming at the Edge for In-Network Video Quality Adaption”. Annals of Telecommunications, <https://doi.org/10.1007/s12243-023-00981-8>.
- WebRTC., (2023)**, ‘Celebrating 10 years of WebM and WebRTC Chromium Blog’, <https://blog.chromium.org/2020/05/celebrating-10-years-of-webm-and-webrtc.html>, (Erişim tarihi 16 Ağustos 2023).
- Westphal, C., Makhijani, K. and Li, R.**, 2019. Packet Trimming to Reduce Buffer Sizes and Improve Round-Trip Times - Extended Abstract (ACM, NewYork, NY).
- Yang, S., Li, F., Trajanovski, S., Yahyapour, R., and Fu, X.**, 2021, Recent Advances of Resource Allocation in Network Function Virtualization. IEEE Transactions on Parallel and Distributed Systems, 32(2), pp. 295-314. <https://doi.org/10.1109/TPDS.2020.3017001>.
- Yang, S., Wei, Z., and Ling, L.**, 2011, Multimedia Communication and Scalable Video Coding. 2011 Fourth International Conference on Intelligent Computation Technology and Automation, pp. 616-619. <https://doi.org/10.1109/ICICTA.2011.439>.

**KAYNAKLAR DİZİNİ (devam)**

- Yang, W., and Fung, C.,** 2016, A survey on security in network functions virtualization. 2016 IEEE NetSoft Conference and Workshops (NetSoft), 15-19. <https://doi.org/10.1109/NETSOFT.2016.7502434>.
- Yi, B., Wang, X., Li, K., Das, S. K., and Huang, M.,** 2018, A comprehensive survey of Network Function Virtualization. *Computer Networks*, 133, pp. 212-262. <https://doi.org/10.1016/j.comnet.2018.01.021>.
- Zacarias, I., Schwarzrock, J., Gaspary, L. P., Kohl, A., Fernandes, R. Q. A., Stocchero, J. M., and De Freitas, E. P.,** 2017, Employing SDN to control video streaming applications in military mobile networks. 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), 1-4. <https://doi.org/10.1109/NCA.2017.8171390>.





## ÖZGEÇMİŞ

**Mustafa TÜKER**

**Yabancı Dil:**

İngilizce: İleri Seviye

**Öğrenim Durumu:**

Doktora, 2016-2023, Ege Üniversitesi, Uluslararası Bilgisayar Enstitüsü, İzmir, Türkiye

Yüksek Lisans, 2011-2013, Muğla Sıtkı Koçman Üniversitesi, Elektronik-Bilgisayar Eğitimi Bölümü, Muğla, Türkiye

Lisans, 2014-2015, Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü (Mühendislik Tamamlama), İzmir, Türkiye

Lisans, 2001-2005, Süleyman Demirel Üniversitesi, Bilgisayar Sistemleri Öğretmenliği, Isparta, Türkiye

Ön Lisans, 1998-2000, Anadolu Üniversitesi, Bilgisayar Programcılığı, Eskişehir, Türkiye.

**Yüksek Lisans Tez Başlığı ve Tez Danışmanı:** Heterojen Bilgisayar Ağları İçin Zeki Seçim Sistemleri Geliştirilmesi - 2013 (Doç. Dr. Serkan BALLI)

**Doktora Tezi Başlığı ve Danışmanı:** Sanallaştırma ve YTA Teknolojilerine Dayalı Video İletimi (Doç. Dr. Müge SAYIT)

**TÜBİTAK Projeleri**

• 2021- : Geleceğin Ağlarında Uçtan Uca Video Akışlandırma ve Çoklu Gönderim (TÜBİTAK: 121E373)

**Yayınlar:**

**Tuker, M., Karakış, E., Sayıt, M., and Clayman, S., 2023.** “Using Packet Trimming at the Edge for In-Network Video Quality Adaption”. Annals of Telecommunications, <https://doi.org/10.1007/s12243-023-00981-8>.

- S. Clayman, M. Tüker, E. Karakiş, M. Sayıt**, 2023. “In-Network Video Quality Adaption using Packet Trimming at the Edge” 2023 26th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN). (ICIN 2023)
- E. Karakiş, S. Clayman, M. Tüker, E. Bozkaya, M. Sayıt M**, 2023. “Towards High Precision End-to-End Video Streaming from Drones using Packet Trimming”. NetSoft 2023 - 5th International Workshop on Performance Evaluation of Next Generation Virtualized Environments and Software-Defined Networks (PVE-SDN). (PVE-SDN 2023)
- E. Karakiş, S. Clayman, M. Tüker, E. Bozkaya, M. Sayıt**, 2023. “Energy Aware Video Streaming From Drones With End-To-End Delivery Using Packet Trimming.” 2023 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom) (BlackSeaCom 2023)
- S. Clayman, M. Tüker, H. Arasan , M. Sayıt**, 2021. “Managing Video Processing and Delivery using Big Packet Protocol with SDN Controllers.” 2021 IEEE 7th International Conference on Network Softwarization (NetSoft). (IEEE NetSoft 2021)
- S. Clayman, M. Tüker, H. Arasan , M. Sayıt**, 2021. “The Future of Streaming Systems : Transferring Video over New IP.” 2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR). (IEEE HPSR'21) (Invited Paper)
- S. Ballı, M. Tüker**, 2018. “A Fuzzy Multi-Criteria Decision Analysis Approach for the Evaluation of the Network Service Providers in Turkey. Intelligent Automation & Soft Computing.” doi: 10.1080/10798587.2017.1306968
- S. Ballı, M. Tüker**, 2014. “Heterojen Bilgisayar Ağlarının Performanslarının Değerlendirilmesi için Parametre Elde Edilmesi.” inet-tr'14 - XIX. Türkiye’de İnternet Konferansı - Yaşar Üniversitesi, İzmir
- M. Tüker, S. Ballı, İ. Pembeci**, 2013. “Çok Etmenli Sistemlerde NetLogo İle Karınca Kolonisi Optimizasyonu.” Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi. 88-96