

**T.C.
MANİSA CELAL BAYAR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ
YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI**

**YAZILIM TANIMLI İÇERİK DAĞITIM AĞLARINDA (SOFTWARE-
DEFINED CONTENT DELIVERY NETWORK, SD-CDN) eMBB
TRAFİĞİ İÇİN ERİŞİM PROTOKOLÜ TABANLI KONTROLÖR
TASARIMI**

EKBER ÇETİN KÜK

**Danışman
Dr. Öğr. Üyesi Müge ÖZÇEVİK**



MANİSA-2021

**Ekber Çetin
KÜK**

**YAZILIM TANIMLI İÇERİK DAĞITIM AĞLARINDA (SOFTWARE-DEFINED
CONTENT DELIVERY NETWORK, SD-CDN) eMBB TRAFİĞİ İÇİN ERİŞİM
PROTOKOLÜ TABANLI KONTROLÖR TASARIMI**

2021

TAAHHÜTNAME

Bu tezin Manisa Celal Bayar Üniversitesi Hasan Ferdi Turgutlu Teknoloji Fakültesi Yazılım Mühendisliği Bölümü'nde, akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

Ekber Çetin KÜK



İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER.....	I
ŞİMGELER VE KISALTMALAR DİZİNİ.....	II
ŞEKİLLER DİZİNİ	III
TABLO DİZİNİ.....	IV
TEŞEKKÜR	V
ÖZET	VI
ABSTRACT	VII
1. GİRİŞ	1
1.1. Literatür Araştırması.....	2
2. SD-CDN AĞ MİMARİSİ	6
3. SİSTEM MODELİ.....	9
3.1. SD-CDN’de Anycast Yönlendirme	9
3.2. eMBB Kalitesi için PSNR Hesaplama	11
4. PERFORMANS DEĞERLENDİRMESİ.....	14
4.1. Test Ortamı	14
4.1.1. Test Ortamı Bileşenleri	14
4.2. Performans Parametreleri.....	15
4.3. Performans Çıktıları	17
4.3.1. Algoritma Testi	18
4.3.2. Kontrolör Ölçekleme Testi	27
5. SONUÇ VE ÖNERİLER	30
KAYNAKLAR.....	31

SİMGELER VE KISALTMALAR DİZİNİ

BGP	Border Gateway Protocol (Sınır Geçit protokolü)
CDN	Content Delivery Network (İçerik Dağıtım Ağı)
CPE	Customer Premises Equipment (Kenar Ağındaki Uç Cihazı)
DNS	Domain Name System (Alan Adı Sistemi)
eMBB	Enhanced Mobile Broadband (Geliştirilmiş Mobil Geniş Bant)
HTTP	Hyper Text Transfer Protocol (Hiper Metin Transfer Protokolü)
IP	Internet Protokol (İnternet Protokolü)
MIMO	MIMO Multiple Input Multiple Output (Büyük Çoklu Giriş Çoklu Çıkış)
mMTC	Massive Machine Type Communications (Büyük Makine Tipi İletişim)
MSE	MSE Mean Squared Error (Ortalama Kare Hatası)
ONF	The Open Networking Foundation (Açık Ağ Kuruluşu)
ONOS	ONOS Open Network Operating System (Açık Ağ İşletim Sistemi)
OPEX	Operational Expenditures (Operasyonel Harcamalar)
CAPEX	Capital Expenditures (Yatırım Harcamaları)
P2P	Peer to Peer (Eşten Eşe)
PSNR	Peak Signal-to-Noise Ratio (Pik Sinyal Gürültü Oranı)
QoS	Quality of Service (Servis Kalitesi)
QUIC	Quick UDP Internet Connections (Hızlı UDP İnternet Bağlantıları)
RTT	Round Trip Time (Gidiş Dönüş Süresi)
SDN	Software Defined Networking (Yazılım Tanımlı Ağ)
TCP	Transmission Control Protocol (İletim Kontrol Protokolü)
TLS	TLS Transport Layer Security (Taşıma Katmanı Güvenliği)
UDP	User Datagram Protocol (Kullanıcı Veribloğu Protokolü)
URLLC	Ultra-Reliable Low Latency Communications (Ultra Güvenilir Düşük Gecikmeli İletişim)
VoD	Video on Demand (Seç İzle)

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1. Geleneksek ağ ve SDN ağı	7
Şekil 2.2. SD-CDN'in Hostway International topolojisi	7
Şekil 3.1. OpenFlow anahtarlarında önerilen sistemin mimarisi ve anycast yönlendirmesi	9
Şekil 3.2. $y = \log_{10}(x)$ grafiği.....	13
Şekil 3.3. Farklı PSNR değerli için örnek sahne	13
Şekil 4.1. Test ortamı	14
Şekil 4.2. SD-CDN topolojimiz.....	15
Şekil 4.3. Host-1 ve host-2 terminal ekranları.....	16
Şekil 4.4. Host-6 ve host-7 terminal ekranları.....	17
Şekil 4.5. Paket istatistikleri ekranı	17
Şekil 4.6. Durum 1, 2 ve 3'e ait paket istatistikleri	19
Şekil 4.7. Durum 4, 5 ve 6'ya ait paket istatistikleri	19
Şekil 4.8. Tüm durumlara için arka plan trafiğine ait paket istatistikleri.....	20
Şekil 4.9. Durum 1 için PSNR grafiği	20
Şekil 4.10. Durum 2 için PSNR grafiği	21
Şekil 4.11. Durum 3 için PSNR grafiği	22
Şekil 4.12. Durum 4 için PSNR grafiği	23
Şekil 4.13. Durum 5 için PSNR grafiği	24
Şekil 4.14. Durum 6 için PSNR grafiği	25
Şekil 4.15. Bit oranlarının karşılaştırılması.....	26
Şekil 4.16. Geleneksel DNS tabanlı yönlendirme ile farklı durumlar için önerilen SD-CDN arasında alınan eMBB içeriğinin QoS(%) karşılaştırması	27
Şekil 4.17. Durum 7 için PSNR grafiği	28
Şekil 4.18. Durum 8 için PSNR grafiği	29

TABLO DİZİNİ

	Sayfa
Tablo 3.1. Önerilen OpenFlow yığını ve SD-CDN'nin eşleştirme alanları.	10
Tablo 4.1. Gerçekleştirilen test durumları için kullanıcı erişim protokolü ve arka plan trafik bilgileri.....	16
Tablo 4.2. Test durumları ve ortalama PSNR değerleri.....	18



TEŐEKKÜR

Çalıőmamın her aőamasında bana sabırla destek olan, bilgi ve deneyimleri ile yol gosteren danıőman hocam Sayın Dr. Öğr. Üyesi Müge ÖZÇEVİK'e, yüksek lisans eğitimim sırasında desteklerini esirgemeyen Doç. Dr. Bora CANBULA, Hüdayi KALMAN ve Ramazan KEPENEK'e, öğrenim hayatım boyunca maddi ve manevi olarak hep yanımda olan aileme yürekten teşekkür ederim.

Ekber Çetin KÜK
Manisa, 2021



ÖZET

Yüksek Lisans Tezi

YAZILIM TANIMLI İÇERİK DAĞITIM AĞLARINDA eMBB TRAFİĞİ İÇİN ERİŞİM PROTOKOLÜ TABANLI KONTROLÖR TASARIMI

Ekber Çetin KÜK

Manisa Celal Bayar Üniversitesi
Fen Bilimleri Enstitüsü
Yazılım Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Müge ÖZÇEVİK

Günümüzde 5G altyapısında akan video trafiği aşırı bir hızla artmaktadır. 5G’de eMBB olarak adlandırılan bu trafik, geleneksel DNS tabanlı CDN ile yönlendirilmektedir. Bu yapıda BGP, kullanıcı isteklerini coğrafi olarak en yakın yerel CDN kullanıcıya yönlendirmektedir ve bu durumun yerel CDN sunucularında aşırı yüklenme ile sonuçlanması kaçınılmazdır. Literatürdeki çalışmalara göre eMBB içeriğine erişim türlerinin dikkate alınması, CDN önbelleğe almada daha fazla isabet oranı ile sonuçlanmaktadır. Ayrıca, anycast tabanlı yönlendirme, coğrafi dezavantajına rağmen trafik akışlarının bir kısmını yerel sunucu yerine daha uzak olan merkezi sunucuya yönlendirebilmektedir. Bu çalışmaların ışığında, eMBB trafiğinin erişim protokollerine göre yönlendirme gereksinimi ortaya çıkmıştır. Günümüzde, eMBB trafiği isteklerini yönlendirmede, kullanılan uygulamanın bulunduğu işletim sistemine göre değişim gösteren TCP tabanlı HTTP/2.0 ve UDP tabanlı QUIC kullanıcı erişim protokol türlerini dikkate alınmaktadır. Ancak önceki çalışmalar CDN’de erişim tipine göre en uygun sunucuyu seçme konusunda merkezi bir görüşe sahip değiller ve protokol tabanlı yük (toplam eşleşme paketlerinin sayısı) ve heterojenlik (paketlerin erişim protokollerine (HTTP2/QUIC) göre oranı) gibi ağ dinamiklerini dikkate almamaktadırlar. Dolayısıyla; 5G’de eMBB gereksinimlerini karşılayamamaktadır.

Bu nedenle, eMBB trafiği için, Veri ve Kontrol düzlemlerini birbirinden ayıran ve anycast yönlendirmesinin kolay uygulanabildiği erişim protokolüne duyarlı(dikkate alan) Yazılım Tanımlı İçerik Dağıtım Ağları (SD-CDN) öneriyoruz. Veri düzlemindeki OpenFlow ağ cihazlarından alınan istatistiklerle, yeni önerilen ağ cihazlarının yük ve heterojenlik metrikleri periyodik olarak hesaplanmaktadır. Ardından SD-CDN’de yeni erişim protokolüne türüne duyarlı yönlendirme algoritması kullanarak cihazlara OpenFlow kurallarını gömülerek yerel veya merkezi CDN sunucusuna yönlendirme gerçekleştirilmesi amaçlanmamıştır. Gerçek bir test ortamında yapılan performans değerlendirmelerine göre; önerilen SD-CDN, yerel veya merkezi CDN sunucularından alınan eMBB içeriğinin PSNR ile ölçülen kalitesine göre kıyaslandığında; geleneksel DNS tabanlı CDN’den yaklaşık olarak %60 daha iyi QoS ile hizmet vermektedir.

Anahtar Kelimeler: SDN, CDN, eMBB, QUIC, HTTP, PSNR

2021, 32 sayfa

ABSTRACT

M.Sc. Thesis

ACCESS PROTOCOL BASED CONTROLLER DESING FOR eMBB TRAFFIC IN SOFTWARE-DEFINED CONTENT DELIVERY NETWORK

Ekber Çetin KÜK

**Manisa Celal Bayar University
Graduate School of Applied and Natural Sciences
Department of Software Engineering**

Supervisor: Asst. Prof. Müge ÖZÇEVİK

Nowadays, video traffic has been tremendously increased in the 5G infrastructure. This traffic which is newly called eMBB is orchestrated by conventional DNS-based CDN. In this infrastructure, BGP routes user requests to the geographically closest local CDN user; and this outputs inevitable results that there is overloading local CDN servers. According to studies in the literature, considering the access types of eMBB content, brings higher hit rate in CDN caching. Moreover, anycast-based routing can route some of the traffic flows to the remote central server despite its geographical disadvantage instead of the local server. In the light of these studies, there has been a requirement for routing eMBB traffic by taking the access protocols into account. The type of access protocols for eMBB vary as HTTP 2.0 and UDP-based QUIC according to the operating system of the application used. However, they do not have a central view to select optimal server according to access type in CDN and they are unaware of network dynamics such as protocol-based load and heterogeneity; i.e. the number of total matches packets and the ratio of the packets according to access protocols (HTTP2/QUIC). Then, they cannot meet 5G requirements in acceptable level.

Therefore, for eMBB traffic, we propose Software Defined Content Delivery Networks (SD-CDN) that separates the Data and Control planes which consider the access protocol. By the taken statistics from OpenFlow network devices in the data plane, the newly proposed load and heterogeneity metrics of network devices are calculated periodically in the control plane. Then; a novel access protocol-aware routing algorithm in SD-CDN embeds OpenFlow rules in devices which includes the rules performing routing to local or central CDN servers. According to performance evaluations in a real test environment; when the proposed SD-CDN is compared to the PSNR-measured quality of eMBB content from local or central CDN servers; it is observed that it provides approximately 60% better QoS than conventional DNS-based CDN.

Keywords: SDN, CDN, eMBB, QUIC, HTTP, PSNR

2021, 32 pages

1. GİRİŞ

Günümüzde 5G mobil abone sayısı hızla artmaktadır. Yalnızca 2020 yılının 3. çeyreğinde, yaklaşık 11 milyon net yeni mobil abone artışı olmuştur. Ericsson mobilite raporuna göre, mobil abone sayısının artarak 2026 yılı sonunda 8.8 milyar aboneye ulaşması ve 5G ağının 3.5 milyar aboneye ulaşarak toplam mobil abonelerin %40'ını oluşturması beklenmektedir. Ayrıca 2026 yılına gelindiğinde, 5G ağının kapsama alanının dünya nüfusunun %60'ını içerecek şekilde genişleyeceği öngörülmektedir. Mobil abonelik sayısındaki bu hızlı artışla birlikte, bireysel olarak ağ kullanımının da arttığı görülmektedir. Özellikle, Covid-19 salgını kısıtlamaları nedeniyle çevrimiçi eğitime geçilmesi ve evden çalışmanın yaygınlaşması sebebiyle video trafiği hızla artmıştır. Bu hızlı artış neticesinde, 2026 yılına gelindiğinde, video trafiğinin tüm mobil trafiğe oranının %77'ye ulaşması beklenmektedir [1]. 5G'deki video trafiği hem kabul edilebilir gecikmeye sahip hem de daha yüksek bant genişliği gerektiren eMBB Enhanced Mobile Broadband (Geliştirilmiş Mobil Geniş Bant) olarak adlandırılmaktadır [2].

5G'de trafik hizmetleri 3 kategoriye ayrılmaktadır:

- i. URLLC Ultra-Reliable Low Latency Communications (Ultra Güvenilir Düşük Gecikmeli İletişim),
- ii. mMTC Massive Machine Type Communications (Büyük Makine Tipi İletişim),
- iii. eMBB.

Bu üç trafik hizmeti kısaca aşağıdaki şekilde tanımlanmaktadır:

mMTC: Bağlantı halinde olacak çok yüksek miktarda cihazın; kapsama, maliyet etkinliği, düşük güç tüketimi ve uzun süreli kullanılabilirlik gibi noktalarına odaklanan 5G hizmet kategorisidir.

URLLC: Uzaktan kontrol, otonom sürüş ve tactile internet gibi gecikme hassasiyeti taşıyan servislerin ihtiyaçlarını karşılayan bir 5G hizmet kategorisidir. 3G

ile kıyaslandığında 4G (LTE) ağlarında, uçtan uca gecikme iyileştirilmiş olsa da hala 30-100 ms aralığında kalmaktadır [3].

eMBB: Yüksek çözünürlüklü video akışı, sanal gerçeklik ve arttırılmış gerçeklik gibi yüksek bant genişliği gereksinimlerini karşılamaya odaklanan 5G hizmeti kategorisidir. Mevcut 4G sisteminde zorluklardan biri veri başarımı (throughput) iyileştirilmesidir. 5G’de, 4G sistemlerine göre yaklaşık olarak 100 kat kapasite artışı elde etmek için spektral verimliliği artıran ve yeni spektrumları kullanan, massive MIMO Multiple Input Multiple Output (Büyük Çoklu Giriş Çoklu Çıkış), milimetrik dalga iletişim gibi fiziksel katman teknolojilerine ihtiyaç duyulmaktadır [3].

5G’deki eMBB içeriğinin yüksek bant genişliği ve kabul edilebilir gecikme gereksinimleri, CPE Customer Premises Equipment (Kenar Ağındaki Uç Cihazı), fog ağındaki branch’lar ve cloud’daki sunucular gibi CDN Content Delivery Network (İçerik Dağıtım Ağı) bileşenleri tarafından gerçekleştirilir [4].

CDN, bir merkezi sunucu ve bu merkezi sunucuya bağlı coğrafi olarak dağıtılmış yerel sunucuların ağıdır. İçerikler, merkezi sunucudan bu yerel sunuculara belirli kurallar doğrultusunda dağıtılmaktadır. DNS Domain Name System (Alan Adı Sistemi) tabanlı CDN’de kullanıcı istekleri, erişim protokolü türüne bakılmaksızın BGP Border Gateway Protocol (Sınır Geçit protokolü) yönlendirmesi kullanılarak en yakın yerel sunucuya yönlendirilir [5]. Tek başına CDN kullanımı, veri trafiği ve gecikme sorunlarına çözüm getirmek, ayrıca 5G eMBB gereksinimlerini karşılamaya çalışmak için yetersiz kalmaktadır. Bu nedenle CDN’de sorunların çözümü için merkezi ve dinamik olarak bir yönlendirmeye ihtiyaç duyulmaktadır. Bu konu ile alakalı olarak literatürde, içeriği verimli bir şekilde dağıtmak için yerel sunuculardaki CDN önbelleğini optimize etmeye odaklanan birçok makale bulunmaktadır.

1.1. Literatür Araştırması

Erişim türüne duyarlı CDN ön belleğe alma, yüksek önbellek isabet oranı ve önbellek değiştirme işlemlerinin sayısı ile ölçülen disk yükünün büyük ölçüde azalması ile sonuçlanmaktadır [6]. Örneğin, Web kullanıcılarının eğlence videolarına talebi daha yüksekken, mobil kullanıcıların TV dizilerine olan talebi daha düşüktür.

Ayrıca web portalı üzerinden erişilen videoların popüleritesinin, mobil ortama kıyasla çok daha yavaş artmasına karşın, videoların popüleritesi daha istikrarlıdır. Mobil ortamda ise, videolar çok daha erken popüler olmakla birlikte, popüleriteleri daha istikrarsızdır. Literatürdeki bir çalışmada [7], yapılan YouTube analizleri doğrultusunda, CDN önbelleğinin daha iyi kullanılması için hem uzun vadeli popüler videoları hem de günlük popüler videoları depolamayı önermektedir. Literatürdeki diğer bir çalışma[8], hibrit bir CDN-P2P Peer to Peer (Eşten Eşe) VoD Video on Demand (Seç İzle) akış sisteminde peer local caching ve CDN global caching verimliliğini deneysel olarak analiz etmiştir.

İsveç, İsviçre ve Kaliforniya'da bulunan üç Youtube içerik sunucusu göz önüne alınarak bir deney yapılmıştır. Tüm konumlar için, bir veya iki favori içerik sunucusu var gibi görünmesine karşın, bu sunucular her zaman için en yakındakiler değildir. Bu, Youtube CDN'ninin en yakın veri merkezlerini kullanmayı tercih ettiğini ancak aynı zamanda aralarındaki yükü dengelemek için trafiği diğer veri merkezlerine yönlendirdiğini gösterebilir. Sonuç olarak, Youtube CDN'ninin, içerik sunucuları arasındaki trafiği yeniden yönlendirmek için birçok farklı strateji kullandığını ve bazı konumların bazı içerik sunucularını tercih ediyor gibi görünmesine rağmen, her zaman bu içerik sunucusunun en yakın sunucu olduğu anlamına gelmediği söylenebilir [9].

Bu araştırmalara doğrultusunda, eMBB içeriğine erişim protokolü türleri, kullanımdaki uygulamalara göre farklılaştırılmalıdır. Yakın zamana kadar, eMBB içerikleri, TCP Transmission Control Protocol (İletim Kontrol Protokolü) protokolü kullanan bir uygulama katmanı protokolü olan HTTP/2.0 Hyper Text Transfer Protocol (Hiper Metin Transfer Protokolü) tarafından taşınırken, son zamanlarda bu durum, artan ihtiyaçlar doğrultusunda farklılaşmaya başlamıştır. Günümüzde bir çok popüler uygulama UDP User Datagram Protocol (Kullanıcı Veribloğu Protokolü) tabanlı QUIC Quick UDP Internet Connections (Hızlı UDP İnternet Bağlantıları) protokolüne geçmektedir. Örneğin IOS işletim sistemini kullanan mobil kullanıcılar hala CDN'de HTTP/2.0 tabanlı erişim protokolü türüne sahipken, Android tabanlı kullanıcıların eMBB trafiği, QUIC protokolü tarafından taşınabilmektedir.

TCP, güvenlik ve güvenilirlik konusunda ihtiyaçları nispeten karşılamakta yeterliyken, 5G'de artan trafik gereksinimlerini karşılamada yeterli kalmaktadır. TCP,

handshake (el sıkışma) işlemi nedeniyle uzun bir RTT Round Trip Time (Gidiş Dönüş Süresi) gecikmesine sahiptir. Ayrıca, sıralı veri iletimi gerçekleştirilmesinden dolayı, herhangi bir paketin kaybı veya bozulması sonrasında, bu paketin tekrar alınması için mevcut akışı bekletiliyor olması da gecikmelere neden olmaktadır. Bu sorunların üstesinden gelebilmek için UDP tabanlı QUIC adı verilen yeni bir iletim protokolü geliştirilmiştir. Bu protokol geliştirilirken TCP, TLS Transport Layer Security (Taşıma Katmanı Güvenliği) ve HTTP/2.0 gibi birçok protokolün en iyi yanlarından esinlenilmiştir. QUIC, en iyi durumda (“0-RTT”) bir bağlantı kurarken verileri doğrudan göndererek bağlantı gecikmesini azaltmayı amaçlamaktadır. Ayrıca, HTTP/2.0 için optimize edilmiş çoklama özelliklerini ve yeni tıkanıklık kontrol yaklaşımlarına izin verebilecek daha zengin geri bildirim bilgileri sağlar [10].

Geleneksel DNS tabanlı CDN'de BGP yönlendirmesi kullanıldığı için, 5G ağının yerel sunucularında aşırı yüklenme kaçınılmaz olur ve 5G eMBB gereksinimleri karşılanamaz. [11]'de bu zorluğun üstesinden gelmek için anycast (herhangi) tabanlı CDN önerilmiştir. Anycast tabanlı CDN'lerde, tüm önbellek sunucuları aynı anycast IP Internet Protokol (İnternet Protokolü) adresini paylaşmaktadır. Anycast tabanlı CDN'lerin geleneksel DNS tabanlı CDN'lerden, kullanıcıların %20'sinin geleneksel DNS'e göre optimum olmayan sunuculara bağlanmasına rağmen daha iyi performans göstermiştir. Ancak, anycast tabanlı CDN'ler beraberinde aşağıdaki şekilde özetlenebilen bir takım eksikleri getirmektedir:

- Son kullanıcının erişim protokolü türünün dikkate alınmaması,
- Erişim protokolü türüne göre en uygun sunucuyu seçmek için merkezi bir görünüme sahip olmaması,
- Protokol tabanlı yük ve heterojenlik gibi ağ dinamikleri hakkında bilgi alınmaması,
- Kullanıcı erişim protokolü türünün dikkate alınmaması nedeniyle aşırı TCP oturumu kesintisi.

Bu eksiklerin giderilmesi amacıyla bu tezde, SDN Software Defined Networking (Yazılım Tanımlı Ağ) aracılığıyla CDN yönlendirmesi için daha dinamik bir model önerilmektedir. Veri ve Kontrol düzlemlerini birbirinden ayırır ve

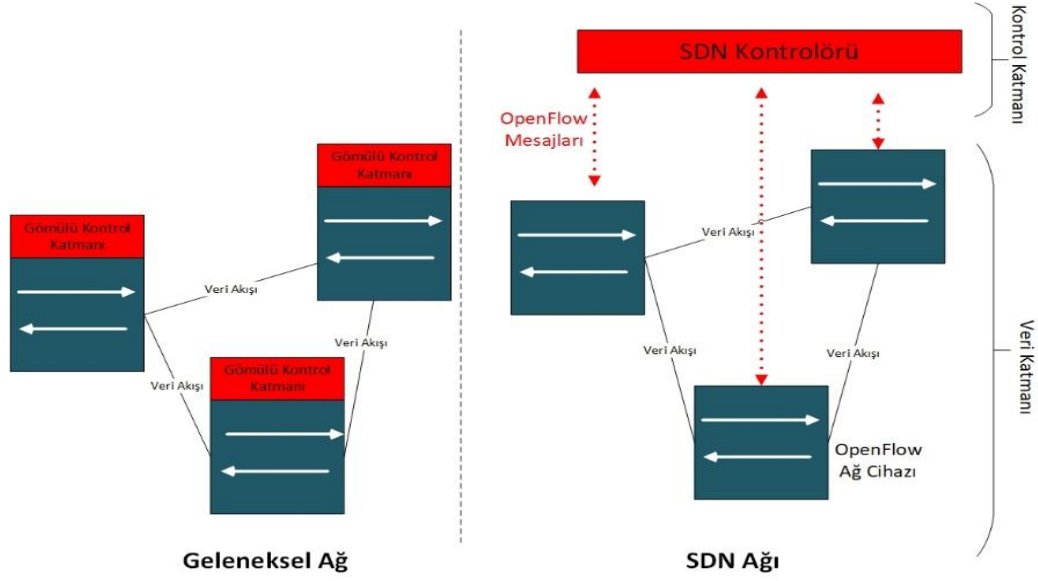
dolayısıyla ađ altyapısında OPEX / CAPEX Operational and Capital Expenditures (Operasyonel ve Yatırım Harcamaları)'i azaltır. Ayrıca, merkezi görünüm, hem trafik yükünü hem de heterojenliđi göz önünde bulundurarak en uygun sunucuyu belirlemek için CDN'deki anycast yönlendirmesini düzenleyebilir. Veri ve Kontrol düzlemleri arasındaki OpenFlow protokolü sayesinde periyodik olarak akış istatistiklerini alarak, kullanıcı erişim protokolüne duyarlı SD-CDN mimarisini dinamik olarak kontrol edebilmektedir. Sonuç olarak bu çalışmada; eMBB trafiđi için erişim protokolü tabanlı SD-CDN önerilmesiyle ařađıdaki katkılar amaçlanmaktadır:

1. SD-CDN'de bir ađ mimarisini, veri ve kontrol düzlemi olarak ayrılmaktadır.
2. CDN'de SDN sayesinde kolayca uygulanabilen anycast tabanlı yönlendirme kullanılmaktadır.
3. CDN yönlendirmesinde dikkate alınması gereken yük ve heterojenlik metriklerini önerilmektedir.
4. SD-CDN'de yeni bir erişim protokolü tabanlı yönlendirme algoritması önerilmektedir.

2. SD-CDN'İN AĞ MİMARİSİ

SDN, ONF The Open Networking Foundation (Açık Ağ Kuruluşu) tarafından ağ denetiminin, yönlendirmeden ayrıldığı ve doğrudan programlanabilir olduğu bir ağ mimarisi olarak tanımlanmaktadır. Yani SDN, ağın kontrol ve veri katmanları olarak sanallaştırılarak ayrılması ve yazılım ile merkezi olarak izlenip, yönetilebilmesi olarak tanımlanabilmektedir [12]. Kontrol katmanı, ağın durumunun merkezden anlık olarak izlenip, yönlendirme için gerekli kuralların belirlendiği katmandır. Veri katmanı ise kontrol katmanı gelen kurallar dahilinde verilerin iletiminin gerçekleştirildiği katmandır.

Ağ cihazları ve kontrolör arasında OpenFlow protokolü ile iletişim sağlanmaktadır. Geneleksi ağ mimarisinde, ağ cihazları yönlendirme işlemleri hesaplamalarını yapmak için yerleşik bir zekaya sahip iken SDN mimarisinde bu durum kontrolör tarafından merkezi olarak gerçekleştirilmektedir ve kurallar ilgili ağ cihazlarına, OpenFlow protokolü aracılığıyla gömülmektedir. Ağ cihazları ise ağın durumu hakkında anlık veri toplayıp yine OpenFlow protokolü aracılığı ile kontrolöre raporlamaktadırlar. Sanallaştırma sayesinde, ağ cihazlarının görevleri, kontrolör tarafından ağın o anki ihtiyaçlara göre yeniden belirlenmektedir. SDN mimarisinde, ağın merkezi olarak izlenebiliyor ve yönetilebiliyor olması, ağa hem esneklik hemde verimlilik katmaktadır.



Şekil 2.1. Geleneksel ağ ve SDN ağı

Bu çalışmada, veri katmanında, CDN için, Şekil 2.2’de gösterilen Hostway International topolojisi kullanılmaktadır. Bu topolojide, eMBB trafiğini, CDN sunucularından kullanıcılara yönlendiren 13 ağ cihazı ve bu anahtarları birbirine bağlayan 16 link bulunmaktadır.



Şekil 2.2. SD-CDN’in Hostway International topolojisi.

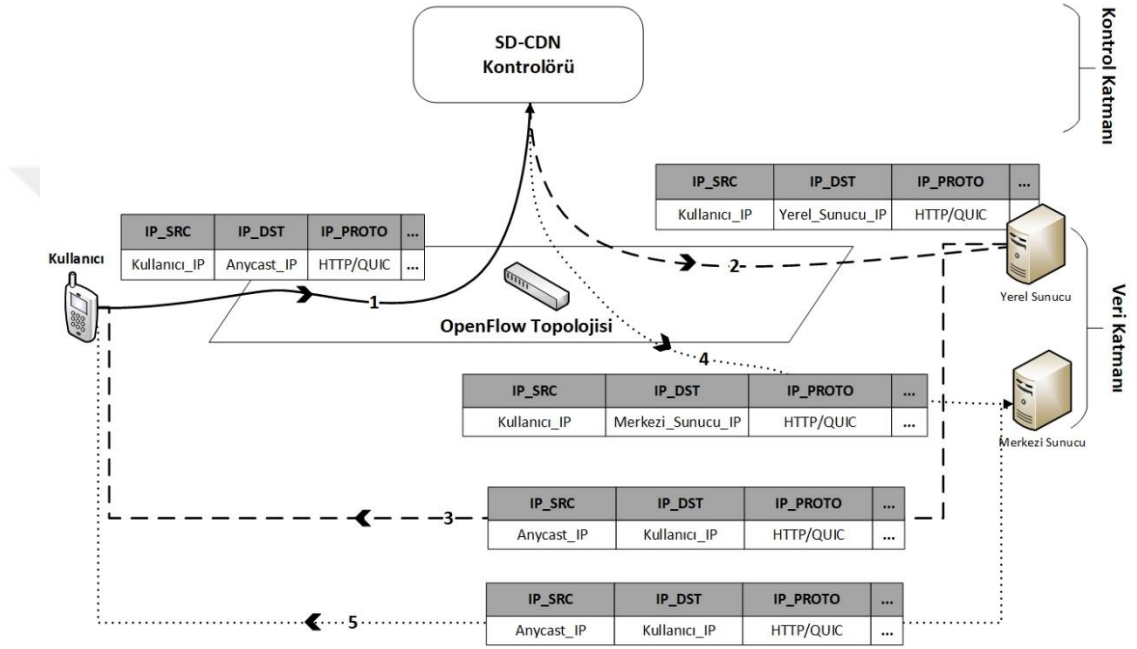
CDN’de bir çok yerel sunucu ve eMBB trafiğinin kıtalara uygun bir şekilde dağıtımından sorumlu olan merkezi bir sunucu bulunmaktadır. Geleneksel olarak, bağlanılan DNS sunucusu, en yakın CDN sunucunun bir IP adresini kullanıcıya döndürmektedir. Bu sayede, eMBB trafiği, yerel sunucudan her bir kullanıcıya yönlendirilmektedir. Ancak bu durumda, hem yerel sunucuda hem de bunun üzerinden yönlendirilen ağ cihazlarında aşırı yüklenme meydana gelmekte ve bu aşırı yüklenme sonucunda da, eMBB trafiğinin QoS Quality of Service (Servis Kalitesi)’inde düşüş görülmektedir.

Bu durumun üstesinden gelebilmek için önerilen SD-CDN, SDN kontrolörü sayesinde hem sunucuları hem de trafik akışlarını dinamik olarak yönetir. Kontrol katmanında ise, kontrolör, OpenFlow anahtarından periyodik olarak istatistik alan erişim protokolü tabanlı yönlendirme algoritmasını çalıştırır. Kontrolör, alınan istatistiksel verilere göre, eMBB içeriği için DNS sunucusuna erişim talebinde bulunan kullanıcıların DNS isteklerini yönetir. Ayrıca, hem trafik yükünü hem de HTTP/2.0 tabanlı veya QUIC tabanlı protokolleri kullanmak gibi trafik heterojenliğini dikkate alarak eMBB trafiğini yönlendirmek için veri katmanındaki OpenFlow rotalarını dinamik olarak günceller.

3. SİSTEM MODELİ

3.1. SD-CDN’de Anycast Yönlendirme

Önerilen sistem mimarisi ve anycast yönlendirme Şekil 3.1’de gösterilmektedir. SD-CDN kontrolü, CDN’in anycast IP ile kullanıcıların ilk DNS isteklerini ve bunu takip eden eMBB isteklerini şu şekilde dinamik olarak yönetmektedir:



Şekil 3.1. OpenFlow anahtarlarında önerilen sistemin mimarisi ve anycast yönlendirmesi.

“1” ile numaralandırılan akışta gösterildiği gibi, kullanıcıdan alınan bir erişim talebi olması durumunda, bu yeni gelen akışın ilk paketi doğrudan SD-CDN kontrolörüne yönlendirilir. Paketin başlığı, hedef IP alanında CDN’nin anycast IP’sini ve ayrıca HTTP/2.0 veya QUIC gibi erişim protokolü türünü içerir. Bu isteğe göre, SD-CDN kontrolörü, yeni bir erişim protokolü tabanlı yönlendirme algoritması çalıştırır. Bu algoritma, veri katmanından OpenFlow istatistiklerini kontrol eder ve OpenFlow kurallarını akılsız cihazlara dinamik olarak gönderir. OpenFlow anahtarının yüküne (ρ) ve erişim protokolü heterojenliğine (H) göre, gelen isteği, “2” ile numaralandırılan akış ile yerel sunucuya veya “4” ile numaralandırılan akış ile merkezi sunucuya yönlendirir. Paket başlığı, hedef olarak, yerel sunucunun veya merkezi

sunucunun fiziksel IP adresine sahiptir. Sonrasında; sunucu, kendi IP adresini kullanmak yerine CDN'in anycast IP adresini kullanarak kullanıcı isteğine, eMBB içeriğini “3” veya “5” ile numaralandırılan akış ile döndürür. OpenFlow tablosundaki eşleştirme alanları Tablo 3.1’de örneklenmiştir.

Tablo 3.1. Önerilen OpenFlow yığını ve SD-CDN'nin eşleştirme alanları.

MATCH FIELDS					ACTION	STATISTICS
IN_PORT	ETH_TYPE	IP_PROTO	IP_SRC	IP_DST	OUTPUT	TX Packets
Port1	0x800	6 (HTTP/2.0)	Kullanıcı_IP	Merkezi_Sunucu_IP	Port3	115286
Port2	0x800	6 (HTTP/2.0)	Anycast_IP	Kullanıcı_IP	Port1	130955
Port3	0x800	17 (QUIC)	Kullanıcı_IP	Merkezi_Sunucu_IP	Port2	1914201
Port4	0x800	17 (QUIC)	Anycast_IP	Kullanıcı_IP	Port4	2115582

OpenFlow anahtarının hem yükünü (ρ) hem de erişim protokolü heterojenliğini (H) hesaplamak için önerilen SD-CDN kontrolöründe aşağıdaki denklemleri kullanılmaktadır:

$$\rho = \frac{\lambda(t-t')}{\mu} \quad t, t' \in N \quad (3.1)$$

$$H = \frac{\lambda_{HTTP2}(t-t')}{\lambda_{QUIC}(t-t')} \quad t, t' \in N \quad (3.2)$$

Burada λ , $(t - t')$ zaman aralığında OpenFlow anahtarına gelen toplam akış ve μ , OpenFlow anahtarının hizmet oranıdır. λ_{HTTP2} , OpenFlow başlıkları kullanılarak HTTP/2.0 akışları olarak ayrıştırılan toplam akış sayısı iken λ_{QUIC} , QUIC akışları olarak ayrıştırılan toplam akış sayısıdır.

SD-CDN kontrolöründe önerilen algoritmanın sözde kodu, Algoritma 3.1’de detaylandırılmıştır. Bu algoritma, OpenFlow istatistiklerini alır ve 2 - 4 satırları arasında görüldüğü gibi her OpenFlow anahtarı için ρ ve H 'yi parametrik olarak hesaplar ve ardından, Denklem 3.1 ve 3.2’de detaylandırıldığı gibi bu metriklere göre

tüm en optimum yolları bulur. OpenFlow kurallarını 6. satırda HTTP/2.0 ve QUIC protokollerini farklılaştırarak gömer.

Algoritma 1: Erişim protokolü tabanlı yönlendirme algoritması

```
Girdi: OpenFlow istatistikleri
Çıktı: HTTP/2.0 / QUIC akışlarının gömülmesi
1: while Kontrolör'e yeni bir akış var do
2:     OpenFlow istatistiklerini al
3:      $List_p \leftarrow YükiHesapla()$ 
4:      $List_H \leftarrow HeterojenliđiHesapla()$ 
5:      $Path \leftarrow List_p$  ve  $List_H$  ile en optimum yolu bul. ▶ En kısa yol
    , geleneksel olarak atlama sayısına göre Dijkstra kullanılarak bulunur.
6:     Yol üzerindeki OpenFlow anahtarlarına HTTP/2.0 ve / veya QUIC
    kurallarını göm.
7: end while
8: function  $YükiHesapla()$ 
9:      $List_p$  'yu tanımla
10:    for all OpenFlow cihaz için paralel yürüt do
11:         $\rho$ 'yu hesapla ▶ denklem 1
12:    end for
13:    return  $List_p$ 
14: end function
15: function  $HeterojenliđiHesapla()$ 
16:      $List_H$  'ı tanımla
17:    for all OpenFlow cihaz için paralel yürüt do
18:         $H$ 'ı hesapla ▶ denklem 2
19:    end for
20:    return  $List_H \leftarrow$  HTTP/2.0 / QUIC
21: end function
```

3.2. eMBB Kalitesi için PSNR Hesaplama

Bir sunucudan kullanıcıya video dosyası gönderilirken, eMBB trafiđi meydana gelmektedir. eMBB trafiđi genellikle çok sayıda frame'in bir araya gelmesi ile oluşmaktadır. Frame'ler, ağ üzerinde paketler halinde iletilmektedir ve genelde bir video frame'inin iletimi, birden fazla pakete bölünerek gerçekleştirilmektedir.

Ağ üzerindeki trafik yükünün artması ve heterojenliđin bilinmemesi sonucunda eMBB trafiđinde paket kayıpları meydana gelmektedir. Bu kayıplar neticesinde de

kullanıcı tarafında, video kalitesinde düşüş meydana gelmektedir. Özellikle ardışık paketlerin kaybı sonrası video frame'lerinde eksik iletim gerçekleşir ve bu durum video izlenmesi sırasında da gözlemlenebilir [13].

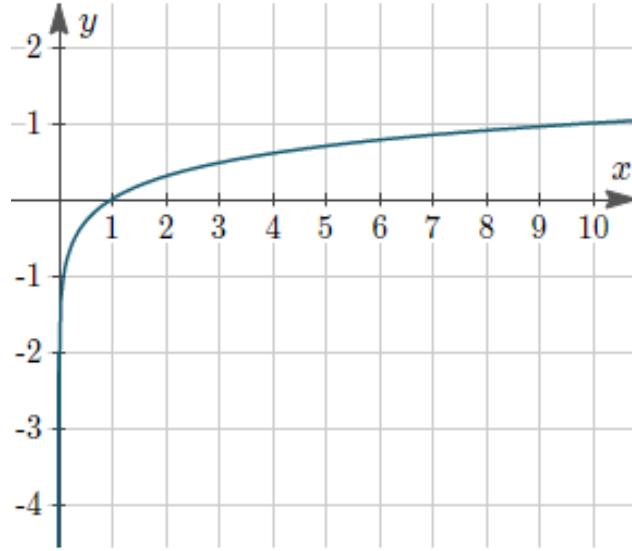
PSNR Peak Signal-to-Noise Ratio (Pik Sinyal Gürültü Oranı), video kalitesini kullanıcı görümüne göre hesaplamada yaygın olarak kullanılan bir ölçümdür. Orijinal bir videonun, format değişikliği, sıkıştırma veya ağ üzerinde iletimi gibi herhangi bir işlem sonrası elde edilen işlenmiş haliyle kıyaslanmasında sıklıkla tercih edilmektedir. Referans alınan orijinal video ile işlenmiş video frame'lerinin karşılıklı olarak ele alındığı PSNR hesabı için, öncelikle Denklem (3.3)'de belirtilen MSE Mean Squared Error (Ortalama Kare Hatası)'e hesaplanmalıdır. Frame'de bulunan tüm piksel değerlerinin hesaba katılması ile elde edilen MSE sonrası, Denklem (3.4) ile PSNR hesaplanmaktadır [14].

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [O(i, j) - D(i, j)]^2 \quad (3.3)$$

$$\begin{aligned} PSNR &= 10. \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20. \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20. \log_{10}(MAX_I) - 10. \log_{10}(MSE) \quad (3.4) \end{aligned}$$

Burada MAX, piksellerin tepe değeridir ve pikseller 8 bit olarak sunulduğunda bu değer 255 dir [14]. O halde Denklem (3.4), aşağıdaki hale gelmektedir.

$$PSNR = 48.1308036087 - 10. \log_{10}(MSE)$$



Şekil 3.2. $y = \log_{10}(x)$ grafiği.

Şekil 3.2’de verilen logaritma grafiğinden de görüleceği üzere, MSE değeri 1’den 0’a doğru azalcıkça PSNR değeri artacaktır. Dahası, iletilen video framlerinde kayıp azaldıkça MSE hesabının sonucu sifıra yaklaşacak şekilde azalır, dolayısıyla PSNR değeri de sonsuza yaklaşacak şekilde artar. Şekil 3.3’de, bir videonun farklı ortalama PSNR değerlerindeki kayıtlarından alınmış, aynı sahneye ait iki görsel gösterilmektedir. Burada (a) görselinin alındığı video için ortalama PSNR değeri 44.509954 iken (b) görselinin alındığı video için ortalama PSNR değeri 26.016668 dir.



(a) Ortalama PSNR= 44.509954



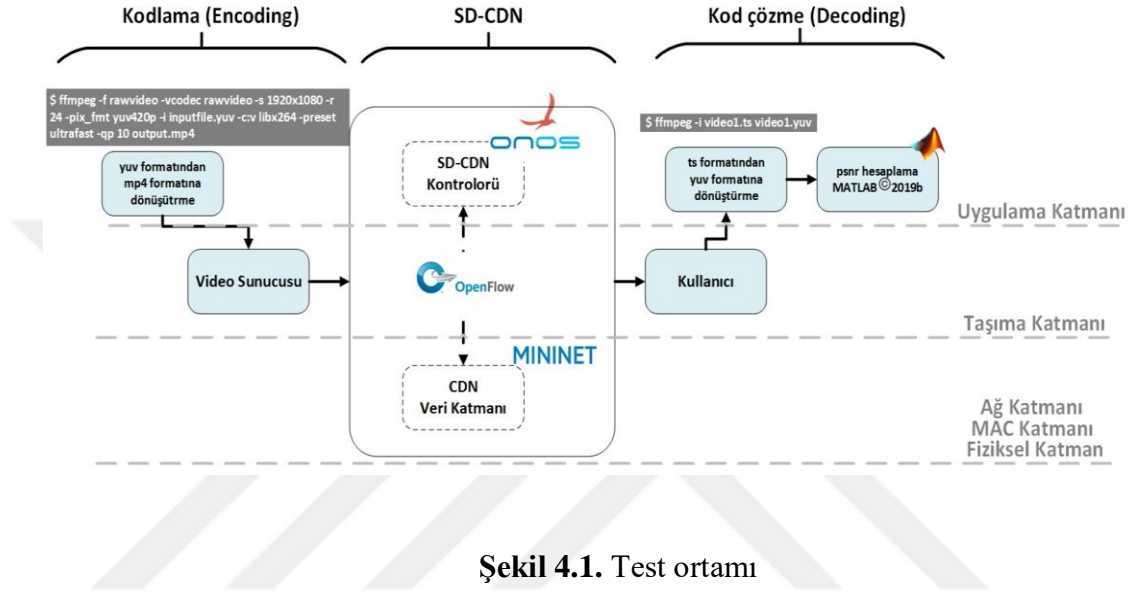
(b) Ortalama PSNR = 26.016668

Şekil 3.3. Farklı PSNR değerli için örnek sahne

4. PERFORMANS DEĞERLENDİRMESİ

4.1. Test Ortamı

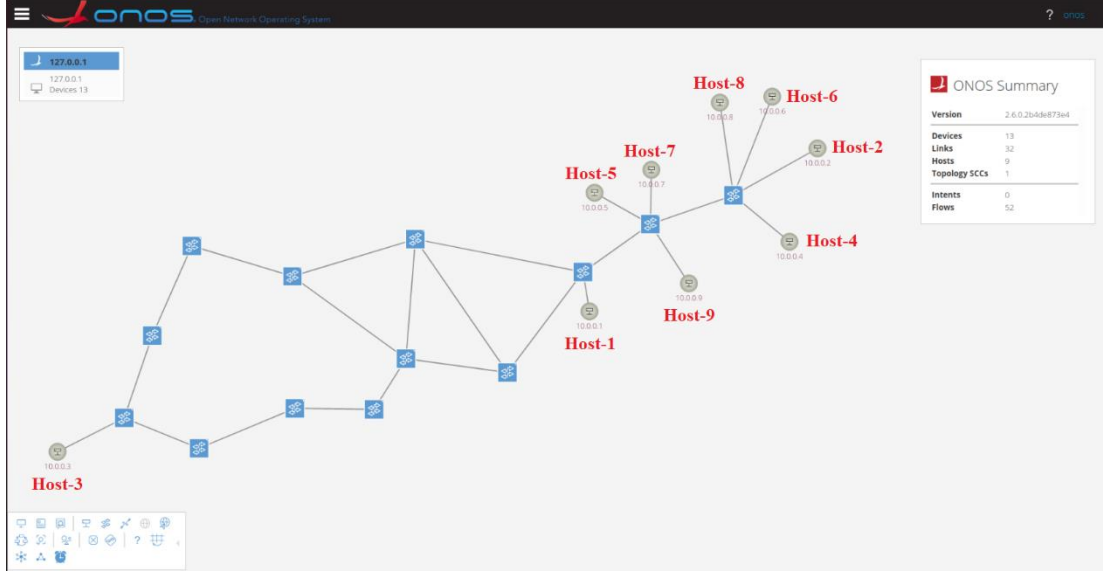
Testlerimizi gerçekleştirmek için Şekil 4.1’de gösterilen ortam hazırlanmıştır. Test ortamımız 3 ana kısımdan oluşmaktadır: sunucu tarafı, SD-CDN tarafı ve kullanıcı tarafı.



4.1.1 Test Ortamı Bileşenleri

Sunucu Tarafı: Performans karşılaştırılmasında referans olarak kullanılmak üzere YUV formatındaki ham videonun bulunduğu bileşendir. Ayrıca videonun akışını gerçekleştirmektedir. Ham video, Şekilde 4.1’de örneklendiği üzere ffmpeg library ile MP4 formatına çevrilip iletme hazır hale getirilmektedir. İletime hazırlanan video, yine ffmpeg library kullanılarak iletilmektedir.

SD-CDN Tarafı: Bölüm 2’de belirtilen topolojiye sahip olmakta. Burada, SD-CDN veri katmanı, ONOS Open Network Operating System (Açık Ağ İşletim Sistemi) tarafından kontrol edilen bir Mininet ortamı kullanılarak oluşturulmuştur. Şekil 4.2’de gösterilen topoloji, 13 ağ cihazı, 4 kullanıcı ve 5 adet sunucudan oluşmaktadır. Veri ve kontrol katmanları, OpenFlow protokolleri aracılığıyla birbirleriyle iletişim kurmaktadır.



Şekil 4.2. SD-CDN topolojimiz

Alıcı Tarafı: Gönderilen videonun .ts uzantısı ile kaydedilmesinden sorumludur. Ardından Şekil 4.1’de örneklendiği üzere ffmpeg library ile YUV formatına dönüştürülerek ham video olarak, performas karşılaştırılmasında kullanılmak üzere kaydedilmektedir.

4.2 Performans Parametreleri

Topolojimizde, toplamda 9 adet host yer almaktadır. Bunlardan host-1, test videosu için kullanıcı olarak tanımlanmışken, host-2 ve host-3 ise CDN içinde, aynı içerikli videoya sahip iki ayrı sunucu olarak tanımlanmıştır. Host-2, kullanıcı olan host-1’e yakın olan yerel sunucu iken, host-1’e göre kullanıcıya daha uzak konumda bulunan host-3 CDN’in merkezi sunucusu olarak tanımlanmıştır. Ayrıca host-4’ten host-5’e, host-6’dan host-7’ye ve host-8’den host-9’a olmak üzere iperf ile trafik oluşturularak, topoloji içerisinde test videosunun trafiği haricinde gerekli olan arka plan trafiği oluşturulmuştur.

Test çalışmalarımız için 14315 frame’den oluşan, çözünürlüğü 1920 x 1080 olan YUV formatında ham bir video seçilmiştir. Öncelikle, ffmpeg kütüphanesi ile YUV formatından MP4 formatına dönüşüm gerçekleştirilerek 09:56 dk uzunluğunda bir MP4 video elde edilmiştir. Dönüşüm gerçekleştirilen video, ardından MP4 formatıyla host-2 ve host-3 sunucularına yerleştirilmiştir.

Tablo 4.1. Gerçekleştirilen test durumları için kullanıcı erişim protokolü ve arka plan trafik bilgileri

Test	Kullanıcı Erişim Protokolü	Arka Plan Trafığı
Algoritma 1 testi	UDP (QUIC)	TCP (HTTP/2.0)
	UDP (QUIC)	UDP (QUIC)
	UDP (QUIC)	-
	TCP (HTTP/2.0)	TCP (HTTP/2.0)
	TCP (HTTP/2.0)	UDP (QUIC)
	TCP (HTTP/2.0)	-
Kontrolör ölçekleme testi	UDP (QUIC)	UDP (QUIC)
	UDP (QUIC)	-

Gerçekleştirilen test durumları ve bu test durumlara ait video trafiği ile arka plan trafiği için gerekli açıklamalar Tablo 4.1’de gösterilmiştir. Şekil 4.3’de sunucu ve alıcı terminal ekranları, Şekil 4.4’de ise iperf ile oluşturulan trafik için örnek terminal ekranları gösterilmiştir. Taşıma katmanı protokolü olan TCP, bizler için uygulama katmanında HTTP/2.0 tabanlı trafik olarak ve taşıma katmanı protokolü UDP bizler için uygulama katmanında QUIC tabanlı trafik olarak değerlendirilmiştir.

```

"Node: h1"
frame= 4921 fps= 22 q=-1.0 size= 822784kB time=00:03:48.54 bitrate=29492.4kbits
frame= 4931 fps= 22 q=-1.0 size= 824576kB time=00:03:49.16 bitrate=29476.0kbits
frame= 4942 fps= 22 q=-1.0 size= 826368kB time=00:03:49.70 bitrate=29470.4kbits
frame= 4955 fps= 22 q=-1.0 size= 827904kB time=00:03:50.25 bitrate=29455.8kbits
frame= 4968 fps= 22 q=-1.0 size= 828672kB time=00:03:50.79 bitrate=29413.9kbits
frame= 4981 fps= 22 q=-1.0 size= 829184kB time=00:03:51.33 bitrate=29363.1kbits
frame= 4994 fps= 22 q=-1.0 size= 829362kB time=00:03:51.87 bitrate=29321.7kbits
frame= 5006 fps= 22 q=-1.0 size= 830464kB time=00:03:52.37 bitrate=29276.6kbits
frame= 5017 fps= 22 q=-1.0 size= 832256kB time=00:03:52.95 bitrate=29265.4kbits
frame= 5027 fps= 22 q=-1.0 size= 832596kB time=00:03:53.50 bitrate=29243.4kbits
frame= 5039 fps= 22 q=-1.0 size= 834816kB time=00:03:54.04 bitrate=29220.5kbits
frame= 5051 fps= 22 q=-1.0 size= 836096kB time=00:03:54.58 bitrate=29197.7kbits
frame= 5062 fps= 22 q=-1.0 size= 836864kB time=00:03:55.08 bitrate=29162.4kbits
frame= 5066 fps= 22 q=-1.0 size= 837632kB time=00:03:55.25 bitrate=29168.5kbits
frame= 5081 fps= 22 q=-1.0 size= 839680kB time=00:03:56.20 bitrate=29121.1kbits
frame= 5089 fps= 22 q=-1.0 size= 841472kB time=00:03:56.83 bitrate=29106.3kbits
frame= 5102 fps= 22 q=-1.0 size= 843008kB time=00:03:57.41 bitrate=29087.8kbits
frame= 5115 fps= 22 q=-1.0 size= 845056kB time=00:03:57.95 bitrate=29092.1kbits
frame= 5126 fps= 22 q=-1.0 size= 846528kB time=00:03:58.50 bitrate=29079.7kbits
frame= 5134 fps= 22 q=-1.0 size= 848128kB time=00:03:59.04 bitrate=29065.5kbits
frame= 5144 fps= 22 q=-1.0 size= 849664kB time=00:03:59.58 bitrate=29082.3kbits
frame= 5155 fps= 22 q=-1.0 size= 851200kB time=00:04:00.08 bitrate=29044.2kbits
frame= 5166 fps= 22 q=-1.0 size= 852736kB time=00:04:00.66 bitrate=29026.1kbits
$ speed= 1.01x

"Node: h2"
frame= 5502 fps= 24 q=-1.0 size= 1202485kB time=00:03:49.20 bitrate=42977.3kbits
frame= 5514 fps= 24 q=-1.0 size= 1204345kB time=00:03:49.70 bitrate=42950.1kbits
frame= 5526 fps= 24 q=-1.0 size= 1206431kB time=00:03:50.20 bitrate=42921.0kbits
frame= 5538 fps= 24 q=-1.0 size= 1207048kB time=00:03:50.70 bitrate=42889.9kbits
frame= 5551 fps= 24 q=-1.0 size= 1207963kB time=00:03:51.25 bitrate=42777.7kbits
frame= 5563 fps= 24 q=-1.0 size= 1208188kB time=00:03:51.75 bitrate=42639.2kbits
frame= 5575 fps= 24 q=-1.0 size= 1208898kB time=00:03:52.25 bitrate=42606.5kbits
frame= 5587 fps= 24 q=-1.0 size= 1210531kB time=00:03:52.75 bitrate=42579.3kbits
frame= 5599 fps= 24 q=-1.0 size= 1212356kB time=00:03:53.25 bitrate=42579.3kbits
frame= 5611 fps= 24 q=-1.0 size= 1213841kB time=00:03:53.75 bitrate=42540.3kbits
frame= 5623 fps= 24 q=-1.0 size= 1215323kB time=00:03:54.25 bitrate=42511.3kbits
frame= 5635 fps= 24 q=-1.0 size= 1216708kB time=00:03:54.79 bitrate=42461.6kbits
frame= 5650 fps= 24 q=-1.0 size= 1218098kB time=00:03:55.37 bitrate=42394.7kbits
frame= 5671 fps= 24 q=-1.0 size= 1220907kB time=00:03:56.25 bitrate=42335.1kbits
frame= 5683 fps= 24 q=-1.0 size= 1222707kB time=00:03:56.75 bitrate=42308.0kbits
frame= 5696 fps= 24 q=-1.0 size= 1224897kB time=00:03:57.29 bitrate=42287.0kbits
frame= 5708 fps= 24 q=-1.0 size= 1226882kB time=00:03:57.79 bitrate=42266.5kbits
frame= 5720 fps= 24 q=-1.0 size= 1228777kB time=00:03:58.29 bitrate=42242.3kbits
frame= 5732 fps= 24 q=-1.0 size= 1230707kB time=00:03:58.79 bitrate=42220.7kbits
frame= 5744 fps= 24 q=-1.0 size= 1232537kB time=00:03:59.29 bitrate=42195.1kbits
frame= 5756 fps= 24 q=-1.0 size= 1234687kB time=00:03:59.79 bitrate=42180.6kbits
frame= 5768 fps= 24 q=-1.0 size= 1236450kB time=00:04:00.29 bitrate=42162.9kbits
frame= 5780 fps= 24 q=-1.0 size= 1238998kB time=00:04:00.79 bitrate=42121.5kbits
$ speed= 1x
    
```

Şekil 4.3. Host-1 ve host-2 terminal ekranları

```

"Node: h6"
root@chtUbuntu:/home/cht/mininet-test# iperf -c 10.0.0.7 -u -b 2G -t 630
-----
Client connecting to 10.0.0.7, UDP port 5001
Sending 1470 byte datagrams, IPG target: 5,48 us (kalman adjust)
UDP buffer size: 208 KByte (default)
-----
[ 5] local 10.0.0.6 port 55819 connected with 10.0.0.7 port 5001

"Node: h7"
root@chtUbuntu:/home/cht/mininet-test# iperf -s -u
-----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 5] local 10.0.0.7 port 5001 connected with 10.0.0.6 port 55819

```

Şekil 4.4. Host-6 ve host-7 terminal ekranları

Şekil 4.5’de, durum 2 için ONOS ekranında akan paket istatistiklerine ait bir ekran alıntısına da yer verilmiştir.

Flows for Device of:0000000000000003 (8 Total)

STATE	PACKETS	DURATION	FLOW PRIORITY	TABLE NAME	SELECTOR	TREATMENT
Added	189	817	40000	0	ETH_TYPE:arp	imm[OUTPUT:CONTROLLER] cleared:true
Added	529	817	40000	0	ETH_TYPE:lldp	imm[OUTPUT:CONTROLLER] cleared:true
Added	529	817	40000	0	ETH_TYPE:lldp	imm[OUTPUT:CONTROLLER] cleared:true
Added	6,012	807	5	0	ETH_TYPE:ipv4	imm[OUTPUT:CONTROLLER] cleared:true
Added	858,540	239	10	0	IN_PORT:2, ETH_DST:00:00:00:00:00:01, ETH_SRC:00:00:00:00:00:02	imm[OUTPUT:1] cleared:false
Added	30,313,130	249	10	0	IN_PORT:2, ETH_DST:00:00:00:00:00:07, ETH_SRC:00:00:00:00:00:06	imm[OUTPUT:4] cleared:false
Added	30,461,041	250	10	0	IN_PORT:2, ETH_DST:00:00:00:00:00:09, ETH_SRC:00:00:00:00:00:08	imm[OUTPUT:5] cleared:false
Added	30,473,712	252	10	0	IN_PORT:2, ETH_DST:00:00:00:00:00:05, ETH_SRC:00:00:00:00:00:04	imm[OUTPUT:3] cleared:false

Şekil 4.5. Paket istatistikleri ekranı

4.3 Performans Çıktıları

Algoritma 1’in testi için 6 adet ve kontrolör ölçekleme testi için 2 adet olmak üzere toplamda 8 farklı durum testi gerçekleştirilmiştir. Host-1 kullanıcısı üzerinden yapılan istekler sonrasında alınan veriler .ts formatında kaydedildikten sonra ffmpeg kütüphanesi kullanılarak YUV formatına dönüştürülmüştür. Daha sonra sunucular tarafında bulunan orijinal YUV formatındaki video ile alıcı host-1 üzerindeki iletilmiş YUV formatındaki videolar karşılaştırılmış, PSNR ve ortalama PSNR değerleri hesaplanmıştır. Ayrıca tüm frame’ler için hesaplanan PSNR değerlerinin grafikleri, x-ekseninde frame numarasına karşılık y-ekseninde PSNR değeri olacak şekilde

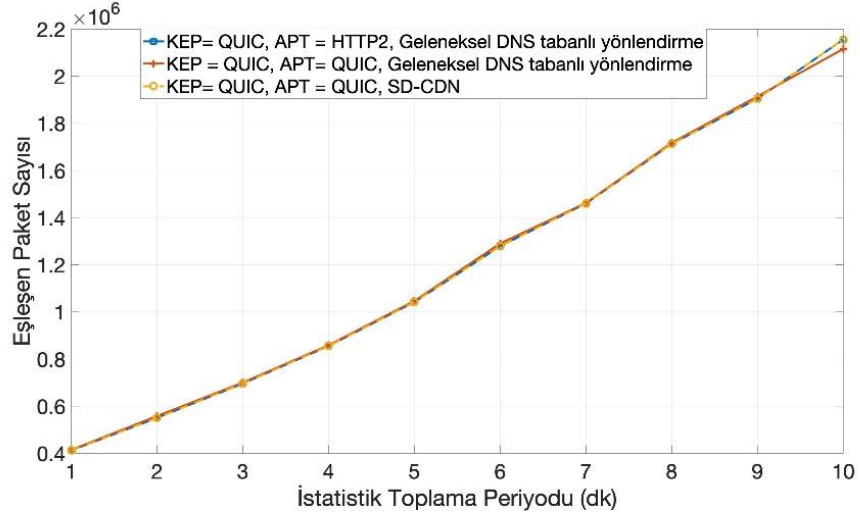
MATLAB ortamında çizilmiştir. Geleneksel DNS tabanlı yönlendirmeye ile kontrolör, yalnızca atlama sayısını (hop count) dikkate alarak topolojimizin graf yapısı üzerinde Dijkstra algoritması yürütmektedir ve en yakın sunucuya yönlendirme gerçekleştirmektedir. Önerilen yapıda ise, ağ cihazlarından toplanan istatistikler ile ağın trafik durumu ve heterojenliğinin, kontrolör tarafında algoritma 1'de değerlendirilmesi sonucunda yönlendirme gerçekleştirilmektedir. Tablo 4.2'de test durumları ve durumlara ait ortalama PSNR değerleri verilmektedir.

Tablo 4.2. Test durumları ve ortalama PSNR değerleri

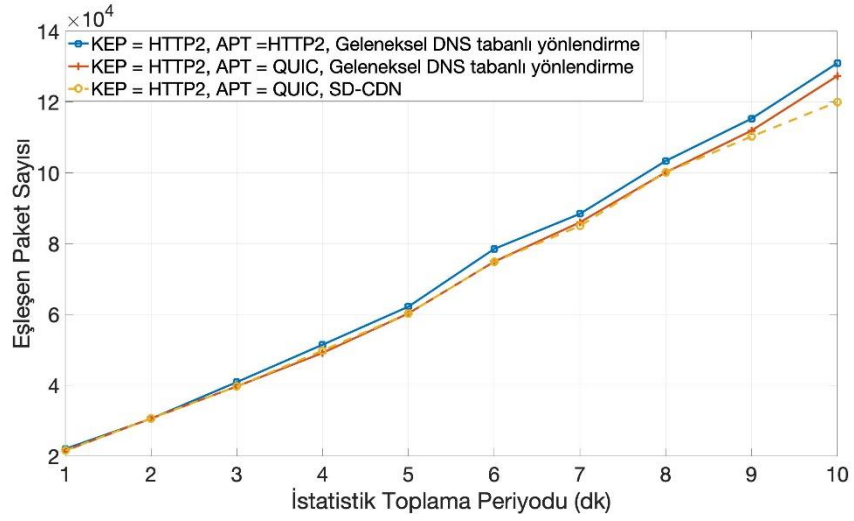
Durum	Test	Kullanıcı	Kullanıcı Erişim Protokolü	Sunucu	Arka Plan Trafiği	Ortalama PSNR
1	Algoritma 1 testi	Host-1	UDP (QUIC)	Host-2	TCP (HTTP/2.0)	26.016668
2		Host-1	UDP (QUIC)	Host-2	UDP (QUIC)	14.062058
3		Host-1	UDP (QUIC)	Host-3	-	44.509954
4		Host-1	TCP (HTTP/2.0)	Host-2	TCP (HTTP/2.0)	51.479389
5		Host-1	TCP (HTTP/2.0)	Host-2	UDP (QUIC)	51.479389
6		Host-1	TCP (HTTP/2.0)	Host-3	-	51.479389
7	Kontrolör ölçekleme testi	Host-1	UDP (QUIC)	Host-3	UDP (QUIC)	14.478212
8		Host-1	UDP (QUIC)	Host-3	-	45.816581

4.3.1 Algoritma Testi

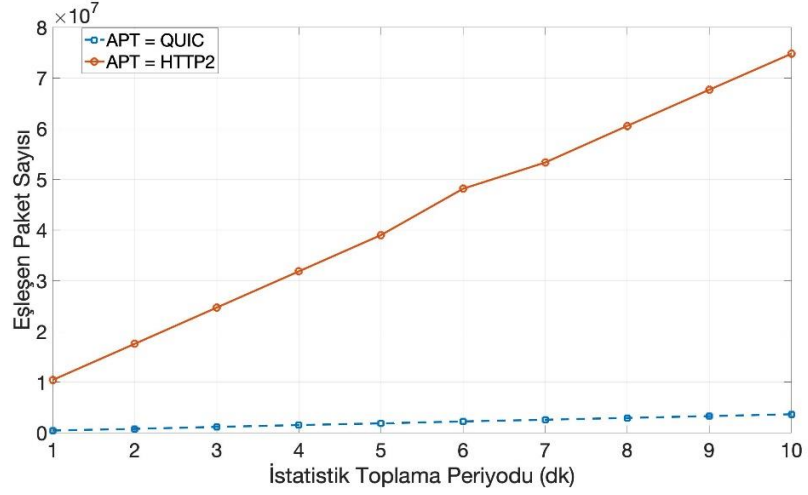
Algoritma 1 için toplanan paket istatistik grafikleri Şekil 4.6, Şekil 4.7 ve Şekil 4.8'de gösterilmektedir. Bu istatistikler ışığında yapılan yönlendirme sonuçları aşağıdaki durumlarda açıklanmaktadır.



Şekil 4.6. Durum 1, 2 ve 3'e ait paket istatistikleri

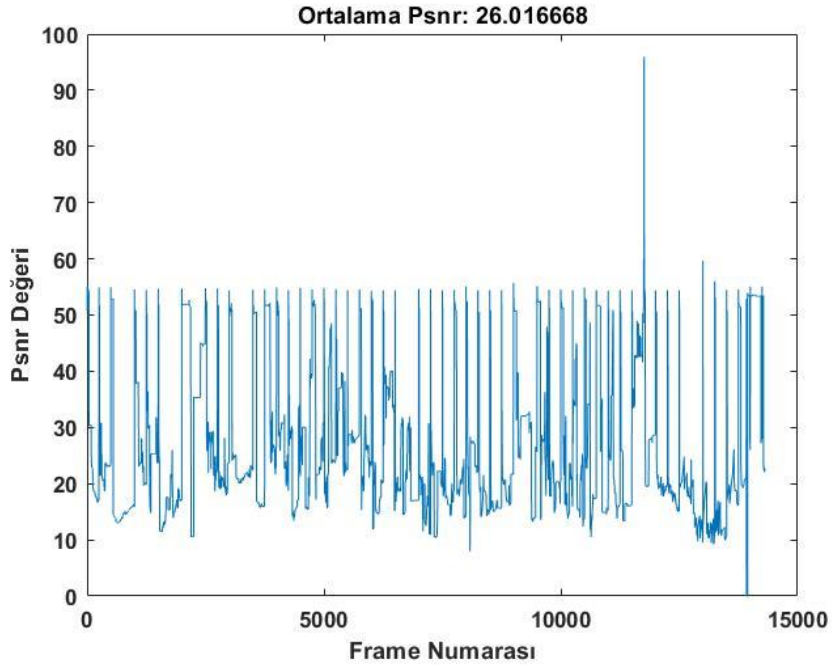


Şekil 4.7. Durum 4, 5 ve 6'ya ait paket istatistikleri



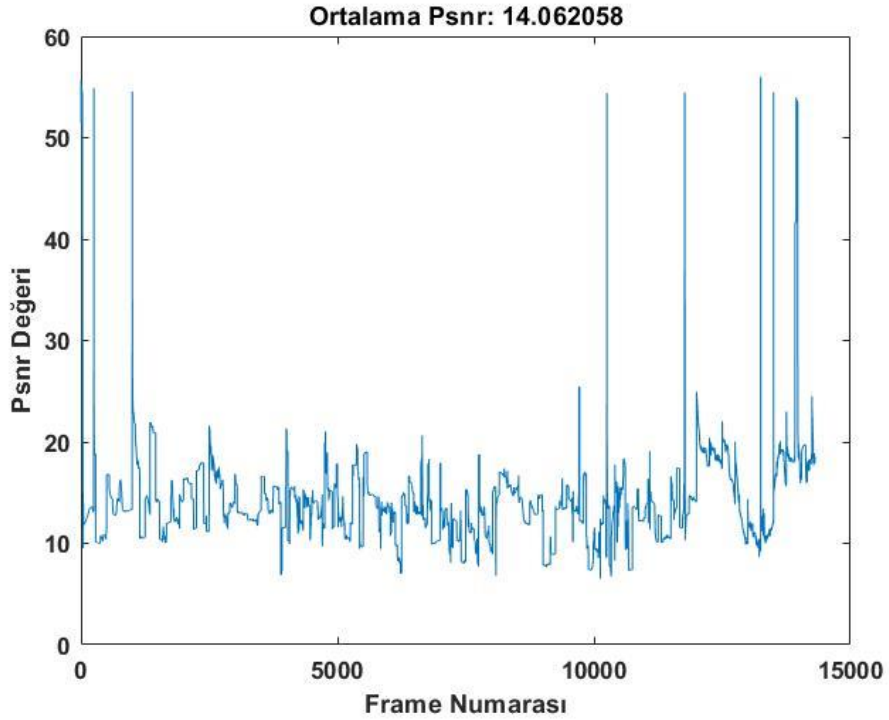
Şekil 4.8. Tüm durumlara için arka plan trafiğine ait paket istatistikleri

Durum 1: Kullanıcı tarafından, eMBB içeriğine erişmek için UDP tabanlı QUIC protokolü ile yapılan ilk istekte kontrolör, geleneksel DNS tabanlı yönlendirme ile yerel sunucuya yönlendirme gerçekleştirmiştir. TCP tabanlı HTTP/2.0 arka plan trafiği varken, yerel sunucu tarafından sağlanan videoya karşın ortalama PSNR değeri 26.016668 olarak hesaplanmıştır ve frame'lere ait PSNR grafiği ise Şekil 4.9'da gösterilmiştir.



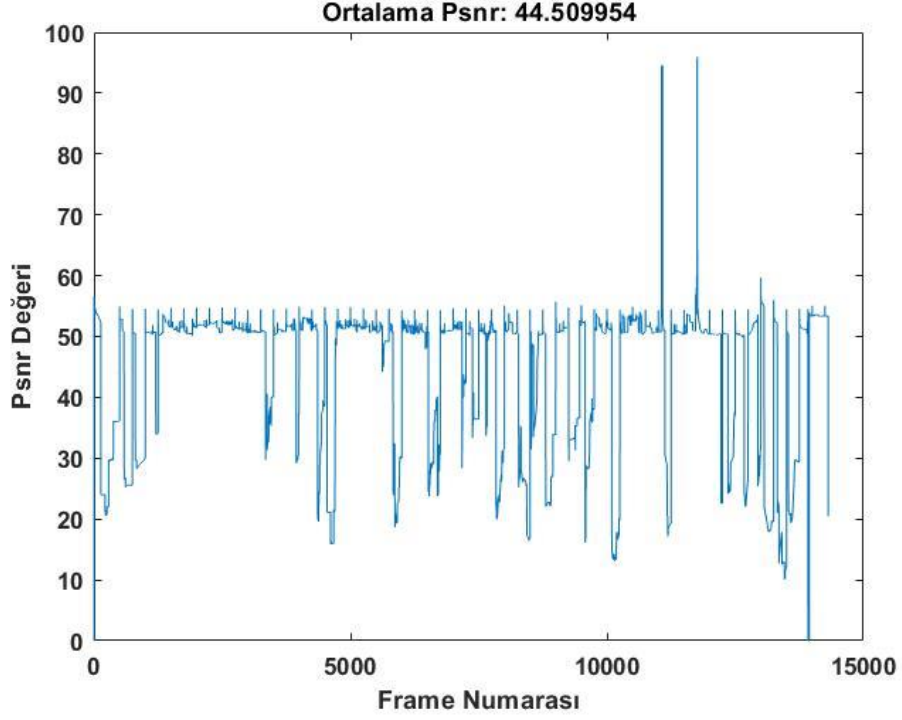
Şekil 4.9. Durum 1 için PSNR grafiği

Durum 2: Kullanıcı tarafından, eMBB içeriğine erişmek için UDP tabanlı QUIC protokolü ile yapılan ikinci istekte kontrolör, geleneksel DNS tabanlı yönlendirme ile yerel sunucuya yönlendirme gerçekleştirmiştir. UDP tabanlı QUIC arka plan trafiği varken, yerel sunucu tarafından sağlanan videoya karşın ortalama PSNR değeri 14.062058 olarak hesaplanmıştır ve frame'lere ait PSNR grafiği ise Şekil 4.10'da gösterilmektedir.



Şekil 4.10. Durum 2 için PSNR grafiği

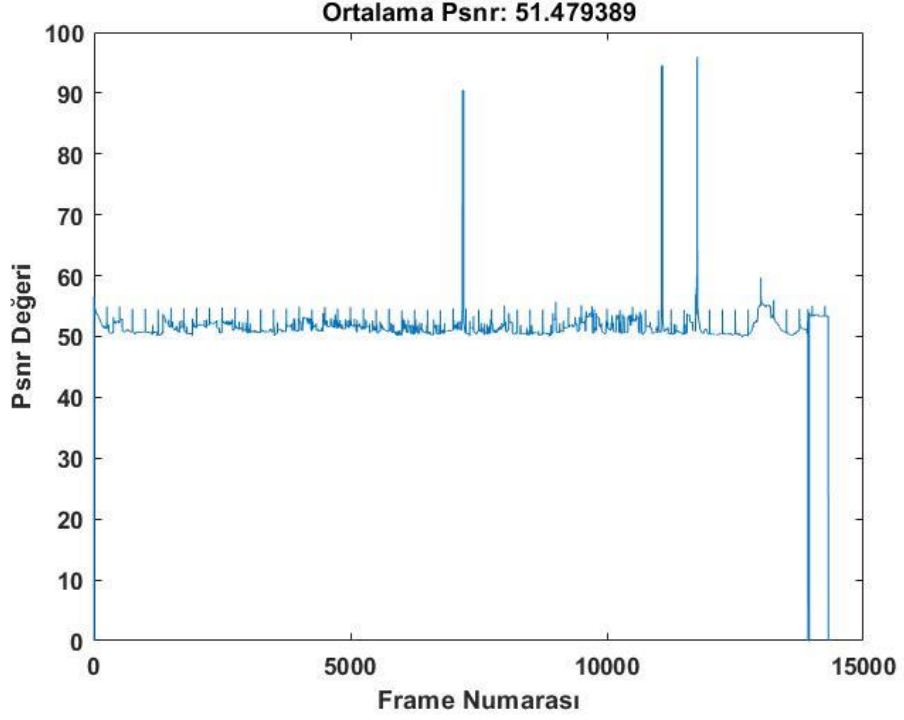
Durum 3: Kullanıcı tarafından, eMBB içeriğine erişmek için UDP tabanlı QUIC protokolü ile yapılan üçüncü istekte kontrolör, Algoritma 1'in kararına göre merkezi sunucuya yönlendirme gerçekleştirmiştir. Merkezi sunucu tarafından sağlanan videoya karşın ortalama PSNR değeri 44.509954 olarak hesaplanmıştır ve frame'lere ait PSNR grafiği ise Şekil 4.11'de gösterilmektedir.



Şekil 4.11. Durum 3 için PSNR grafiği

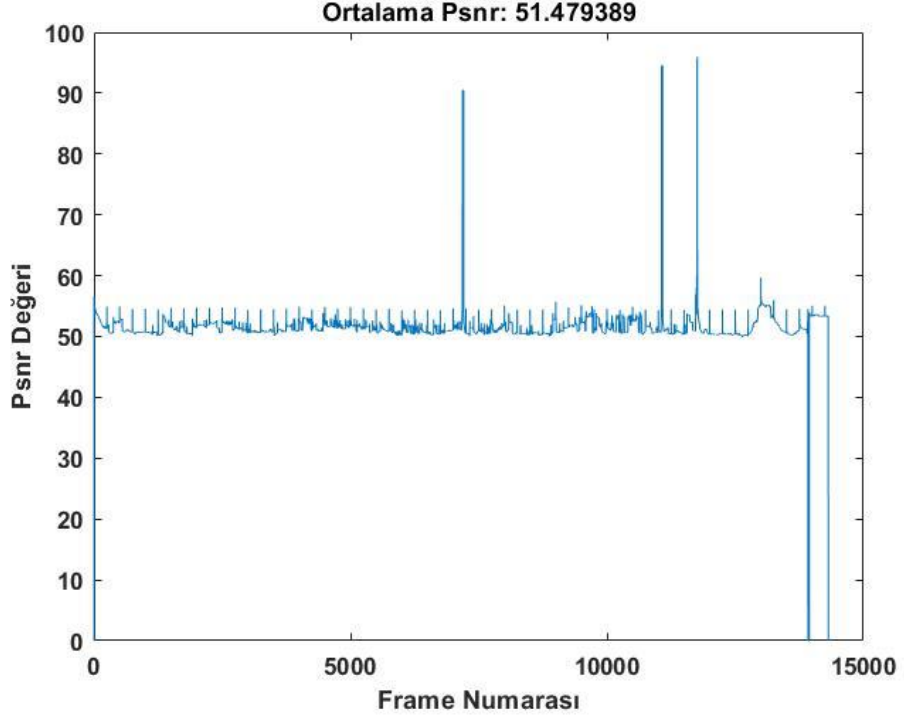
TCP tabanlı arka plan trafiğine sahip Durum 1’de ortama PSNR değeri 26.016668 olarak hesaplanmışken arka plan trafiğinin UDP tabanlı olduğu durum 2’de PSNR değeri 14.062058’e gerileyerek daha kötü bir hale gelmektedir. Çünkü eMBB trafiğini yönlendirmek için ağ dinamikleri dikkate alınmamıştır. Diğer taraftan, ağ dinamikleri dikkate alınarak Algoritma 1 kararına göre, merkezi sunucuya yapılan yönlendirme ile durum 3’de PSNR değeri 44.509954 olarak hesaplanmıştır. Burada seçilen merkezi sunucu coğrafi olarak daha dezavantajlı olmasına rağmen, arka plan trafik yükü olmadığı için eMBB trafiğinin daha iyi bir QoS’e sahiptir.

Durum 4: Kullanıcı tarafından, eMBB içeriğine erişmek için TCP tabanlı HTTP/2.0 protokolü ile yapılan dördüncü istekte kontrolör, geleneksel DNS tabanlı yönlendirme ile yerel sunucuya yönlendirme gerçekleştirmiştir. TCP tabanlı HTTP/2.0 arka plan trafiği varken, yerel sunucu tarafından sağlanan videoya karşın ortalama PSNR değeri 51.479389 olarak hesaplanmıştır ve frame’lere ait PSNR grafiği ise Şekil 4.12’de gösterilmektedir.



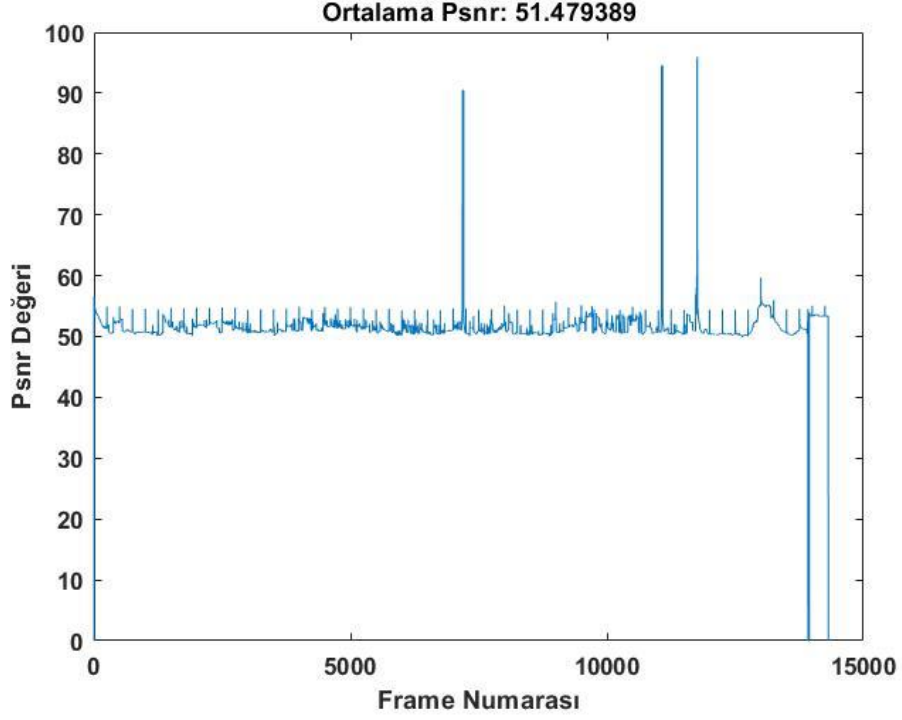
Şekil 4.12. Durum 4 için PSNR grafiği

Durum 5: Kullanıcı tarafından, eMBB içeriğine erişmek için TCP tabanlı HTTP/2.0 protokolü ile yapılan beşinci istekte kontrolör, geleneksel DNS tabanlı yönlendirme ile yerel sunucuya yönlendirme gerçekleştirmiştir. UDP tabanlı QUIC arka plan trafiği varken, yerel sunucu tarafından sağlanan videoya karşın ortalama PSNR değeri 51.479389 olarak hesaplanmıştır ve frame'lere ait PSNR grafiği ise Şekil 4.13'de gösterilmektedir.



Şekil 4.13. Durum 5 için PSNR grafiği

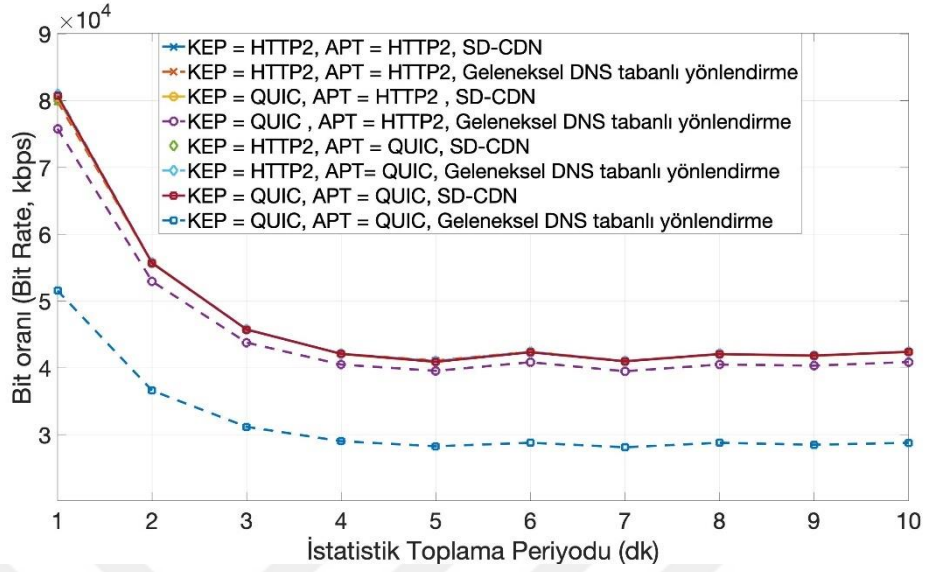
Durum 6: Kullanıcı tarafından, eMBB içeriğine erişmek için TCP tabanlı HTTP protokolü ile yapılan altıncı istekte kontrolör, Algoritma 1'in kararına göre merkezi sunucuya yönlendirme gerçekleştirmiştir. Merkezi sunucu tarafından sağlanan videoya karşın ortalama ortalama PSNR değeri 51.479389 olarak hesaplanmıştır ve frame'lere ait PSNR grafiği ise Şekil 4.14'de gösterilmiştir.



Şekil 4.14. Durum 6 için PSNR grafiği

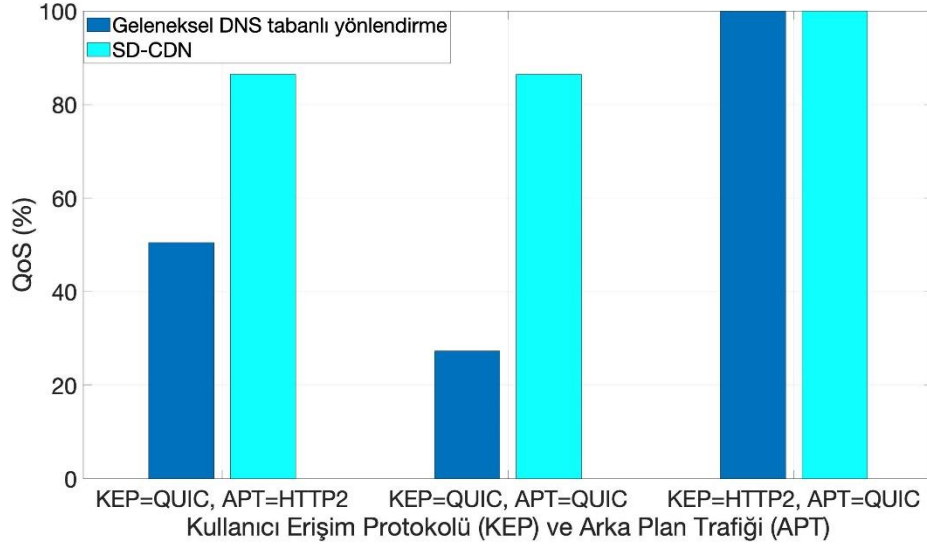
Farklı arka plan trafik türü ve eMBB içerik sunucusuna sahip olan durum 4, durum 5 ve durum 6'da aynı PSNR değerleri elde edilmiştir. Bu sonuç, 3 durumda da eMBB trafiğinin TCP tabanlı olmasından kaynaklanmaktadır. Güvenilir veri iletimi sağlarken aynı zamanda tıkanıklık kontrolü de yapan TCP protokolünün bu avantajları yanında bir takım dezavantajları da bulunmaktadır. Sıralı veri iletimi gerçekleştiren TCP'nin, herhangi bir paket kaybı veya bozulması sonrasında, mevcut akışı kayıp veya bozuk paket için bekletiliyor olması, iletimin toplam süresinde artışa yol açmaktadır. Bu nedenle TCP ile iletim, düzenli (smoothly) bir şekilde değildir, uzun boşa kalma zamanları içerebilmektedir. PSNR sonucu olarak değerlendirildiğinde, TCP ile iletimde her ne kadar sonuçlar benzer seviyelerde olsalarda, ağın durumuna göre gerçekleşecek kayıplar ile aynı oranda iletim süresinde gecikmeler yaşanacaktır [15][13]. Bu yüzden algoritma 1 kullanılarak yapılan yönlendirme sonucunda, geleneksel DNS tabanlı yönlendirme ile aynı PSNR sonuçları elde edilse de iletim süresi olarak daha kısa bir süre elde edilecektir [15]. Bu durumu destekleyen bir başka parametre olarak da bit oranı verilebilir. Şekil 4.15'de verilen bit oranları grafiğinden de görüleceği üzere kullanıcı erişim protokolü ve arkaplan trafiği UDP tabanlı QUIC

olduğunda, SD-CDN, geleneksel DNS tabanlı yönlendirmeden daha iyi bir bit oranına sahiptir.



Şekil 4.15. Bit oranlarının karşılaştırılması

Şekil 4.16'da geleneksel DNS tabanlı yönlendirme ile önerilen SD-CDN için farklı kullanıcı erişim protokolü ve arka plan trafik türlerine göre değişen durumlarda elde edilen eMBB içeriğinin QoS(%)'inin karşılaştırılması verilmektedir. Bunlar, iletilen video içeriklerinden elde edilen PSNR sonuçlarından hesaplanmıştır. Bu grafik y-ekseninde QoS değerini ve x-ekseninde ise farklı durumları gösterir. eMBB içeriğine QUIC protokolü ile erişim sağlandığında, geleneksel DNS yönlendirmesi ile birlikte arka planda HTTP/2.0 trafiği varken %50,53 iken QUIC trafiği varken %27,31 QoS değerleri ile son kullanıcıya hizmet verilmektedir. O halde geleneksel DNS yönlendirmesi, arka planda QUIC tabanlı trafik varken, HTTP/2.0 arka plan trafiğine göre daha kötü performans gösterir. Önerilen SD-CDN, anycast yönlendirmesinde yük ve heterojenlik değerlendirme sayesinde %86,47 eMBB içeriği kalitesi ile son kullanıcıya hizmet verebilir. SD-CDN sayesinde son kullanıcıya geleneksel DNS tabanlı yönlendirmeye göre yaklaşık %60 daha iyi QoS sunar. Diğer taraftan; kullanıcı erişim protokolü HTTP/2.0 ve arka plan trafiği QUIC olduğunda, geleneksel DNS yönlendirmesi ve önerilen SD-CDN için elde edilen QoS sonuçları aynıdır. Burada, TCP kullanımı nedeniyle aynı sonuçlar elde edilmiş olsa da, önerilenin üstünlüğü, eMBB içeriğinin iletim süresinin daha az olacaktır.

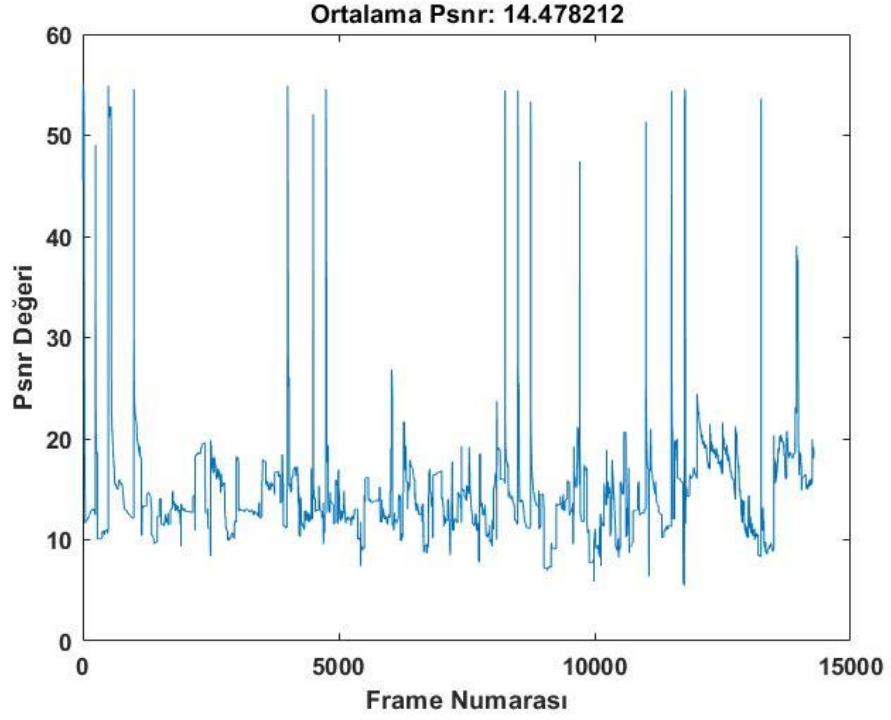


Şekil 4.16. Geleneksel DNS tabanlı yönlendirme ile farklı durumlar için önerilen SD-CDN arasında alınan eMBB içeriğinin QoS (%) karşılaştırması.

4.3.2 Kontrolör Ölçekleme Testi

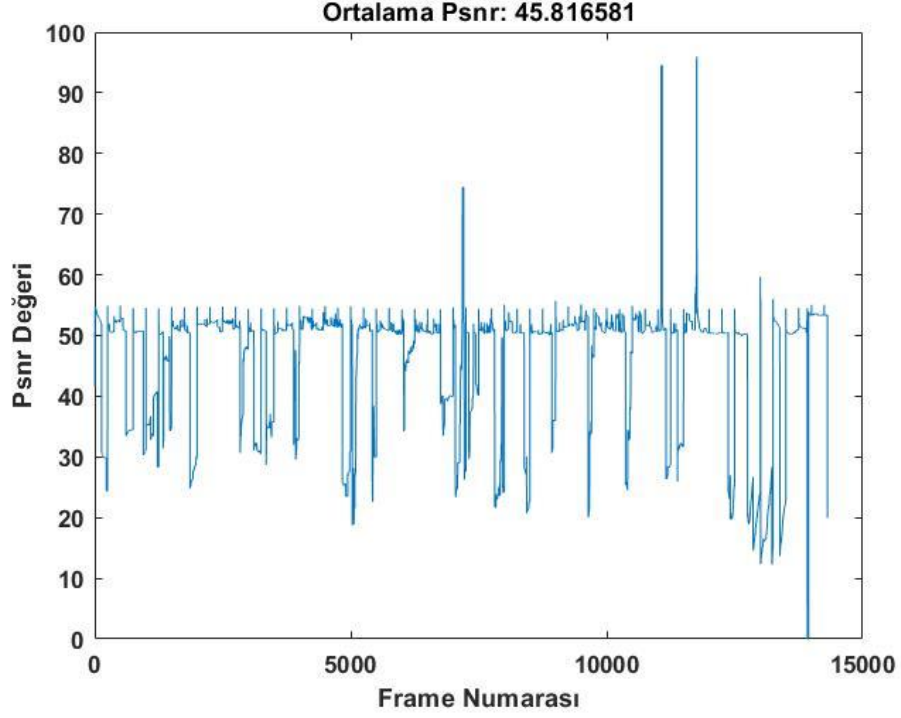
Durum 7 ve durum 8 testleri, kontrolör performansının değerlendirilmesi amacıyla gerçekleştirilmiştir. SD-CDN'e gelen OpenFlow paketlerinin kontrolörümüzü ne kadar yorduğunu incelenerek, kontrolör ölçekleme konusu değerlendirilmiştir. Durum 7 ve durum 8'de hem kullanıcı erişim protokolü hem de arka plan trafiği için UDP tabanlı QUIC protokolü kullanılarak istekler host-1'den host-3'e yönlendirmiştir. Ancak, durum 7'de açıklanan ilk testte video iletim sırasında, host-4 ile host-5, host-6 ile host-7 ve host-8 ile host-9 arasında arka plan trafiği oluşturulmuşken, durum 8'de açıklanan ikinci testte herhangi bir arka plan trafiği yaratılmamıştır. Burada dikkat edilecek husus, Şekil 4.2'den de görüleceği üzere, oluşturulan arka plan trafiğinin, host-1 ile host-3 arasındaki video iletim yolu ile kesişmiyor olmasıdır. Söz konusu arka plan trafiği oluşturulmasındaki amaç sadece kontrolör üzerindeki yükün incelenmesidir.

Durum 7: QUIC protokolü ile host-1 tarafından host-3'e yapılan yedinci istek sonrasında sağlanan videoya karşın ortalama PSNR değeri 14.478212 olarak hesaplanmıştır ve frame'lere ait PSNR grafiği ise Şekil 4.17'de gösterilmiştir.



Şekil 4.17. Durum 7 için PSNR grafiği

Durum 8: QUIC protokolü ile host-1 tarafından host-3'e yapılan sekizinci istek sonrasında sağlanan videoya karşın ortalama PSNR değeri 45.816581 olarak hesaplanmıştır ve frame'lere ait PSNR grafiği ise Şekil 4.18'de gösterilmiştir.



Şekil 4.18. Durum 8 için PSNR grafiği

Durum 7 ve durum 8’de elde edilen ortalama PSNR değerleri kıyaslandığında, yaklaşık 3 katlık bir artış görülmüştür. Bu durum, varsayılan yönlendirme modunda kontrolöre gönderilen çoklu kontrol paketlerinden kaynaklanır. Başlangıçta, tüm OpenFlow anahtarları, OpenFlow tablolarında yalnızca “SENT TO CONTROLLER” kuralına sahiptir. Ardından, denetleyici, orijinal sunucu CDN’i aracılığıyla doğrudan yönlendirmeyi OpenFlow tablolarına gömer. Kontrolöre gönderilen bu kontrol paketleri, UDP tabanlı QUIC arka planı ile aşırı yüklenir. Bu sonuçların ışığında, 9 host, 13 ağ cihazı ve bu ağ cihazlarının arasındaki 16 bağlantıyı içeren topolojimizin kontrolü için, yetki alanını bölecek ve eMBB içeriğinde kabul edilebilir QoS oranı elde edecek bir denetleyiciye daha ihtiyacımız olduğu sonucuna varılmıştır.

5. SONUÇ VE ÖNERİLER

Bu tezde, 5G ağının eMBB içeriği için kullanıcı erişim protokolü türüne duyarlı yeni bir yönlendirme algoritmasına sahip SD-CDN mimarisi önermekteyiz. Artan 5G eMBB trafiğinin gereksinimlerinin karşılanabilmesi için, CDN yönlendirmesinde kullanıcı erişim protokolü türünün dikkate alınması yanı sıra, ağın merkezi olarak izlenmesiyle protokol tabanlı trafik yükü ve heterojenlik gibi ağ dinamiklerinin de değerlendirilmesi gerekmektedir. Günümüzde yaygın olarak kullanılan iOS ve Android tabanlı uygulamalarda erişim protokolü türü, TCP tabanlı HTTP/2.0 ve UDP tabanlı QUIC olarak farklılık göstermektedir.

Önerilen kontrolör, sahip olduğu merkezi görünüm sayesinde, yük ve heterojenlik metriklerine göre anycast yönlendirmesi ile CDN'deki en uygun sunucuyu seçer. Burada toplam eşleşen paket sayısı ve erişim protokollerine göre paketlerin oranı göz önünde bulundurulmaktadır. Gerçek bir test ortamında yapılan performans değerlendirmesi sonucunda elde edilen eMBB içeriğinin QoS'leri göz önüne alındığında, önerilen SD-CDN yönlendirmesi, Dijkstra ile sadece topolojideki atlama sayısını göz önünde bulunduran geleneksel DNS tabanlı yönlendirmeden %60 daha iyi PSNR değeri vermektedir.

Seçilen 1080p video üzerinden gerçekleştirdiğimiz testlerin sonuçlarına göre; önerilen mimari 44,5 PSNR ile %86,47 QoS sağlamakta iken, geleneksel DNS yönlendirmesi ile bu oran %14.06 ile 5G gereksinimlerinde kabul edilemeyecek seviyelerde kalmaktadır. İleri bir çalışma olarak ise; bu çalışmada önerilen yönlendirme algoritması temel alınarak, hem CDN önbellek yöntemi hem de CDN'deki yerel ve/veya merkezi sunucuların içerik yoğunluğu dikkate alınarak bu iki amaç fonksiyonu arasındaki optimizasyona odaklanılması düşünülmektedir.

KAYNAKLAR

- [1] Cerwall, P. Ericsson Mobility Report, Ericsson, Stockholm, Sweden, 2020, 1-35.
- [2] Elayoubi, S. E., Benjemaa, S., Altman, Z., Galindo Serrano, A. M. 5G RAN slicing for verticals : Enablers and challenges. *IEEE Communication Magazine*. 2019, 57(1), 28-34.
- [3] Ji, H., Park, S., Yeo, J., Kim, Y., Lee, J., Shim, B. Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects. *IEEE Wireless Communications*. 2018, 25(3) 124–130.
- [4] Erel-Ozcevik, M., Tekce, F., Kayapinar, F. Analytical delay model of virtual CPE for 5G eMBB services in SD-CDN. 16th International Conference on Network and Service Management (CNSM), 2-4 Kasım, 2020, İzmir.
- [5] Prince, M. A brief primer on anycast. <https://blog.cloudflare.com/a%0225-brief-anycast-primer/> (Erişim Tarihi: 04. 06. 2020).
- [6] Xie, G., Li, Z., Kaafar, M. A., Wu, Q. Access types effect on internet video services and its implications on CDN caching. *IEEE Transactions on Circuits and Systems for Video Technology*. 2018, 28(5), 1183–1196.
- [7] Cha, M., Kwak, H., Rodriguez, P., Ahn, Y. Y., Moon, S. Analyzing the video popularity characteristics of large-scale user generated content systems. *IEEE/ACM Transactions on Networking*. 2009, 17(5), 1357–1370.
- [8] Zhang, G., Liu, W., Hei, X., Cheng, W. Unreeling Xunlei Kankan: understanding hybrid CDN-P2P video-on-demand streaming, *IEEE Transactions on Multimedia*. 2015, 17(2), 229–242.
- [9] Rao, S., Karvonen, S. Experimentation with the YouTube content delivery network (CDN). 2013, DOI: 10.13140/2.1.2730.2409.
- [10] Cui, Y., Li, T., Liu, C., Wang, X., Kuhlewind., M. Innovating transport with QUIC: Design approaches and research challenges. *IEEE Internet Computing*. 2017, 21(2), 72–76.
- [11] Fu, Q., Rutter, B., Li, H., Zhang, P., Hu, C., Pan, T., Huang, Z., Hou, Y. Taming the wild: A scalable anycast-based CDN architecture (T-SAC). *IEEE Journal on Selected Areas in Communications*. 2018, 36(12), 2757–2774.
- [12] Xia, W., Wen, Y., Foh, C. H., Niyato, D., Xie, H. A survey on software-defined networking. *IEEE Communications Surveys & Tutorials*. 2015, 17(1), 27–51.
- [13] Chan, A., Zeng, K., Mohapatra, P., Lee, S. J., Banerjee, S., Metrics for evaluating video streaming quality in lossy IEEE 802.11 wireless networks. *Proceedings IEEE INFOCOM*, 2010.
- [14] Yalman, Y., Ertürk, I. A new color image quality measure based on YUV transformation and PSNR for human vision system, *Turkish Journal of*

Electrical Engineering and Computer Science, 2013, 21(2), 603–612.

- [15] Raman, S., Balakrishnan, H., Srinivasan, M. ITP: An image transport protocol for the internet. *IEEE/ACM Transactions on Networking*. 2002, 10(3), 297–307.

