

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**YAZILIM TANIMLI AĞ SİSTEMLERİNDE EŞ ZAMANLI ÇOKLU KATMANLI  
VIDEO AKIŞLARINDA KALİTE ENİYİLEME**

**YÜKSEK LİSANS TEZİ**

**Yamaç ERGİZ**

**Elektrik ve Elektronik Mühendisliği Anabilim Dalı**

**Tez Danışmanı: Dr. Ali Murat DEMİRTAŞ**

**MAYIS 2021**





## TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Yamaç ERGİZ



## ÖZET

Yüksek Lisans Tezi

### YAZILIM TANIMLI AĞ SİSTEMLERİNDE EŞ ZAMANLI ÇOKLU KATMANLI VIDEO AKIŞLARINDA KALİTE ENİYİLEME

Yamaç ERGİZ

TOBB Ekonomi ve Teknoloji Üniversitesi  
Fen Bilimleri Enstitüsü  
Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Ali Murat DEMİRTAŞ

Tarih: Mayıs 2021

Önümüzdeki yıllarda olası internet kullanım alışkanlıkları üzerine yapılan incelemelerde, internet kullanımında artış olacağını ve bu artışa bağlı olarak daha etkili çözümler üretmek gerekeceğini öngörülmüştür. Son yıllarda önem kazanan yazılım tanımlı ağlar (SDN) tabanlı sistemlerle, paket yönlendirme ve kaynak kullanımı konularında etkili bir merkezi kontrol imkanı sağlanmıştır. Bu çalışmada, SDN sisteminde ölçeklendirilebilir videoları iletmek için jitter ve gecikme kısıtlarını dikkate alan eş zamanlı bant genişliği ve çoklu yol paket yönlendirme eniyileme problemi tanımlanmış ve çözülmüştür. Önerilen çözümün uygulanma maliyeti dolayısıyla SDN denetleyicisinde düşük maliyetle hayata geçirilebilecek sezgisel bir algoritma geliştirilmiştir. Problemin iki farklı yapılandırma ile benzetim ve emülasyonları yapılmıştır. Yapılan nicel değerlendirmeler sonucunda her iki yapılandırmada önerdiğimiz sistem modelinin, kendisiyle karşılaştırmalı yapılan algoritmalarından daha iyi sonuç verdiği görülmüştür. Mininet'te gerçekleştirilen emülasyonlarla, elde edilen kuramsal sonuçların doğrulaması yapılmıştır.

**Anahtar Kelimeler:** Yazılım tanımlı ağlar, Katmanlı video kodlama, Çok yollu yönlendirme, Kaynak ayırımı, MIP, Seyir kalitesi, Gecikme, Jitter



## ABSTRACT

Master of Science

JOINT MULTIPATH FLOW AND LAYER ALLOCATION FOR SCALABLE

VIDEO STREAMING

Yamaç ERGİZ

TOBB University of Economics and Technology  
Institute of Natural and Applied Sciences  
Electric and Electronic Engineering Science Programme

Supervisor: Dr. Ali Murat DEMİRTAŞ

Date: May 2021

Projections estimating future online traffic reveal that better techniques are required to meet the expected demand. An emerging network paradigm, software defined networking (SDN) provides effective central management in routing and resource management. In this article, we define and solve a jointly optimal bandwidth allocation and multipath routing problem for SDN based delay/jitter constrained transmission of scalable videos. We also develop a heuristic approach that can easily be implemented in an SDN controller. Two different configurations of the problem are simulated. Numerical evaluations show that proposed multipath routing solutions provide considerable improvement in both configurations when compared to benchmark algorithms. Mininet emulation results also agree with the theoretical results.

**Keywords:** Software defined networking, Scalable video coding, Multipath routing, Resource allocation, MIP, Quality of experience, Delay, Jitter





## TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren Deęerli Hocalarım Ali Murat **DEMİRTAŐ**'a, ve Tolga **GİRİCİ**'ye kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Elektrik ve Elektronik Mühendislięi Bölümü öğretim üyelerine ve destekleriyle her zaman yanımda olan aileme çok teşekkür ederim.



## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET</b> .....	iv
<b>ABSTRACT</b> .....	v
<b>TEŞEKKÜR</b> .....	vi
<b>İÇİNDEKİLER</b> .....	vii
<b>ŞEKİL LİSTESİ</b> .....	viii
<b>ÇİZELGE LİSTESİ</b> .....	ix
<b>KISALTMALAR</b> .....	x
<b>SEMBOL LİSTESİ</b> .....	xi
<b>1. GİRİŞ</b> .....	<b>1</b>
1.1 Motivasyon .....	1
1.2 Tezin Katkısı .....	2
1.3 Tezin Organizasyonu .....	3
1.4 Literatür Araştırması .....	3
1.5 Gelişmiş ve Ölçeklendirilmiş Video Kodlama .....	5
1.5.1 Gelişmiş video kodlama .....	5
1.5.1.1 Tahminleme işlemleri .....	6
1.5.1.2 Dönüştürme ve Niceleme .....	6
1.5.1.3 Değişken uzunluklu kodlama .....	6
1.5.2 Ölçeklendirilmiş video kodlama .....	7
1.5.2.1 Zamanda ölçeklendirme .....	7
1.5.2.2 Uzamsal ölçeklendirme .....	8
1.5.2.3 Kalite ölçeklendirmesi .....	10
<b>2. SİSTEM MODELİ</b> .....	<b>11</b>
2.1 Kuramsal Model .....	12
2.2 Sezgisel Algoritma .....	16
<b>3. BENZETİM DEĞİŞKENLERİ SONUÇLARI</b> .....	<b>19</b>
3.1 Simülasyonda Kullanılan Ölçeklendirilebilir Video Yapılandırılması .....	19
3.2 Ağ Benzetim Değişkenleri .....	21
3.3 Benzetim Sonuçları .....	22
3.3.1 Kıyaslama algoritması 1 tek yönlendirme .....	23
3.3.2 Kıyaslama algoritması 2 sabit yol yönlendirme .....	27
3.4 Yeniden Yapılandırılmış (Adil Olmayan) Sistem Modeli .....	28
<b>4. MİNİNET VE POX EMÜLASYON SONUÇLARI</b> .....	<b>35</b>
4.1 Gerçek zamanlı uygulanması .....	35
4.2 Emülasyon Sonuçları .....	37
<b>5. SONUÇ VE ÖNERİLER</b> .....	<b>39</b>
<b>KAYNAKLAR</b> .....	<b>41</b>
<b>ÖZGEÇMİŞ</b> .....	<b>45</b>



## ŞEKİL LİSTESİ

### Sayfa

Şekil 1.1 : Zamanda ölçeklendirme.....	8
Şekil 1.2 : Uzamsal ölçeklendirme.....	9
Şekil 1.3 : Uzamsal ölçeklendirmede kodlama .....	9
Şekil 1.4 : Kalite ölçeklendirmesinde kodlama.....	10
Şekil 2.1 : Video kaynağını,hedefini, ağ anahtarlarını ve katmanlı kodlamayı gösteren sistem modeli .....	11
Şekil 2.2: Çok Yollu Çok katmanlı Video Aktarımı Algoritması.....	18
Şekil 3.1: Benzetimlerde kullanılan ağ topolojisi .....	22
Şekil 3.2: Tek yollu çok katmanlı video aktarımı algoritması .....	26
Şekil 3.3 : Performans karşılaştırması (Yapılandırma 1) .....	28
Şekil 3.4 : Performans karşılaştırması (Yapılandırma 2) .....	29
Şekil 3.5 : Uçtan uca gecikmenin performans üzerindeki etkisi (Yapılandırma 1) ...	30
Şekil 3.6 : Uçtan uca gecikmenin performans üzerindeki etkisi (Yapılandırma 2) ...	31
Şekil 3.7 : Yolları kısıtlanmış ağda uçtan uca gecikmenin performans üzerindeki etkisi .....	32
Şekil 3.8: Uçtan uca gecikmenin yapılandırma 1 için performans karşılaştırmasının SSIM cinsinden gösterimi .....	33
Şekil 4.1: Gecikmeyi ölçmek için kullanılan Mininet senaryosu.....	36
Şekil 4.2: Mininet emülasyon sonuçları (Yapılandırma 1) .....	37
Şekil 4.3: Mininet emülasyon sonuçları (Yapılandırma 2) .....	38



## ÇİZELGE LİSTESİ

	<b><u>Sayfa</u></b>
Çizelge 3.1: Videoları aktarmak için gereken bant genişlikleri.....	20
Çizelge 3.2 : Ağ benzetim değişkenleri.....	21
Çizelge 3.3 : Her bir video katmanının sağladığı bireysel kalite.....	23
Çizelge 3.4 : Her bir video katmanının sağladığı bireysel kalite SSIM cinsinden...	23
Çizelge 3.5 : Bireysel katman kalitelerinin göndermek için gereken veri hızına oranı.....	25







## KISALTMALAR

<b>BFS</b>	: Sığ Öncelikli Arama (Breadth First Search )
<b>MILP</b>	: Tamsayı Karışık Doğrusal Programlama (Mixed Integer Linear Programming)
<b>MSE</b>	: Ortalama Kare Hatası (Mean Square Error)
<b>PSNR</b>	: En Yüksek Sinyal Gürültü Oranı (Peak Signal to Noise Ratio )
<b>QoE</b>	: Deneyim Kalitesi (Quality of Experience)
<b>QP</b>	: Niceleme Değişkeni (Quantization Parameter)
<b>SDN</b>	: Yazılım tanımlı Ağlar (Software Defined Network)
<b>SSIM</b>	: Yapısal Benzerlik Gösterge Ölçevi (Structural Similarity Index Measure)
<b>SVC</b>	: Ölçeklendirilebilir Video Kodlaması (Scalable Video Coding)



## SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
$k$	Akış Dizini
$l$	Katman Dizini
$i$	Anahtar dizini
$j$	Anahtar dizini
$x^{k,l}$	k akışının l katmanının aktarılıp aktarılmadığı
$Q^{k,l}$	k akışının l katmanının sağladığı kalite
$R^{k,l}$	k akışının l katmanını göndermek için gereken bant genişliği
$y_n^{kl}$	k akışının l katmanının n yolundan aktarılıp aktarılmadığı
$f_n^{kl}$	k akışının l katmanının n yolundan aktarılan miktarı
$r_{ijn}$	n yolunun i-j bağlantısını içerip içermediği
$f_{ij}^0$	Ağda i-j bağlantıları üzerinden geçen çapraz trafik miktarı
$B_{ij}$	i-j bağlantısının bant genişliği
$d_{ij}$	Sistemde izin verilen en yüksek bağlantı gecikmesi
$D_{max}^{link}$	Sistemde izin verilen en yüksek gecikme
$D_{max}^{eze}$	Sistemde izin verilen en yüksek uçtan uca gecikme
$a_{ij}$	i-j arasında bağlantı olup olmadığı

$J_{max}$	Sistemde izin verilen en yüksek jitter
$J_{p,p'} \in \{0,1\}$	Eğer p ve p' videonun aynı katmanı için bölüştürülürse 1 değeri alır.
$B_{i,j}^R$	Bağlantı i-j'deki kalan bant genişliği
$R_{k,l}^R$	k videosunun l katmanının iletilmemiş kısmı
$R_p$	P yolunun kalan kapasitesi
$p_{k,p}^{used}$	Yol p, k videosu için kullanılıyorsa 1 değerini alır.
$l_k \in L$	k videosu için iletmeyi bekleyen katman



## 1. GİRİŞ

Dünya Bankası verilerine göre dünya çapında internet kullanan bireylerin sayısı, 1993 yılından beri düzenli bir artış göstermektedir. Örneğin 2016 yılında internet kullanım oranı, önceki yıla oranla yüzde 2,8 artarak yüzde 45,8'e ulaşmıştır [1]. Yapılan tahminler, yıllık küresel İnternet trafiğinin 2022 yılı itibariyle 396 exabyte'a ulaşacağını öngörmektedir. Aynı şekilde, tahminler 2022 yılı itibariyle küresel video trafiğinin tüm İnternet trafiğinin yüzde 75'ine ve canlı çevrimiçi video trafiğinin tüm İnternet trafiğinin yüzde 17'sine tekabül edeceğini öngörmektedir [2]. Çevrimiçi video trafik miktarında beklenen bu artış, bize gelecekte daha etkili dolaşım çözümlerine ihtiyaç duyulacağını göstermektedir.

### 1.1 Motivasyon

Ölçeklendirilebilir kodlanmış videolar (SVC), H.264/AVC standardına eklenti olarak standartlaştırılmış bir tasarımdır. SVC; kaynak videoyu uzamsal, zamanda ve kalite ölçekleme ile kodlayarak bir videoyu temel katman ve yardımcı katmanlar olarak bit akışlarına dönüştürmektedir. Bu katmanlar videonun aktarılmasında elde edilen kaliteye bağlı olarak değişen miktarda öneme sahiptir. Çeşitli ağ araçları kullanılarak bir ağın karakteristik özellikleri gözlemlenebilir ve Ölçeklendirilmiş Video Kodlama aktarım esnasındaki koşulları da hesaba katarak Seyir Kalitesini (QoE) arttırmak için video uygun bir biçimde kodlanabilir [3].Yazılım tanımlı Ağlar (SDN) temelde bir denetleyici ve uygulama programlama arayüzünden oluşan merkezi ve programlanabilir bir ağ mimarisidir [4]. Openflow bir SDN protokolü olup Denetleyici adı verilen bir yapı aracılığıyla anahtarların iletme katmanı üzerinde değişiklikler yaparak ağı dinamik olarak kontrol edebilmektedir. Openflow protokolü kullanılarak her bir video katmanı farklı koşullarla gönderilerek ölçeklendirilebilir kodlanmış bir videonun seyir kalitesini eniyilenebilir.

## 1.2 Tezin Katkısı

Bu tezde, eniyileme tabanlı ve SDN ile birlikte kullanılabilen bir ağ kontrolü ve veri aktarımı problemi çalışıldı ve çoklu yol seçimi yaparken eş zamanlı olarak kaynak ayırımı ve kaliteyi eniyileyebilecek bir problem tanımlandı. Çalışmada, eniyilemeye dayalı ve ayrıca SDN ile uygulanabilir bir sezgisel yaklaşım temel alındı. Önerilen sistem, video trafiğinin QoE (gecikme ve jitter) gereksinimlerini karşılamak için eşzamanlı eniyilendirilmiş çok yollu yönlendirme ve bant genişliği tahsisi yapılabilecek kapasitededir.

Problem en iyi sonucu verecek yolları çözerken gecikme ve jitter gereksinimlerini içine alan QoE gereksinimlerini sağlamaya çalışır. Modelleme için gerekli parametreleri elde etmek için; ölçeklendirilebilir kodlanmış videolardan deneysel yolla elde ettiğimiz bir veri hızı kalite oranı kullanılarak her bir videonun her bir katmanına bir değer biçildikten sonra bu oran, kullanılan bant genişliği ve elde edilen kaliteyi dengeleme imkânı sunar. Böylece tanımlanan eniyileme problemi çözülerek en iyi kalite imkânı sağlayacak yol seçilebilir. Bunu yapabilmek için Tam Sayı Doğrusal Programlama (MILP) türü bir eniyileme problemi tasarlandı. Bunun yanı sıra tanımlanan modelle karşılaştırma yapabilmek için iki sezgisel yaklaşımı sahip algoritma tasarlandı.

Bunlara ek olarak önerilen MILP algoritmasının maliyeti, gerçek zamanlı kullanılmayacak kadar yüksek olduğu için kaliteden kısmen ödün vererek işlem maliyetini düşürecek Ford Fulkerson algoritması tabanlı sezgisel bir algoritma tasarlandı. Literatürdeki bu alandaki çalışmalardan farklı olarak bu tezde, uçtan uca gecikme ve jitter kısıtları dikkate alındı. Ayrıca literatürdeki çalışmaların çoğunda katmanları ya da bütün videoyu aktarmak için tek yol iletim kullanılırken bu çalışmada çoklu yollu iletim tercih edilmiştir [5][6][7][8][9][10].

### 1.3 Tezin Organizasyonu

Tez şu şekilde düzenlenmiştir: Bölüm 1.4'te video akışı üzerine SDN tabanlı bazı önemli çalışmalar özetlenmiştir. Bölüm 1.5'te bu çalışmada kullanılan gelişmiş video kodlama ile ilgili bilgiler verilmiştir. Bölüm 2'de en iyi şekilde hangi katmanların iletileceğini ve iletilecek katmanların hangi yolları izleyeceği seçmek için bir MILP modeli, tanımlanmıştır. Bölüm 2.2'de eniyileme algoritmasının yüksek maliyeti dolayısıyla, daha az maliyetli ve eniyileme algoritmasının yerine kullanılabilir sezgisel bir yaklaşım geliştirilmiştir. Bölüm 3'te, tek yol yönlendirme algoritması tabanlı bir kıyaslama algoritması tanıtılmış ve performans karşılaştırması için farklı algoritma yapılandırmalarıyla sayısal değerlendirmeler yapılmıştır. Bölüm 4'te Mininet ile emülasyonlar yapılarak benzetim sonuçlarının sağlanması yapılmıştır. Son bölümde ise sonuçlar verilmiştir.

### 1.4 Literatür Araştırması

Video trafiğinin çok yönlü aktarımı [5], [6], [7] ile verilen çalışmalarda olduğu gibi kapsamlı bir şekilde incelenmiştir. Ancak bu çalışmalar kablosuz ağlara ve enerji verimliliğine odaklanmaktadır ve çalışmalarda, ölçeklendirilebilir video kodlamasına, yazılım tanımlı ağlara değinilmemekte ve yönlendirme problemi çözülmemektedir. Çalışma [8], ağ üzerine yeni bir inceleme olup incelemede SDN tabanlı video aktarımına dair herhangi bir çalışmadan bahsedilmemektedir. Çalışmalar [9] ve [10]'da SDN tabanlı çok yönlü TCP tabanlı video akışını ele alınmakla birlikte ölçeklendirilebilir video kodlamalarından bahsedilmemektedir. Ayrıca birden çok alternatif yol arasında akış optimizasyonu, jitter ve uçtan uca gecikme sorunları da dikkate alınmamaktadır.

Literatürde, yazılım tanımlı ağlarda ölçeklendirilebilir video aktarımı üzerine çalışan dikkate değer çalışmalar vardır. Çalışma [11]'de müşteriler ölçülen ağ koşullarının kısıtlaması altında uygun video codec bileşenini seçmeye yönlendirilir. Çalışma [12]'de katmanların sayısını dinamik olarak ayarlamak için bir algoritma ve iletim yollarını aynı anda optimize etme olanağı sunulmaktadır. Problem 0/1 Knapsack modeli olarak belirtilmiş ve Yapay Balık Sürüsü ile çözülmüştür. Ancak, uçtan uca gecikme ve jitteri ele alınmamaktadır. Çalışma [13]'te sunucu, codec ve video katmanlarının sayısına SDN denetleyicisi tarafından karar verilen ölçeklendirilebilir



bir video akışı mimarisi, tasarlanmıştır. Çalışma [14]'te, kalite modeline göre bir video hızı uyarlama yöntemi önerilmiştir. Oluşturulan hız-kalite modelinde, belirli bir hızın altındaki maksimum kalite farklı akışlardaki video kalitelerinin ağırlıklı toplamı ile ilişkilendirilmiştir. Ancak, yazarlar yönlendirmeye, uçtan-uca gecikme ve jitter kısıtlamalarına değinmemiştir. Yazarlar tarafından [15] 'te SDN tabanlı adaletli dağıtımı esas alan bir video hızı uyarlaması algoritmasının çalışma performansı analiz edilmektedir. Çalışma [16] 'da QoE ölçüsünü maksimize etmek içinde Karma Tamsayı Programlama tabanlı optimizasyonun bulunduğu bir sistem formüle edilmiştir. Çalışma [17] 'de iletilecek video katmanlarının sayısına ve aktarım yoluna karar verme problemi tanımlanmış ve seçilen her bir video katmanının yönlendirilmesi MDP olarak modellendikten sonra problem, Q Learning ile çözülmüştür. Çalışma [18]'de, dinamik bir QoS yönlendirme optimizasyonu tanımlanmış ve bu problemin çözümünde katmanlara birbirlerinden farklı öncelik seviyeleri atanmıştır. Çalışma [19] 'da, SDN kullanan multimedya uygulamaları için tek yöllü yönlendirme incelenmiştir. Çalışmalar [20] ve [21] 'de, yazarlar tarafından multimedya hizmetleri için bir QoE yönetimi yaklaşımı önerilmektedir, ancak bu çalışmalarda ölçeklendirilebilir video akışı veya uçtan uca gecikme-jitteri dikkate alınmamaktadır. Çalışma [22]'de, ölçeklendirilebilir bir video iletimi için SDN tabanlı tek yöllü bir yönlendirme çözümü önerilmektedir. Çalışma [23]'te yazarlar tarafından, küçük hücrelerden oluşan bir radyo erişim ağı değerlendirilmektedir, video katmanı seçimi ve segment tabanlı yönlendirme problemi ele alınmaktadır. Bununla birlikte yazarlar tarafından uçtan uca gecikme ve jitter ele alınmamış, algoritmalar SDN kullanarak emüle edilmemiştir. Tüm bu çalışmalarda tek yöllü yönlendirme, bir katmanı veya videonun tamamını iletmek için kullanılmıştır. Ayrıca bu çalışmalarda gecikme ya dahil edilmemiş ya da gecikme kısıtlamasını gerçek uçtan uca gecikmeyi yansıtmayan bir şekilde basitleştirilmiştir.

Çalışma [24]'te, SDN tabanlı akış kontrolü ve yönlendirme üzerine çok yeni ve kapsamlı bir ankettir. Ancak bu ankette bahsedilen çalışmaların hiçbiri ölçeklendirilebilir kodlanmış videonun çok yöllü yönlendirmesini ele almamaktadır. Çalışma [25], kablosuz çok yöllü video akışı hakkında yakın zamanda yapılmış bir incelemedir ve kablosuz ağ içeren çalışmalara odaklanmıştır. İncelemede bu alandaki çalışmalar özetlenmiştir ve çalışmalar, kullanılan katmanlı ağ mimarisi standartlarına göre sınıflandırıldıktan sonra bu standartlar altında ayrıntılı incelenmiştir.

İncelemede, ölçeklendirilebilir video kodlaması ile çok yönlü yönlendirme üzerine herhangi bir çalışmaya atıf yapılmamıştır. Ayrıca, SDN'nin çok yönlü video yönlendirmesi ile entegrasyonu açık bir araştırma sorunu olarak listelenmemiştir.

## **1.5 Gelişmiş ve Ölçeklendirilmiş Video Kodlama**

### **1.5.1 Gelişmiş video kodlama**

H.264 AVC (Advanced Video Coding) video kodlama için yaygınlıkla kullanılan bir video kodlama standardıdır. Bu kodlama metoduyla yapılan sıkıştırılmalar videoyu iletirken ya da diske kaydederken videonun daha az yer kaplamasını sağlamaktadır. Kodlayıcı (Encoder) vasıtasıyla video sıkıştırılmakta ve çözücü (Decoder) sayesinde eski haline getirilmektedir. Bu araçların standartlaştırılması, farklı üreticiler tarafından geliştirilen video oynatma kapasitesine sahip cihazların bu standartla kodlanmış videolarla uyumlu olmasını sağlamıştır.

H.264 video kodlayıcısı sıkıştırılmış bir H.264 bit akışı yaratmak için sırasıyla tahminleme, dönüştürücü, nicemleme ve değişken uzunluklu kodlama işlemlerini gerçekleştirmektedir. Çözücü tarafından bu işlemlerin tersi yapılarak bir video akışı oluşturulmaktadır. Çoğunlukla çözücü tarafından oluşturulan video ile orijinal video, sıkıştırma işleminden doğan kayıplardan dolayı aynı değildir. Sıkıştırılan video her ne kadar az yer kaplasa da bit akışında meydana gelecek en ufak bir hata videoda büyük bir bozulma meydana getirir.

H.264 standardında video kareleri genellikle (16x16) makrobloklar şeklinde işlenir. Öncelikle, kodlayıcıda bir tahminleme bloku yaratılır ve eldeki bloktan çıkartılarak bir artık bloku elde edilir. Bu blok dönüştürülmekte, nicemlenmektedir ve kodlanmaktadır. Bu işleme aynı anda nicemlenmiş blok boyutları yeniden ölçeklendirildikten sonra ters dönüşümden geçirilerek artık blokuyla birleştirilir ve daha sonraki kodlama işlemlerinde kullanılır. Tahminleme işlemi karenin kendi içinde yapılırsa buna içsel tahminleme (intra coding), diğer karelerle yapılırsa buna kareler arası tahminleme (inter prediction) denir.

Çözücü kodlayıcıda oluşturulan artık bloğun aynısını oluşturur ve çözülmüş bir makroblok oluşturmak için artık makrobloklara ekler. Kodlama işlemi sırasında aşağıdaki işlemler gerçekleşir.

### **1.5.1.1 Tahminleme işlemleri**

Kodlayıcı daha önce kodlanan bloklardan faydalanarak şu anki makroblokun bir tahminini oluşturur. Bu tahminlemeyi yapmak için iki çeşit yol vardır. İnter tahminlemede karenin kendi içinde, inter tahminlemede ise daha önce kodlanmış kareler kullanılır. Intra tahminlemede daha önce kodlanan komşu pikseller kullanılarak şu anki blokun tahminlemesi oluşturulur. Gerçek bloktan, bu oluşturulan blok çıkarılarak artık blok oluşturulur. Inter tahminlemede ise görüntüleme sırasında kendinden önce yada sonra gelebilecek daha önce kodlanmış bloklar kullanılır. İnter tahminlemede P ve B kareleri olmak üzere iki çeşit kare vardır. P-kare geçmiş karelere motion compensation işlemi uygulayarak gelecek kareleri tahmin eder. B-karelerinde ise iki yönlü tahminleme vardır. B-kare P-karelere oranla daha fazla sıkıştırma sağlar.

### **1.5.1.2 Dönüştürme ve Niceme**

Tahminleme işlemi ile elde edilen artık blok Ayrık Kosinüs Dönüşümü ile dönüştürülür. Dönüşüm ile artık blokunu yeniden yaratmak için kullanılacak dönüşüm katsayıları elde edilir. Bu dönüşüm katsayıları bloku, verilen bir niceme parametresine göre nicemlenir. Niceme parametresinin büyük tutulması, daha fazla dönüşüm katsayısının sıfırlanmasına yol açar ve video kalitesinin düşmesine sebep olur. Buna karşılık videonun diskte kapladığı alan küçülür. Aynı şekilde parametrenin düşük seçilmesi, çözülen videonun kalitesini artırır ve videonun diskte kapladığı alan büyür.

### **1.5.1.3 Değişken uzunluklu kodlama**

Bu aşamada, daha önceki işlemlerde elde edilen nicemlenmiş katsayılar ve çözücünün videoyu yeniden oluştururken ihtiyaç duyacağı bilgiler sıkıştırılarak kullanılır. Girdiler, bu işlemde değişik uzunluktaki anahtar kelimelere kodlanır. Sıklıkla kullanılan kodlar, daha kısa anahtar kelimeler şeklinde kodlanırken; daha az sıklıkla kullanılan kodlar, daha uzun anahtar kelimeler şeklinde kodlanır.

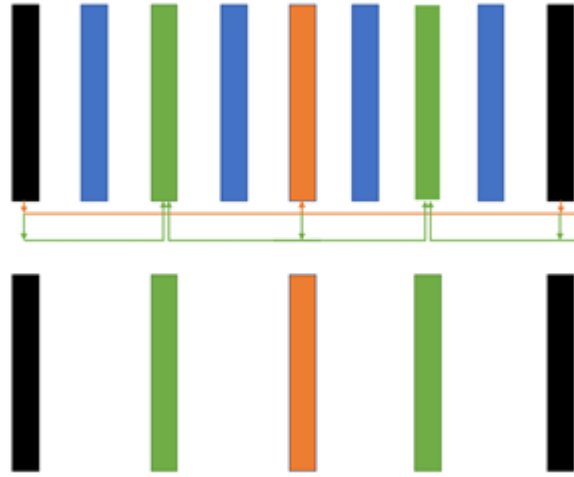
## 1.5.2 Ölçeklendirilmiş video kodlama

Birçok video sıkıştırma algoritması için aynı video serisinin farklı versiyonlarının (farklı kalitelerde vb.) iletilmesi büyük bir engeldi. Bu işlem aynı video akışının farklı parametrelerle elde edilmiş versiyonlarının arasındaki artıklar değerlendirilerek elde edilebilir. Bu katmanlardan ilki temel katman olarak adlandırılır. Temel katmandan tek katmanlı bir çözücü kullanılarak mümkün olan en düşük kalite/çözünürlükte video elde edilebilir. Daha yüksek kalitede bir video elde etmek için bir ya da daha fazla geliştirme katmanının temel katmanla beraber çözülmesi gerekir. Temel katman olmadan geliştirme katmanları bir işe yaramaz. SVC ardışık geliştirme katmanlarının tahminlemesini temel katmanı ve kendinden daha önce kodlanmış katmanları kullanarak gerçekleştirir.

SVC'nin temel konsepti, video bit akışının hepsini kullanmadan da anlamlı bir video oluşturabilmesidir. Temel katmanın yanında çözülen geliştirme katmanlarının sayısı video kalitesini belirlemektedir. SVC'de videolar 3 şekilde kodlanmaktadır.

### 1.5.2.1 Zamanda ölçeklendirme

Temel katman, düşük zamansal çözünürlükte kodlanmaktadır. Geliştirme katmanları, çözülen video serisinin kare hızını arttırmaktadır. Öncelikle temel katmanı elde etmek için video  $F_0$  fps ile kodlanır. Örneğin temel katman; 0, 2,4, 6 vb. sıralı karelerden oluşabilir ve tek başına anlamlı bir şekilde çözülebilir. Katman 1 ise 1,3,5,7 vb. karelerden oluşabilir. Çözücüde temel katman, katman 1 ile birleştirilerek çözümlenerek  $2 * F_0$  fps sahip bir video elde edilebilir.

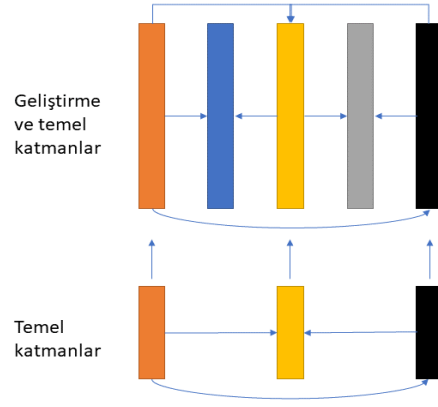


Şekil 1.1 : Zamanda ölçeklendirme

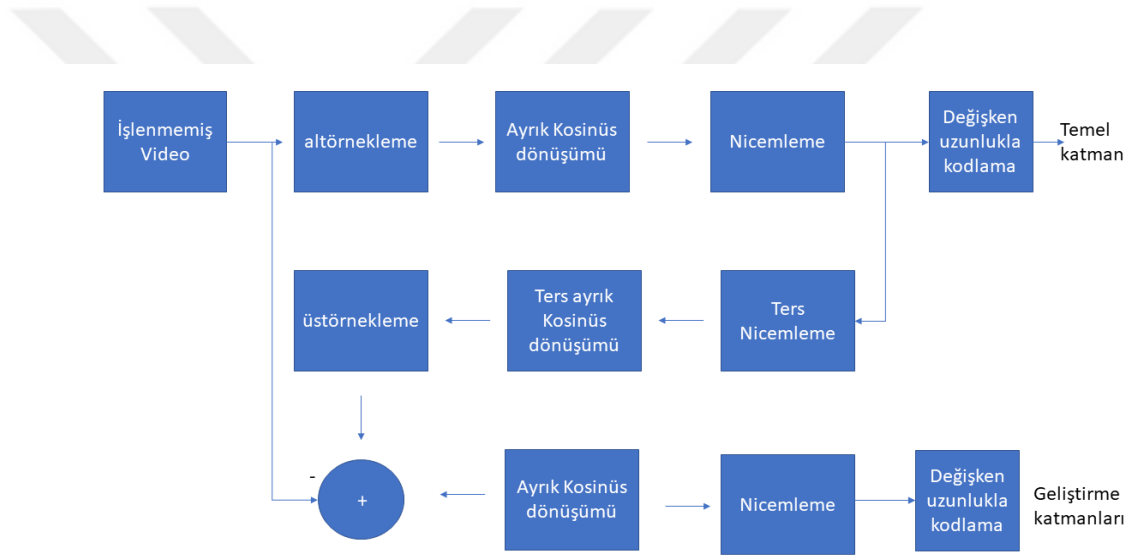
### 1.5.2.2 Uzamsal ölçeklendirme

Temel katmanlar düşük uzamsal çözünürlüklerle kodlanır. Geliştirme katmanları çözülen video serisinin uzamsal çözünürlüğünü arttırmaktadır.

Uzamsal ölçeklendirmede, öncelikle video karelerinden alt örnekleme yoluyla videonun düşük çözünürlüklü versiyonları elde edilir. Bu yolla temel katman elde edilmektedir. Kodlayıcı, bu düşük çözünürlüklü kareyi yeniden inşa eder ve üst örnekleme yoluyla orijinal kareyle aynı boyuta ölçeklendirir. Bu kare, tahminleme işlemi için referans olarak alınır ve geliştirme katmanlarının kodlanmasına imkan sağlar. Üstörneklenmiş kare altörnekleme, kodlama ve üstörnekleme işlemlerinden kaynaklı bazı bozulmalara rağmen iyi bir referanstır. Çözücüde ise kodlayıcıda elde edilen düşük çözünürlüklü temel katman üst-örnekleyerek ilk geliştirme katmanı elde edilir. Daha sonra ardışık olarak geliştirme katmanları, temel katmana eklenerek bir sonraki geliştirme katmanı elde edilir. En yüksek çözünürlüklü katman, orijinal videoyla aynı çözünürlüğe sahiptir. Çözücüde kullanılan katman sayısı azaldıkça, elde edilen videonun çözünürlüğü de azalır.



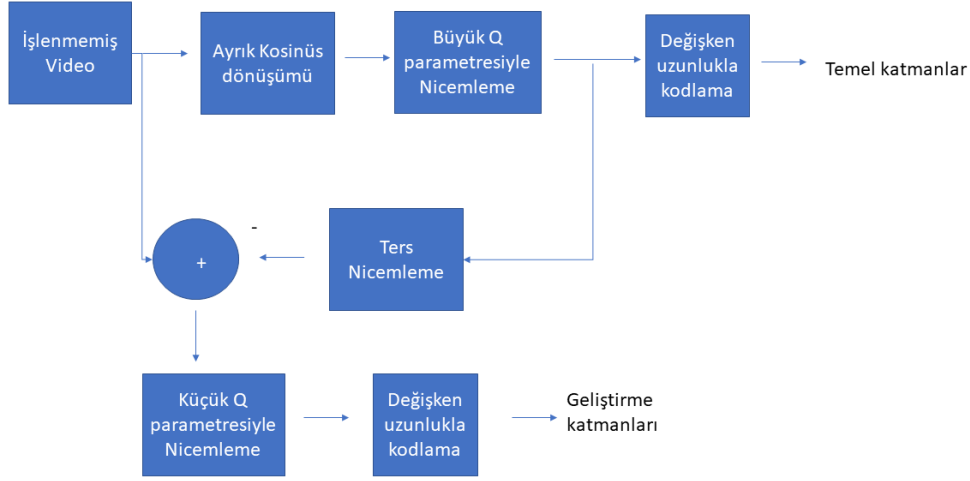
Şekil 1.2 : Uzamsal ölçeklendirme



Şekil 1.3 : Uzamsal ölçeklendirmede kodlama

### 1.5.2.3 Kalite ölçeklendirmesi

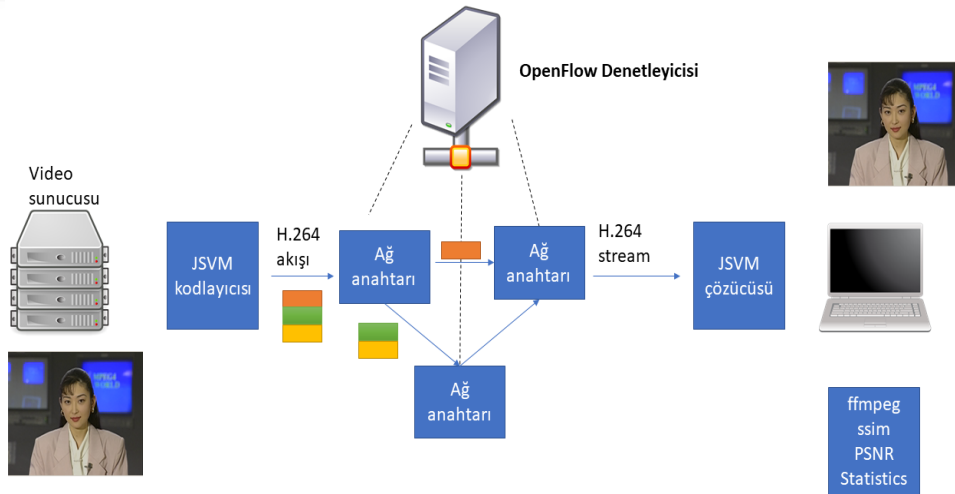
Temel katman, yüksek nicemleme parametresi kullanılarak kodlanır. Geliştirme katmanları, çözülen serinin görsel kalitesini artırır. Kalite ölçeklendirmesinde, belirli bir nicemleme parametresi kullanarak temel katman bit akışı elde edilir. Bu bit akışı yeniden inşa edilerek ve daha düşük bir nicemleme paritesiyle yeniden kodlanarak daha yüksek kalitede bir video elde edilir. Bu kare, geliştirme katmanlarını oluşturmada tahminleme işlemi için kullanılır. Bu aşamalar, daha düşük bir nicemleme paritesiyle tekrar edilerek diğer geliştirme katmanları elde edilir [26].



Şekil 1.4 : Kalite ölçeklendirmesinde kodlama

## 2. SİSTEM MODELİ

Bu çalışmada, bir SDN denetleyicisi ve K tane eş zamanlı video iletimi içeren bir kablolu ağ topolojisi varsayıldı. Her bir akış bir kalite tabanlı (QP) ölçeklendirilebilir video akışı olarak kodlandı. Bu sayede, video akışının bazı bölümleri kanal koşullarına bağlı olarak iletilebilir. Akışlar varsayılan olarak aynı kaynak düğümünden başlar ve aynı havuz düğümüne yönlendirilir. Her akış, aynı sayıda L katmanıyla kodlanmıştır. Ağ bağlantıları sabit bir bant genişliğine sahiptir. Sürekli gecikme ve jitterın da bağlantılarda gerçekleştiği varsayılır. Önerilen çözümlerde, denetleyicinin kanal koşullarını ölçebileceği ve tüm paketlerin dinamik olarak yönlendirebildiği kabul edildi. Ayrıca yönlendirici tarafından, paketler önceden tanımlanan kurallara bağlı olarak işlenebilir veya ölçüm gecikmesi olmadan paketler aynı anda yönlendirilebilir. Şekil 2.1’de sistem modeli gösterilmiştir. Ardından çok yönlü eniyileme problemi işlenecektir.



Şekil 2.1 : Video kaynağını, hedefini, ağ anahtarlarını ve katmanlı kodlamayı gösteren sistem modeli



## 2.1 Kuramsal Model

Kaliteyi en üst düzeye çıkararak eniyilendirilmiş yönlendirme mekanizmasını belirlemek için gönderilecek ölçeklendirilebilir video katmanlarının yönlendirilmesini, iletimi kalite ile bağdaştıran bir kalite eniyileme problemi tanımlanması gerekmektedir. İkili değişken  $x^{k,l}$  eğer  $k$  video akışının  $l$  katmanı denetleyici tarafından gönderilirse 1 değerini almaktadır. Eşitlik (1) 'de elde edilen toplam kalite, iletilen ardışık her bir katmanın iletimiyle elde edilen bireysel kalitelerinin toplamı olarak ifade edilmiştir. Başka bir deyişle (1) ağ akışı sistemimizde en yüksek ödül fonksiyonudur, ikili değişken  $x^{k,l}$  ağdaki video iletiminin toplam ödülünün en yüksek değeri için çözülen eniyileme değişkenidir ve ağdan kaynaklanan fizibilite gereksinimlerini karşılamaktadır. Toplam ödülün varsayılan değeri sıfırdır. En yüksek toplam ödül, tüm videoların tüm katmanlarının başarılı bir şekilde hedef düğüme iletilip maksimum kalite elde edildiği duruma karşılık gelir. Eşitlik (2)'gösterilen  $Q_{k,l}$ ,  $k$  videosunun  $l$  katmanının çözülmesinden sağlanan diferansiyel kalite ölçüsünü simgeler.

$$\max_{f, \bar{x}, \bar{y}} = \sum_{k=1}^K \sum_{l=1}^L x^{k,l} \times Q_{k,l} \quad (1)$$

$$Q^{k,l} = \text{Quality}(\sum_{m=1}^l R_{k,m}) - \text{Quality}(\sum_{m=1}^{l-1} R_{k,m}) \quad (2)$$

Burada  $R_{k,m}$ ,  $k$ . videonun  $m$  katmanını göndermek için gereken hızı temsil eder ve Quality fonksiyonunu, aşağıda benzetim sonuçlarında da görüleceği gibi belirli bir videoya tahsis edilen belirli bir veri kapasitesi için kaliteyi gösteren değerdir.

Çalışmada, uçtan uca tanımlanmış bir yollar kümesi olduğu varsayılmıştır ve bu yollar  $P$  sembolüyle ifade edilir. Bu yollar keşif edilebilirler. Bu yollar, ayrıca döngü içermez. Kaynaktan hedefe giden yolları keşfetmek için Kapsamlı İlk Arama (BFS) kullanılmıştır. Bir videonun bir katmanı iletilecekse, alternatiflerden bir dizi yol seçilir. Burada  $y_p^{kl}$ , eğer  $P$  yolu  $k$ 'inci videonun  $l$  katmanı kullanılacaksa 1, aksi

takdirde 0 değerini alan ikili bir karar değişkeni olarak tanımlanmıştır.  $f_p^{k,l}$ , sürekli bir karar değişkeni olabilir ve bu da  $k$  videosunun  $l$  katmanının akışına ait  $p$  yolunu kullanarak iletilen akış miktarını gösterir.  $r_{i,j,p}$ , eğer yol  $p(i,j)$  bağlantısını içeriyorsa 1 değerini alan bir ikili değişkendir. Bu parametrenin aldığı değerler BFS ile belirlenmiştir.

Kısıtlamalar aşağıdaki gibi tanımlanır. Bir düğüme giden toplam akış, düğümden çıkan akış miktarına eşit olmalıdır. Akış koruma yasası kısıtlaması (3)'te tanımlanmış olup her bir videonun katmanlarına ( $k, l$  sırasıyla video ve katman indeksleri) ait akışların toplamının sistemde sabit kalacağı varsayılmıştır ve gerçek hayatta karşılaşılabilecek paket kaybı gibi olaylar göz ardı edilmiştir.  $d(k)$ ,  $k$  videosunun hedefidir,  $s(k)$ , video  $k$ 'nin kaynağıdır,  $f_p^{k,l}$ ,  $k$  videosunun  $l$  katmanının  $p$  yolundan iletilen bölümüne karşılık gelen akış miktarıdır. Kaynak düğümlerin her birinde net bir akış kazancı vardır. Öte yandan, havuz düğümlerinde net bir akış kaybı vardır. Sistem modelimiz, yetersiz bant genişliği nedeniyle tek bir yoldan iletilmeyen katmanların birleşim noktalarında daha küçük parçalara bölünmesine ve daha sonra birden çok yolla iletilmesine olanak tanımaktadır.

$$\sum_{p \in P} \sum_{j \in N, i \neq j} f_p^{k,l} r_{i,j,p} - \sum_{p \in P} \sum_{j \in N, i \neq j} f_p^{k,l} r_{i,j,p} = \begin{cases} R_{k,l} \times x^{k,l} & i = d(k) \\ -R_{k,l} x^{k,l} & i = s(k), \forall k \in K, i \in N, l \in L \\ 0 & i \neq s(k), i \neq d(k) \end{cases} \quad (3)$$

Bağlantının  $(i, j)$  bant genişliği kısıtlaması (4) 'te verilmiştir. Burada,  $B_{i,j}$  bağlantı akışında bir üst sınır görevi gören  $(i, j)$  bağlantısının bant genişliğini simgelemektedir.  $(i, j)$ 'deki çapraz trafik, dışarıdan olası etkileri modelleyen  $f_{i,j}^0$  ile belirtilir.

$$f_{i,j}^0 + \sum_{p \in P} \sum_{k \in K} \sum_{l \in L} f_p^{k,l} r_{i,j,p} \leq B_{i,j}, \forall i, j \in N \quad (4)$$

$k$  videosunun  $l$  katmanı için gecikme kısıtlaması (5) 'da ifade edilir, burada  $D_{max}^{link}$  bağlantı  $(i, j)$  ' de izin verilen maksimum bağlantı gecikmesidir.

$$y_p^{k,l} \times d_{i,j} \leq D_{max}^{link}, \forall i, j, n, k, l \quad (5)$$

Burada  $d_{i,j}$ ,  $(i, j)$  bağlantısında ölçülen bağlantı gecikmesidir. Bağlantı gecikme kısıtlamasına ek olarak, uçtan-uca gecikme kısıtlaması (6) 'da ifade edilir, burada

$D_{max}^{e2e}$  kaynak ve havuz düğümleri arasındaki bir akış için izin verilen maksimum gecikmedir.

$$\sum_{i \in N} \sum_{j \neq i \in N} y_p^{k,l} r_{i,j,p} d_{i,j} \leq D_{max}^{e2e}, \forall p \in P, k \in K, l \in L \quad (6)$$

Kısıtlama (7)'de bir akışın var olması için  $i$  ve  $j$  düğümleri arasında fiziksel bir bağlantı olması gerektiği ifade edilir.

$$f_n^{kl} r_{i,j,p} \leq y_p^{kl} \times B_{ij} \times a_{ij}, \forall k, l, i, j, p \quad (7)$$

İkili parametre  $a_{ij}$ ,  $i$  ve  $j$  düğümleri arasında bağlantı  $(i, j)$  varsa, 1 değerini alır. Kısıtlama (8), aynı videonun  $l$  ve  $l'$  katmanlarının yollarının  $J_{max}$ 'ın izin verilen gecikme açısından fazla ayrılmamasını sağlayan jitter kısıtlamasıdır.

$$-J_{max} \leq \sum_{i \in N} \sum_{j \neq i \in N} y_p^{k,l} r_{i,j,p} d_{i,j} - \sum_{i' \in N} \sum_{j' \neq i' \in N} y_{m'}^{k',l'} r_{i',j',p'} d_{i',j'} \leq J_{max}, \forall k, l, l', p, p' \quad (8)$$

Ayrıca, yollardan geçen akışlar için sıfır varsayılan alt sınır olarak tanımlanmış olup akışların kesinlikle negatif olmaması sağlanmıştır. Akış kesinlikle negatif olmamaya zorlanarak, benzetimlerde geriye dönük trafik göz ardı edilmiştir; çünkü geriye dönük trafik, sistem sonuçlarının sezgisel olarak anlaşılmasını etkileyebilir. Kısıtlama (9), herhangi bir bağlantıdaki akışın negatif değerler alamayacağını gösterir.

$$f_{ij}^{kl} \geq 0, \forall k \in K, l \in L, i, j \in N \quad (9)$$

Yukarıda tanımlanan ağ kısıtlamalarına ek olarak, Ölçeklendirilebilir video kodlamanın doğası gereği bazı kısıtlamalar da tanımlanmıştır. Kısıtlama (10),  $l$  katmanı ilemeden önce, önceki  $l - 1$  katmanlarının iletilmiş olmasını şart koşar. Bir videonun  $l$  geliştirme katmanının iletimine başlanmadan önce videonun temel katmanından itibaren  $l - 1$  katmanına kadar olan bütün katmanların iletilmiş olması gerektiği ifade edilir. Bu kısıtlamanın sebebi, hiyerarşide kendisinden önce gelen katmanların iletiminin başarısız olması durumunda yeni geliştirme katmanları iletilmenin kaynak israfına sebep olmasıdır. Kısıtlama (11), adaletli iletim kısıtı olup her bir video temel katmanının geliştirme katmanlarına geçmeden önce iletilmesini zorunlu tutar.

$$x^{k,l} \leq x^{k,l-1}; \forall k \in K, l = 2, 3, \dots, L \quad (10)$$

$$x^{k,2} \leq x^{m,1}, \forall m, k \in K \quad (11)$$

Kısıtlama (12) ve (13)'te  $x$  ve  $y$  ikili değişkenleri ifade edilir,

$$x^{k,l} \in \{0,1\}; \forall k \in K, l \in L \quad (12)$$

$$y_p^{k,l} \in \{0,1\}; \forall k \in K, l \in L, p \in P \quad (13)$$

Sorunun amacı ve kısıtlaması şu şekilde sıralanmıştır:

$$\max_{f, x, y} = \sum_{k \in K} \sum_{l \in L} x^{k,l} \times Q_{k,l} \quad (14)$$

$$\sum_{p \in P} \sum_{j \in N, i \neq j} f_p^{kl} r_{i,j,p} - \sum_{p \in P} \sum_{j \in N, i \neq j} f_p^{kl} r_{j,i,p} = \begin{cases} R^{k,l} \times x^{k,l} = d(k) \\ -R^{k,l} \times x^{k,l} = s(k) \\ 0 \quad i \neq s(k), d(k) \end{cases}, \forall k, i, l \quad (15)$$

$$f_{ij}^0 + \sum_{n \in N} \sum_{k \in K} \sum_{l \in L} f_p^{k,l} r_{i,j,p} \leq B_{i,j}, \forall i, j \in N \quad (16)$$

$$y_p^{kl} \times d_{ij} \leq D_{\max}^{\text{link}}, \forall i, j, p, k, l \quad (17)$$

$$\sum_{i \in N} \sum_{j \neq i \in K} y_p^{kl} r_{i,j,p} d_{ij} \leq D_{\max}^{\text{e2e}}, \forall p \in P, k \in K, l \in L \quad (18)$$

$$f_p^{k,l} r_{i,j,p} \leq y_p^{kl} \times B_{i,j} \times a_{ij}, \forall k, l, i, j, p \quad (19)$$

$$\left| \sum_{i \in N} \sum_{j \neq i \in N} y_p^{k,l} r_{i,j,p} d_{i,j} - \sum_{i \in N} \sum_{j \neq i \in N} y_{p'}^{k',l'} r_{p,q,m} d_{p,q} \right| \leq J_{\max}, \forall k, l, l', p, p' \quad (20)$$

$$f_{ij}^{kl} \geq 0, \forall k \in K, l \in L, i, j \in N \quad (21)$$

$$x^{k,l} \leq x^{k,l-1}, \forall k \in K, l \in L \quad (22)$$

$$x^{k,2} \leq x^{m,1}, \forall m, k \in K \quad (23)$$

$$x^{k,l} \in \{0,1\}; \forall k \in K, l \in L \quad (24)$$

$$y_p^{k,l} \in \{0,1\}; \forall k \in K, l \in L, p \in P \quad (25)$$

Sistem modeli incelenirse hem  $f$  gibi sürekli değişkenlerin hem de  $x$  ve  $y$  gibi ikili değişkenlerin olduğu görülmektedir. Bununla birlikte,  $Q_{k,l}$  değişkeni bir sabit olduğu için amaç fonksiyonu doğrusaldır. Aynı şekilde, kısıtlamalar da doğrusaldır ve konvektir. Sonuç olarak, önerilen sistem modeli MILP tabanlı bir çözümdür ve CPLEX tarzı hazır çözücülerle çözülebilir.

## 2.2 Sezgisel Algoritma

Algoritma 1, çok yollu ve çok katmanlı video aktarımı için önerilen sezgisel yöntemi gösterir. Tablo 2’de bu algoritmayı tanımlamak için kullanılan semboller verilmiştir.

Sezgisel algoritma, Şekil 2.2’de gösterilmiş olup katman kalitelerini ( $Q_{k,l}$ ), katmanları göndermek için gereken bant genişliğini ( $R^{k,l}$ ), bağlantı gecikmelerini ( $d_{i,j}$ ), uçtan uca gecikmeleri ( $d_p^{e2e}$ ), izin verilen uçtan uca jitterları ( $J_{max}$ ) ve bağlantı kapasitelerini ( $B_{i,j}$ ) girdi olarak alır. Algoritma, iletilen video katmanlarını tutan  $s_{k,l}$  listesini çıktı olarak döner. Yol gecikmelerini ve jitterları elde etmek için öncelikle alternatif uçtan uca yollar keşfedilir. Yollar, BFS (Satır 2) kullanılarak keşfedilir. Uçtan uca gecikme ve jitter kısıtlamalarını ihlal eden yollar elenir (Satır 4) ve yol kümesi  $P$  oluşturulur. Eğer  $p$  ve  $p'$  yollarının gecikmeleri arasındaki fark jitter kısıtlaması  $J_{max}$ ’tan büyükse, değişken  $J_{p,p'}$  1 değerini alır ve bu iki yol bir akış için birlikte kullanılamaz (Satır 7). Kalan bağlantı kapasiteleri ( $B_{i,j}^R$ ), kalan bant genişliği gereksinimleri ( $R_{k,l}^R$ ) kalan yol kapasiteleri ( $R_p$ ) Satır 8’de hesaplanır.

9-19. Satırlar arasında, her akış için temel katmanlar tahsis edilir. Her adımda algoritma bir temel katman tahsis edilmemiş akışlar arasından birim veri hızı oranı başına en yüksek kaliteye sahip akışı seçer (Satır 10). Seçilen akış için en yüksek kalan kapasiteye sahip yol atanır (Satır 11-12). Yollardan geçen akışlar, bağlantılarda kalan kapasite ve yol kapasitesi 13-15. Satırlarda güncellenir. Eğer bir katmanın kalan hız gereksinimi sıfırsa, o katman bitmiş (gönderilmiş) olarak kabul edilir (Satır 17). Üçüncü aşamada geliştirme katmanlara tahsis edilir (Satır 21-31). Bir önceki aşamaya benzer bir prosedür, burada tekrar edilir. Sezgisel yöntemin karmaşıklığı akış sayısının karesi ( $K^2$ ), ve katmanların sayısının ( $L$ ) çarpımıyla ikinci derece olarak bulunur. Bununla birlikte eğer Satır 11’de ve Satır 2’te yapılan işlemler döngülerin dışında tüm  $K$  ve  $L$ ’ler için ortak olarak yapılırsa ve sonuçlar Birleştirme Sıralaması gibi bir algoritma ile sıralanırsa algoritmanın karmaşıklığı  $O(K \log K)$

olarak bulunur. Önerilen yöntem, Ford-Fulkerson'ın (FF) algoritmasına benzer. FF'den farklı olarak sistem modelinde anlatıldığı gibi sezgisel algoritmada katmanlar arasında öncelik tahsis edilmiştir. Önerilen sezgisel algoritma, yolları belirlemek amacıyla ilk aşamada BFS gerçekleştirir. BFS'nin tipik karmaşıklığı kenar ve köşe sayılarının toplamı olup doğrusaldır.



---

**Algorithm 1** Çok Yollu Çok Katmanlı Video Yönlendirme

---

- 1: Girdi:  $Q_{k,l}, R_{k,l}, D_{link}^{max}, D_{e2e}^{max}, J_{max}, B_{i,j}, d_{i,j}, d_p^{e2e}$
  - 2: Birinci Adım: Uçtan uca yönlendirme yollarını  $\mathcal{P}'$  BFS kullanarak keşfet.
  - 3: Max uçtan uca gecikme ve jitter gereksinimlerini aşan yolları ele.
  - 4: Değer Ata:  $p \in \mathcal{P}$ , eğer  $d_{i,j} \leq D_{max}^{link}$  ve  $d_p^{e2e} \leq D_{max}^{e2e} \forall p \in \mathcal{P}'$
  - 5: Güncelle:  $r_{i,j,p} \forall p \in \mathcal{P}$
  - 6: Tanımla:  $J_{p,p'} = 0$  eğer  $|d_p^{e2e} - d_{p'}^{e2e}| \leq J_{max} \forall n$  ve 1, Diğer durumlarda
  - 7: Başlangıç Değeri Ata:  $B_{i,j}^R = B_{i,j}, \forall i, j \in \mathcal{N}, R_{k,l}^R = R_{k,l}, s_{k,l} = 0, l_k = 1, \forall k \in \mathcal{K}, l \in \mathcal{L}$   
 $R_p = \min_{i,j|r_{i,j,p}=1} \{B_{i,j}^R\}, \forall p \in P$
  - 8: İkinci Adım: Temel katmanlar için Kaynakları Bölüştür.
  - 9: **while** Temel katmanları gönderebilecek kaynak varken **do**
  - 10: Seç:  $k^* = \arg \max_{s_{k,1}=0} \{ \frac{Q_{k,1}}{R_{k,1}} \}$
  - 11: Seç:  $p^* = \arg \max_{p:R_p>0} \{R_p\}$
  - 12: Değer Ata:  $x_{k^*,p^*} = 1, y_{p^*,1}^{k^*} = 1, p_{k^*,p^*}^{used} = 1$
  - 13: Güncelle:  $B_{i,j}^R = B_{i,j}^R - \min\{R_{p^*}, R_{k^*,l}^R\}, \forall i, j$  s.t.  $r_{i,j,p^*} = 1$
  - 14: Güncelle:  $R_{k^*,1}^R = \max\{R_{k^*,1}^R - R_{p^*}, 0\}$
  - 15: Güncelle:  $R_p = \min_{i,j|r_{i,j,p}=1} \{B_{i,j}^R\}, \forall p \in P$
  - 16: **if**  $R_{k^*,1}^R = 0$  **then**
  - 17:      $s_{k^*,1} = 1, l_{k^*} + = 1$
  - 18: **end if**
  - 19: **end while**
  - 20: Üçüncü Adım: Geliştirme katmanları için kaynakları bölüştür
  - 21: **while** Kaynak varken **do**
  - 22: Seç:  $k^* = \arg \max_{s_{k,l_k}=0} \{ \frac{Q_{k,l_k}}{R_{k,l_k}} \}$
  - 23: Seç:  $p^* = \arg \max_{p:R_p>0} \{r_p\}$
  - 24: Değer Ata:  $x_{k^*,p^*} = 1, y_{k^*,1,p^*} = 1, p_{k^*,p^*}^{used} = 1$
  - 25: Güncelle:  $B_{i,j}^R = B_{i,j}^R - \min\{R_{p^*}, R_{k^*,l}^R\}, \forall i, j$  s.t.  $r_{i,j,p^*} = 1$
  - 26: Güncelle:  $R_{k^*,l_{k^*}}^R = \max\{R_{k^*,l_{k^*}}^R - R_{p^*}, 0\}$
  - 27: Güncelle:  $R_p = \min_{i,j|r_{i,j,p}=1} \{B_{i,j}^R\}, \forall p \in P$
  - 28: **if**  $R_{k^*,l}^R = 0$  **then**
  - 29:      $s_{k^*,l} = 1, l_{k^*} + = 1$
  - 30: **end if**
  - 31: **end while**
  - 32: **Çıktı:**  $s_{k,l}, \forall k \in \mathcal{K}, l \in \mathcal{L}$
- 

Şekil 2.2: Çok Yollu Çok katmanlı Video Aktarımı Algoritması

### 3. BENZETİM DEĞİŞKENLERİ SONUÇLARI

Bu bölümde önerilen sistem modeli en yüksek ödül için çözülmüştür ve kıyaslama algoritmalarıyla model arasında performans karşılaştırması yapılmıştır. Video akışlarına ait veri hızı ve kalite parametreleri Çizelge 3.1’de, 3.3 ve 3.4’te gösterilmiştir. Ağ benzetim parametreleri ise Çizelge 3.2’de gösterilmiştir. Benzetimler Gams ve Matlab kullanılarak gerçekleştirilmiştir.

#### 3.1 Simülasyonda Kullanılan Ölçeklendirilebilir Video Yapılandırılması

Önerilen modelin benzetimlerinin yapılması ve önerdiğimiz eniyileme-sezgisel metodlarının kıyaslama algoritmalarıyla karşılaştırılabilmesi için deneysel birer model tanımlanması gerekmektedir. Bu amaçla sistemi modellemek için 2 CIF ve bir 4CIF videosu seçildi. Video 1 ve Video 3 sırasıyla “Akiyo” ve “Soccer” isimli videolar olup  $352 \times 288$  çözünürlüklü Common Intermediate Format (CIF) videolarıdır ve 300 karedir. Video 2 ise  $704 \times 576$  çözünürlüğe sahip “City” isimli bir 4CIF video seçilmiştir ve 600 karedir. Videoların Kare hızı saniyede 30 karedir (fps).

Joint Scalable Video Model’i (JSVM) [27] kullanarak videoları göndermek için gereken veri hızları ve iletilebilecek her bir katmanın çözülünce sağladığı ek bireysel kalite ölçüldü. Bunun için öncelikle videolar ölçeklendirilebilir olarak kodlandı. Kodlayıcı, bu işlemleri gerçekleştirirken videoları aynı niceleme parametreleriyle kalite ölçeklemesi olarak kodlayacak şekilde yapılandırıldı. Benzetimler için seçilen videolar en fazla 8 katman olarak kodlandı, burada 1. Katman temel katmana denk gelmektedir. PSNR değeri olarak Gray Scale Referans alınmıştır. Temel katman Niceleme değişkeni 44 ile kodlanırken her bir ardışık katmanda QP 2 azaltılmıştır. Sonra da her bir kodlanmış video istenilen sayıda katmanlar için alt akışlara ayrıldı. Daha sonra bu alt akışlar çözüldü ve PSNR değerleri ölçüldü. PSNR kümülatif bir



ölçüm olduğu için her bir ardışık katmanın bireysel kalitesini ondan önceki kümülatif toplamdan çıkararak her bir katman tarafından sağlanan kalite bulunmuştur.

Ölçeklendirilebilir videoyu kodlamak için kaliteyi ölçeklendirilmesi seçilmiştir. Benzetimler için, videolar en fazla 8 katmanla kodlanmıştır ve ilk katman her bir videonun temel katmanını göstermektedir. Gray Scale Peak Signal to Noise Ratio(PSNR) değerleri yeniden oluşturulmuş videonun kalitesini, referans videoyu baz alarak ölçmek için kullanılır. Kaliteyi ölçmek için öncelikle Eşitlik (26)'da video karelerinin ortalama karesel hatalarının hesaplamakta kullanılan formül gösterilmiştir. Burada  $F$  ve  $F'$  sembolleri sırasıyla orijinal ve yeniden oluşturulmuş videoları gösterir.  $R$  ve  $C$ , video karesinin satır ve sütun sayısını ifade eder.  $F(m, n)$ , video karesinde m'inci satırın n'inci sütununun sahip olduğu piksel değerini gösterir. Daha sonra ortalama karesel hatayı kullanarak Eşitlik (27)'de PSNR hesaplama formülü kullanılmıştır. Burada  $F_{max}$  maksimum piksel şiddetini ifade eder. Çizelge 3.1'de ve Çizelge 3.2'de benzetimlerde kullanılan videoların özellikleri verilmiştir.

PSNR'a ek bir referans olarak, önerilen modelin QoE çıktısını diğer çalışmalarla karşılaştırabilmek için videoların Structural Similarity Index Metric (SSIM)'leri de hesaplanmıştır [28]. SSIM videoların parlaklık, karşıklık ve yapısal özelliklerini baz alarak görsel kaliteyi hesaplayan bir kalite referansıdır.

$$MSE = \frac{1}{RC} \sum_{m=1}^R \sum_{n=1}^C \|F(m, n) - F'(m, n)\|^2 \quad (26)$$

$$PSNR = 20 \log \frac{F_{max}}{\sqrt{MSE}} \quad (27)$$

Çizelge 3.1 : Videoları aktarmak için gereken bant genişlikleri

$R^l_k$ (kbps) her bir katman için			
Katman dizini	Akış 1	Akış 2	Akış3
Katman1	95.0	648.8	233.8
Katman2	29.2	173.0	35.4
Katman3	38.4	231.7	45.3

Katman4	49.3	303.4	75
Katman5	54.6	379.1	109
Katman6	75.6	533.0	171
Katman7	94.5	721.0	295
Katman8	109.4	924.0	303.0

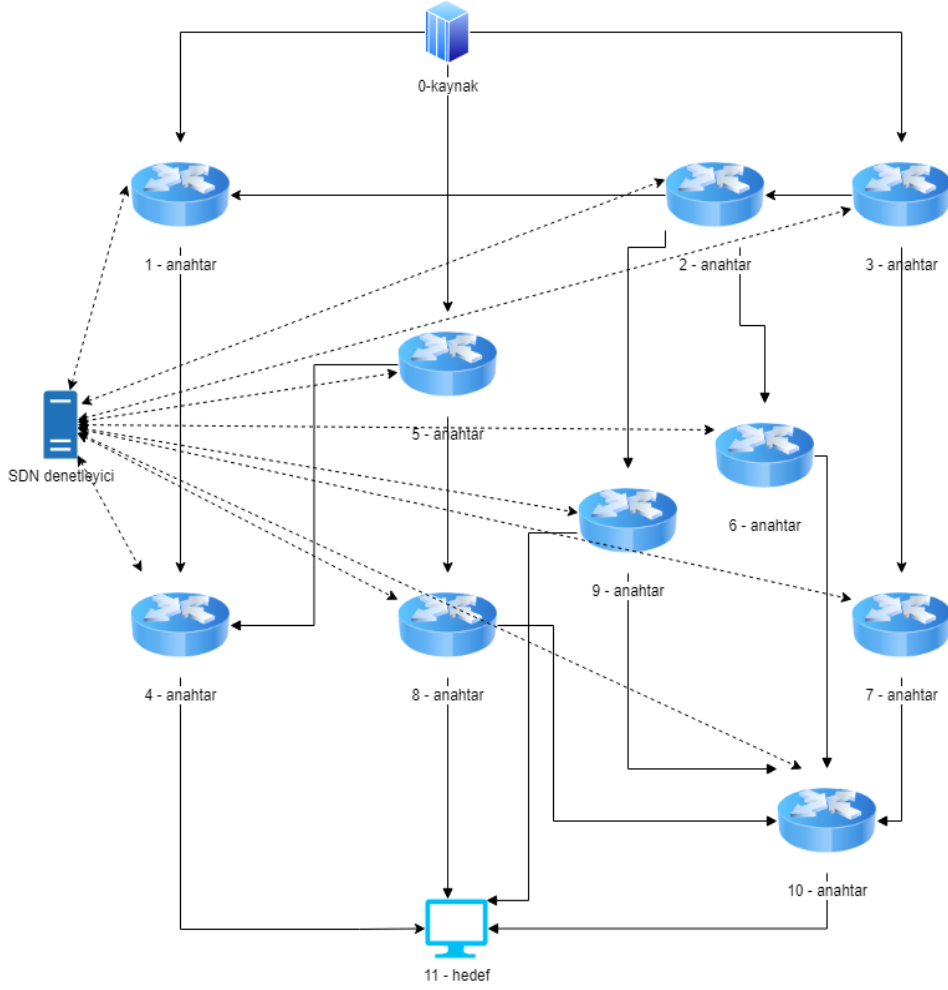
### 3.2 Ağ Benzetim Değişkenleri

Benzetimlerde Şekil 3.1’de gösterilen topoloji kullanılmıştır. Topoloji 12 düğüm ve 19 bağlantıdan oluşmaktadır. 0 düğümü kaynak ve 11 düğümü hedef olarak seçilmiştir. Örneğin bir akış 0–3–7–10–11 yolunu izleyebilir. Bu şekilde kışların izleyebileceği 11 yol tespit edilmiştir. Bağlantılara ölçülen gecikme ve çarpaz trafik için uygun değerler tahsis edilmiştir. Burada amaç kaliteyi en iyileyecek şekilde akışları belirlemektir.

Benzetimlerde tıkanma veya paket kaybı gibi ağ kısıtlamaları ihmal edilmiştir. Eğer yeterli bant genişliği sağlanamazsa bir akış olmayacağı kabul edilmiştir. Önerilen algoritmaları ve kıyaslama algoritmalarını değerlendirmek için kullanılan değişkenler Çizelge 3.2’de verilmiştir. Aksi belirtilmedikçe her bir bağlantı gecikmesi 20 ve izin verilen uçtan uca gecikme 100 milisaniyedir.

Çizelge 3.2 : Ağ benzetim değişkenleri

Değişken	Değeri
i-j bağlantısında ölçülen gecikme	20msec $\forall i, j$
İzin verilen maksimum bağlantı gecikmesi	20 msec
İzin verilen maksimum uçtan-uca gecikme	120 msec
İzin verilen uçtan-uca jitter	40 msec
i-j bağlantısından geçen çarpaz trafik	20Mbps
i-j bağlantısının bant genişliği	> 20Mbps



Şekil 3.1: Benzetimlerde kullanılan ağ topolojisi

### 3.3 Benzetim Sonuçları

Daha önceden kuramsal modeli verilen optimizasyon problemi, kontrollü değişken görevi gören bağlantı kapasitesinin sıralı değerleri için Gams'ta çözülmüştür. Buna ek olarak tek yol yönlendirmeye dayalı iki kıyaslama algoritması önerilmiştir. Son olarak eniyileme çözümüne ek olarak gösterdiğimiz sezgisel algoritmanın (Algoritma 1) benzetimi yapılmıştır. Çizelge 3.3'te her bir video katmanının sağladığı bireysel kaliteler gösterilmiştir. Çizelge 3.4'te bireysel katman kaliteleri SSIM metriği cinsinden gösterilmiştir.

Çizelge 3.3 : Her bir video katmanının sağladığı bireysel kalite

$Q'_k$ ( PSNR)			
Katman dizini	Akış 1	Akış 2	Akış3
Katman1	33.9536	32.7021	31.0309
Katman2	0.6347	0.5491	0.1047
Katman3	0.5862	0.4911	0.0789
Katman4	0.5478	0.48718	0.1570
Katman5	0.5966	0.4713	0.3634
Katman6	0.8778	0.6674	0.6523
Katman7	1.0339	0.7056	0.9440
Katman8	0.9702	0.7775	0.9406

Çizelge 3.4 : Her bir video katmanının sağladığı bireysel kalite SSIM cinsinden

$Q'_k$ ( SSIM)			
Katman dizini	Akış 1	Akış 2	Akış3
Katman1	0.9158	0.9003	0.7636
Katman2	0.0089	0.0122	0.0081
Katman3	0.0089	0.0107	0.0090
Katman4	0.0092	0.0085	0.0101
Katman5	0.0065	0.0072	0.0121
Katman6	0.0077	0.0083	0.0181
Katman7	0.0064	0.0076	0.0242
Katman8	0.0049	0.0069	0.0243

### 3.3.1 Kıyaslama algoritması 1 tek yollu yönlendirme

Önerilen sistem modelini analiz edebilmek için tek yol yönlendirmeye dayalı bir sezgisel yönlendirme algoritması geliştirilmiştir. Ayrıca bu algoritma için yeni bir kalite referansı geliştirilmiştir. Çizelge 3.5'te katmanların bireysel kalitelerinin, katmanları göndermek için gereken veri hızına bölünmesinden elde edilen sonuç gösterilmiştir ve bu metrik kıyaslama algoritmalarında temel alınmıştır. (28)'de görüldüğü gibi bu oranı yol uzunluğuna bölerek kıyaslama algoritmalarının kalite referansı olarak kullanılmaktadır. Burada  $\pi$  kullanıma uygun bütün yollardan oluşan bir kümedir.  $\pi_{k,l}$ ,  $k$  videosunun  $l$  katmanının gönderilmesi için seçilen yolu gösterir ve  $|\pi_{k,l}|$  iletimde kullanılan yoldaki düğümler arasındaki atlayış sayısını gösterir.  $k$  videosunun  $l$  katmanın seçilen yol için göreceli kalite değeri  $IM_{\pi_{k,l}}$  olarak gösterilir ve referans kalite (28)'deki gibi hesaplanır.

$$IM_{\pi_{k,l}} = \frac{Q_{k,l}}{R_{k,l} \times |\pi_{k,l}|} \quad (28)$$

Algoritma 2’de önerdiğimiz çözümü karşılaştırmak için tasarladığımız kıyaslama algoritması tanıtılmıştır. Kıyaslama algoritması her bir katman için sezgiseldir, her bir katmanın gönderilmesi için en iyi yolu diğer katmanların gönderilmesinden bağımsız hesaplar. Öncelikle, sezgisel algoritmaya benzer şekilde BFS algoritması yardımıyla kaynaktan hedefe giden olası yollar keşfedilir (Satır 4-8). Keşfedilen yollardan ağ kısıtlarını sağlayanlar diğerlerinden ayrılır ve  $P$  kümesinde saklanır. Bu yollar kullanılarak bir kombinasyon tablosu yaratılır. Bu tablo her videonun her bir katmanının kullanabileceği bütün olası yolların kümesinden oluşur. Bu olasılıklara herhangi bir katmanın gönderilmeme olasılığı da dahildir. Bu olasılıklar aday gönderim seçenekleri olarak adlandırılır. Tabloda her bir satırda  $k$  video akışını göndermek için kullanılacak yol seçenekleri yer alır. Bu tablo döngü içerisine sokularak her bir olası yol seçimi için beklenen kaliteler hesaplanabilir ve bu sonuçlara dayanarak bir seçim yapılabilir. Her bir katman için kısıtları sağlayan tüm aday yapılandırmalar değerlendirilir (Satır 10-25). Bir katmanın gönderilmesinin ön koşulu olarak, kaynak israfını engellemek için kendisinden önceki katmanların gönderilmesi şart koşulmuştur (Satır 16-18). İkinci olarak da test edilmekte olan yolun kullanılabilir bant genişliğinin ( $B_{i,j}$ ) katmanı göndermek için gereken veri hızını ( $R_{k,l}$ ) karşılayıp karşılamayacağı kontrol edilmiştir. Eğer bu iki şart karşılanırsa denenen yol uygun bir aday olarak kabul edilir ve hesaplanan aday kalite aday listesi  $C_{ci}$ ’de saklanır. Eşitlik (28), aday yapılandırmanın kullanımından beklenen kaliteyi hesaplamak için kullanılır. Burada,  $Q_{k,l}$ ,  $k$  video akışının  $l$  katmanının sağladığı bireysel kalite artışını,  $R_{k,l}$ ,  $k$  video akışının  $l$  katmanını iletmek için gereken veri miktarını,  $|\pi_{k,l}|$  video katmanının iletimi sırasında kaç farklı anahtar düğümünden geçildiğini gösterir. Sonraki aşamada, (Satır 13-18) kullanımından en yüksek kaliteyi sağlayacak yol yapılandırmaları seçilir ve  $s_{k,l}$  seçim listesine kaydedilir.

Son olarak, eniyilenmiş yol seçimine göre sistemde kullanılmayan bant genişliğini gösteren ( $B_{i,j}$ ) tablosunda kullanılan  $i-j$  bağlantılarının kalan kapasitesi güncellenir. Algoritma, en yüksek kaliteyi sağlamak için iletilecek katmanları ( $s_{k,l}$ ) ve bu katmanları göndermek için seçilen yol yapılandırmalarını (**pathselected** $_{k,l}$ ) çıktı olarak döner.

Çizelge 3.5: Bireysel katman kalitelerinin göndermek için gereken veri hızına oranı

$Q^l/R^l_k$			
Katman dizini	Akış 1	Akış 2	Akış3
Katman1	0.3574	0.0459	170.7
Katman2	0.0217	0.0370	59.0
Katman3	0.0150	0.0293	79.8
Katman4	0.0111	0.0230	114.5
Katman5	0.0109	0.0249	131.9
Katman6	0.0100	0.0186	188.7
Katman7	0.0100	0.0146	247.7
Katman8	0.0088	0.0114	303.0

---

**Algorithm 2** Tek Yollu Yönlendirme

---

```
1: Girdiler:  $R_{k,l}$ ,  $Q_{k,l}$ ,  $B_{i,j}$ .
2: Tanımla  $s_{k,l} = 0$ ,  $\forall l \in \mathcal{L}, \forall k \in \mathcal{K}$ .
3: Birinci Adım: Uçtan uca yönlendirme yollarını BFS kullanarak keşfet.
4: Max uçtan uca gecikme ve jitter gereksinimlerini aşan yolları ele.
5: Değer Ata:  $p \in \mathcal{P}$ , if  $d_{i,j} \leq D_{max}^{link}$  and  $d_p^{e2e} \leq D_{max}^{e2e} \forall p \in \mathcal{P}'$ 
6: Başlangıç değeri ata:  $B_{i,j}^R = B_{i,j}$ ,  $\forall i, j \in \mathcal{N}$ ,  $s_{k,l} = 0$ ,  $\forall k \in \mathcal{K}, l \in \mathcal{L}$ 
7: Uygun yollar setinden yol kombinasyon listesi  $T_{ci,k}$  oluştur.  $T_{ci,k}$  her bir
   akış için bütün olası yol tercihlerini tutmaktadır.
8: İkinci Adım: Bütün yol yapılandırmaları için benzetimler gerçekleştir.
9: for  $l=1$  to  $L$  do
10:   Başlangıç değeri ata:  $C_{ci}$  aday kalite listesi:  $\forall p \in \mathcal{P}$ .
   Başlangıç değeri ata  $X_k$  Aday gönderim listesi:  $\forall k \in \mathcal{K}$ .
11:   for  $T_{ci,k}$ 'de yer alan her bir uygun yol kombinasyonu için do
12:     Değer ata: Sırası gelen yol kombinasyon setini  $\pi_{k,l}$ 'a ata.
13:     Başlangıç değeri ata:  $B_{i,j}^R = B_{i,j}$ ,  $\forall i, j$ ,  $s_{k,l} = 0$ ,  $\forall k \in \mathcal{K}, l \in \mathcal{L}$ ,
14:     for  $k^*=1$  to  $K$  do
15:       Güncelle:  $B_{i,j}^R = B_{i,j}^R - R_{k,l}; \forall i, j, i \neq j \in \pi_{k,l};$ 
16:       Başlangıç değeri ata: bool D true
17:       if  $L > 0$  then
18:         Değer ata:  $D : 1 == S_{l-1,k};$ 
19:       end if
20:       if  $B_{i,j}^R \geq 0; \forall i, j$  and  $D$  then
21:         Hesapla:  $C_{ci} = C_{ci} + \frac{Q_{k,l}}{R_{k,l} \times |\pi_{k,l}|}$ 
22:       else
23:         break
24:       end if
25:     end for
26:   end for
27:   Seç:  $maxselection = \arg \max_{ci} C_{ci}$ 
28:   Seç:  $pathselected_{k,l} = C_{maxselection}$ 
29:   Güncelle:  $B_{i,j} = B_{i,j} - R_{k,l}; \forall i, j, i \neq j \in pathselected_{k,l}; \forall k$ 
30:   Güncelle:  $s_{(k:pathselected_{k,l} \neq null),l} = 1; \forall k \in \mathcal{K}$ 
31: end for
32: Çıktı:  $s_{k,l}, pathselected_{k,l}$ .
```

---

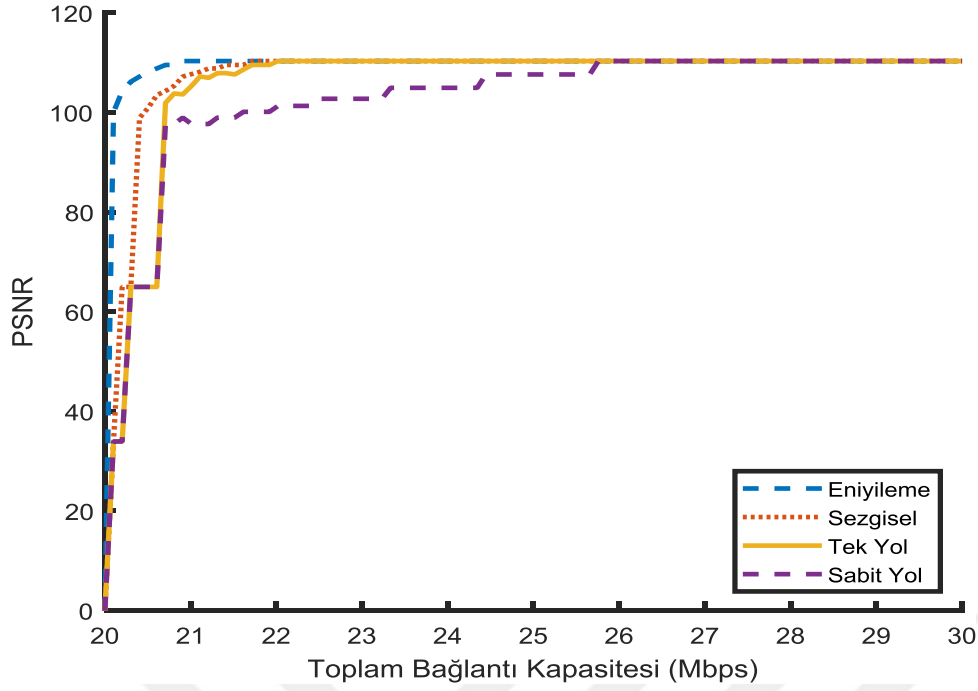
Şekil 3.2: Tek yollu çok katmanlı video aktarımı algoritması

### 3.3.2 Kıyaslama algoritması 2 sabit yol yönlendirme

Kıyaslama algoritması 1'e benzer bir şekilde Kıyaslama algoritması 2 sabit yol yönlendirmeye dayalı bir algoritmadır. Ama Kıyaslama Algoritması 1'den farklı olarak en iyi yol seçimi yalnızca temel katmanlar için yapılır. Bütün geliştirme katmanları temel katmanları için seçilen yolları kullanır.

Şekil 3.3'da önerilen çok yollu eniyileme ve ona ek olarak sunulan sezgisel çok yollu çözümle kıyaslama algoritmaları olan tek yol ve sabit yol algoritmalarının performans karşılaştırılması görülmüştür. Şekil 3.3'te kullanılan yapılandırmada geliştirme katmanlarından önce tüm temel katmanların aktarılması gerektiği şartı ön koşulmuştur. Bu şart Yapılandırma 1 olarak sınıflandırılmaktadır. Bu modelde adaletli dağıtım yüksek kaliteden daha önceliklidir. Burada yatay eksen bağlantı kapasitelerini gösterir. Linklerin kapasitelerinin 20 Mbps'si çarpaz trafiklere ayrılmıştır. Şekil 3.3 eniyilenmiş çoklu yol algoritmasının, sezgisel çok yollu algoritmanın, tek yol ve sabit yol kıyaslama algoritmalarının kıyaslamasını göstermektedir. Sonuçlar incelenirse önerilen çok yollu eniyileme algoritmasının tepe noktasına ulaştığı noktayla çok yollu sezgisel algoritmanın tepe noktasına ulaştığı nokta arasında 0.5 Mbps fark olduğu görülecektir. Çok yollu eniyileme, çok yollu sezgisel algoritma, tek yol ve sabit yol algoritmalarının en yüksek değere ulaşmak için sırasıyla 0.2, 0.4, 0.6 ve 2 Mbps her bir bağlantı için fazladan kapasiteye ihtiyacı vardır. Önerilen çok yollu sezgisel algoritma neredeyse çok yollu eniyileme algoritması kadar iyi sonuç vermektedir.





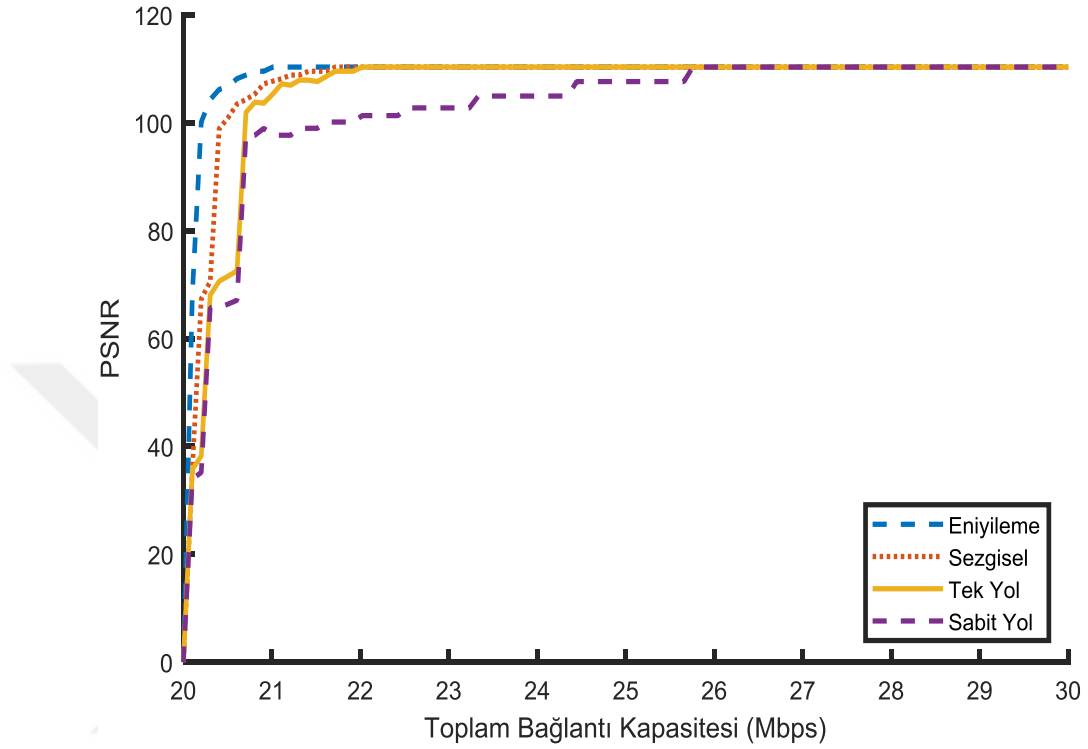
Şekil 3.3 : Performans karşılaştırması (Yapılandırma 1)

### 3.4 Yeniden Yapılandırılmış (Adil Olmayan) Sistem Modeli

Daha ayrıntılı performans analizleri için bu çalışmada önerilen modelin gönderilme şartlarında esneklik sağlanmıştır. Geliştirilme katmanları gönderilmeden önce bütün temel katmanların gönderilmesini şart koşan Kısıtlama (12) kaldırıldı. Bu yapılandırmada adaletten önce elde edilecek toplam kalitenin önemsendiği söylenebilir. Sonuç olarak Yapılandırma 1'den farklı olarak bir video akışının geliştirme katmanları diğer videoların temel katmanların aktarılmasını beklemeden gönderilebilir. Bu Yapılandırma 2 olarak sınıflandırılmıştır. Bu yeni yapılandırma sezgisel çok yollu ve kıyaslama algoritmalarına da uygulandı. Yapılandırma 1'e benzer bir metodoloji kıyaslama algoritmalarına da uygulandı.

Şekil 3.4, Yapılandırma 2 için yapılan benzetimleri göstermektedir. Burada öncelikli olarak tüm temel katmanların aktarılma zorunluluğu ortadan kaldırılmıştır. Sonuç olarak bir videonun geliştirme katmanları, videonun temel katmanı aktarıldıktan sonra iletmeye başlanabilir, diğer videoların temel katmanların aktarılmasını beklemez. Yapılandırma 1'e benzer şekilde çok yollu eniyileme algoritması ve çok yollu sezgisel algoritma kısıtlı kaynaklar için kıyaslama algoritmalarından daha iyi performans göstermektedir. Özellikle link kapasitesinin 20 ve 21 Mbps arasındaki

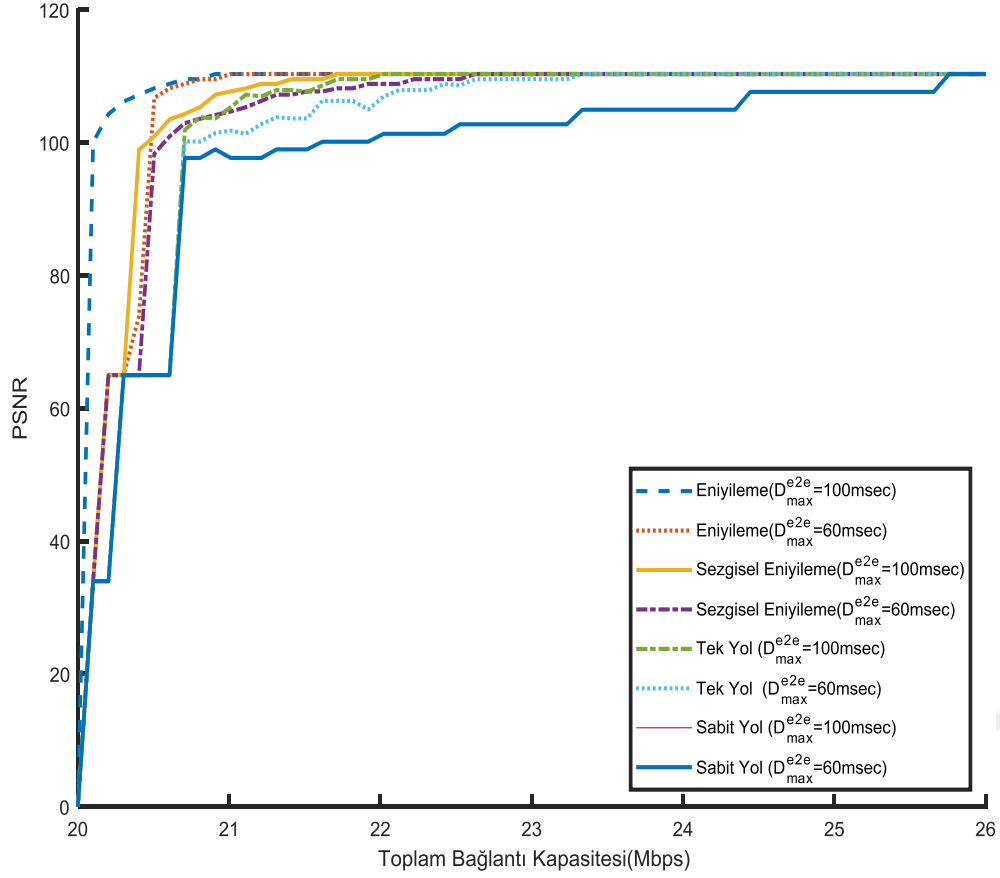
değerleri için kıyaslama algoritmalarına oranla %40 daha iyi performans göstermektedir. 21 Mbps'ten büyük kapasite değerleri için tek yol algoritma önerilen çözüme yaklaşmaktadır. Sabit yol algoritması diğerlerinden çok daha kötü performans göstermektedir.



Şekil 3.4 : Performans karşılaştırması (Yapılandırma 2)

Şekil 3.5'te izin verilen uçtan uca gecikme miktarının performans üzerindeki etkisini göstermektedir. Seçilen maksimum uçtan uca gecikme değerleri sırasıyla 36,60 ve 100 milisaniyedir. Çok yollu eniyileme algoritmasının performansı, gecikme değeri 60'tan 100'e çıktıkça iyileşmektedir. İzin verilen uçtan uca gecikme 100 milisaniye için çok yollu eniyileme algoritması, tepeye ulaşmak için 0.3 – 0.4 Mbps daha az kapasiteye ihtiyaç duymaktadır.

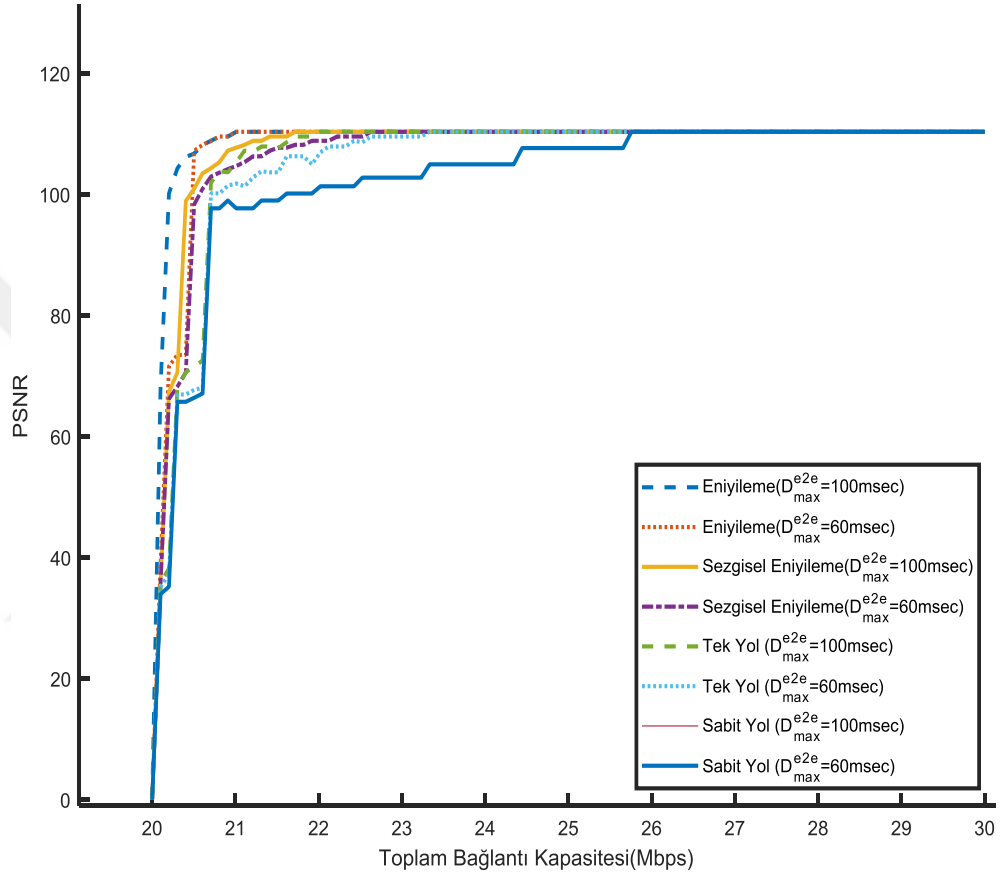
Önerilen sezgisel algoritma eniyileme algoritmasının performansına çok yakın sonuçlar vermiştir. Uçtan uca gecikme kısıtının tüm değerleri için hem çok yollu eniyileme hem de çok yollu sezgisel kıyaslama algoritmalarından daha iyi sonuç vermiştir.



Şekil 3.5 : Uçtan uca gecikmenin performans üzerindeki etkisi (Yapılandırma 1)

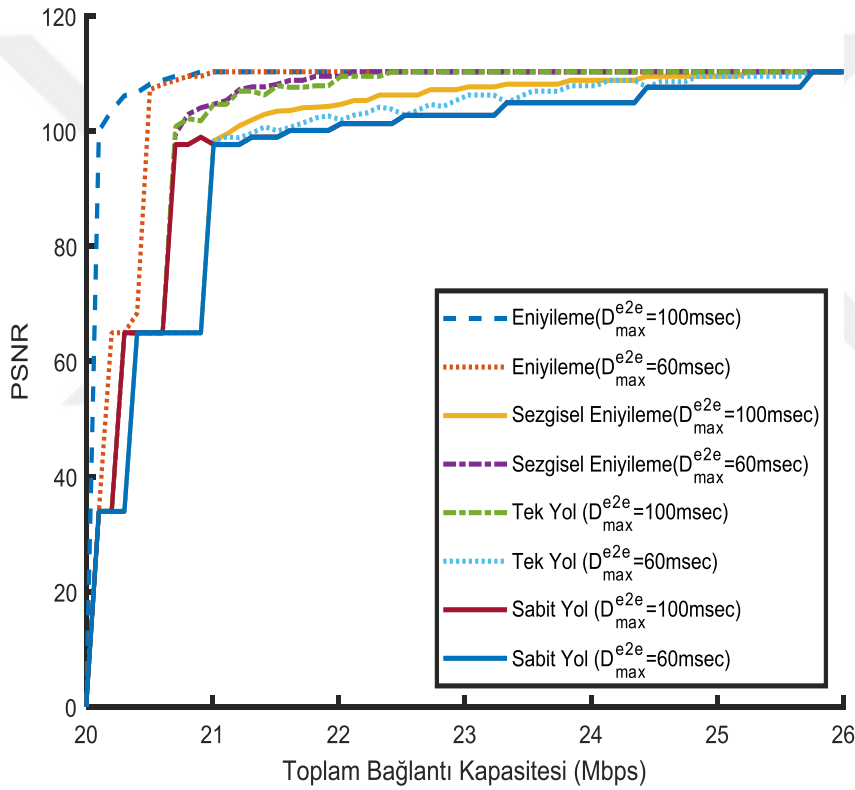
Kıyaslama algoritmalarının performansı bağlantı kapasitesi artarken artış göstermektedir. Bununla birlikte 60 milisaniye uçtan uca gecikme değeri için önerilen eniyileme algoritmasının performansına ulaşamamaktadır. Bunlara ek olarak önerilen algoritmaların kıyaslama algoritmalarının performanlarında uçtan uca jitterın etkisi de test edildi ama sonuçlarda algoritmalar arasında bir etki gözlenemediğinden sonuçlarına bu tezde yer verilmedi. Jitter ve yol uzunluğu 60 milisaniye ve 100 milisaniye uçtan uca izin verilen gecikme kısıtları arasındaki performans farkından ötürü önerilen çok yollu yönlendirme algoritması üzerinde bir etkisi olabilir.

Şekil 3.6'da Yapılandırma 2'de izin verilen uçtan uca gecikmenin farklı değerleri için performans değerlerini göstermektedir. Yapılandırma 1'e benzer bir şekilde 100 milisaniye uçtan uca gecikme, 60 milisaniye uçtan uca gecikmeye göre 0.3 Mbps performansta iyileşme göstermektedir. Tek yol algoritması en yüksek PSNR değerine 22.5 Mbps bağlantı kapasitesinde ulaşırken, sabit yol algoritmasının en yüksek PSNR değerine ulaşması için 26 Mbps bant genişliği gereklidir.



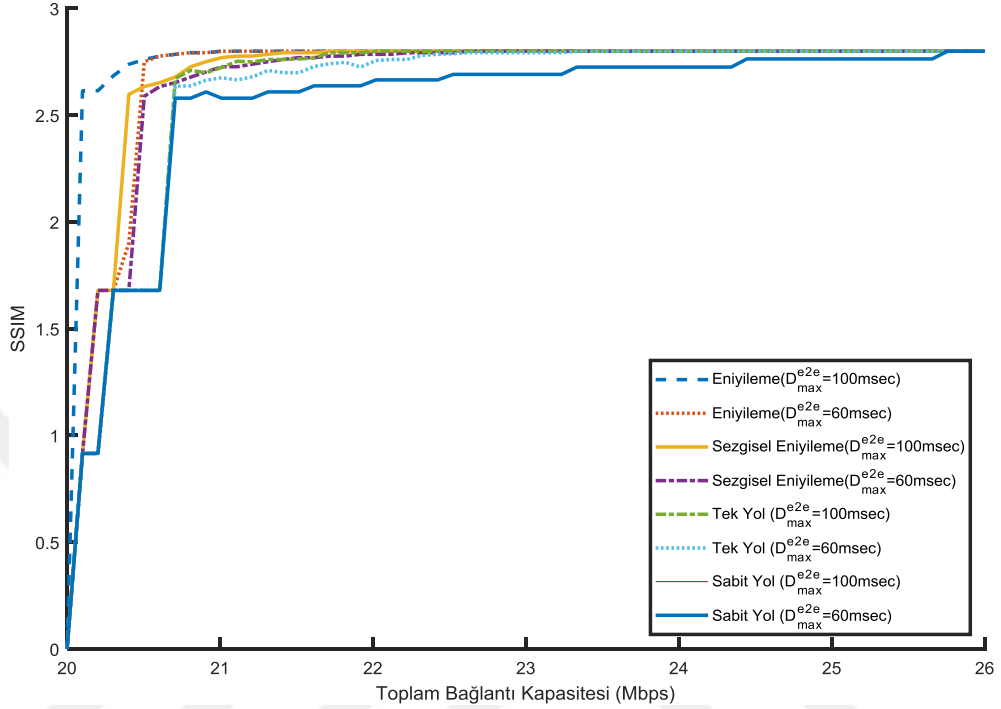
Şekil 3.6 : Uçtan uca gecikmenin performans üzerindeki etkisi (Yapılandırma 2)

Şekil 3.7’de, Yapılandırma 1’de alternatif yolların kısıtlanması sonucu algoritmaların performanslarını göstermektedir. Burada 3 düğümden oluşan oluşan 3 yolun 2’sinin kullanılmasına izin verilmemiştir. Sonuçlar yol kısıtlamasının, önerdiğimiz çok yollu eniyileme algoritmasının performansında önemli bir değişim yaratmazken sezgisel çok yollu algoritmasının performansında küçük bir düşüş yarattığını göstermiştir. Yol kısıtlaması en büyük etkiyi kıyaslama algoritmalarında göstermiştir. Özellikle kıyaslama algoritmalarında biri olan sabit yol algoritması bütün katmanlar için aynı yolu kullanmasından ve diğer algoritmalara kıyasla daha katı olmasından dolayı algoritmanın performansı kısıtlamadan ötürü daha fazla düşmüştür.



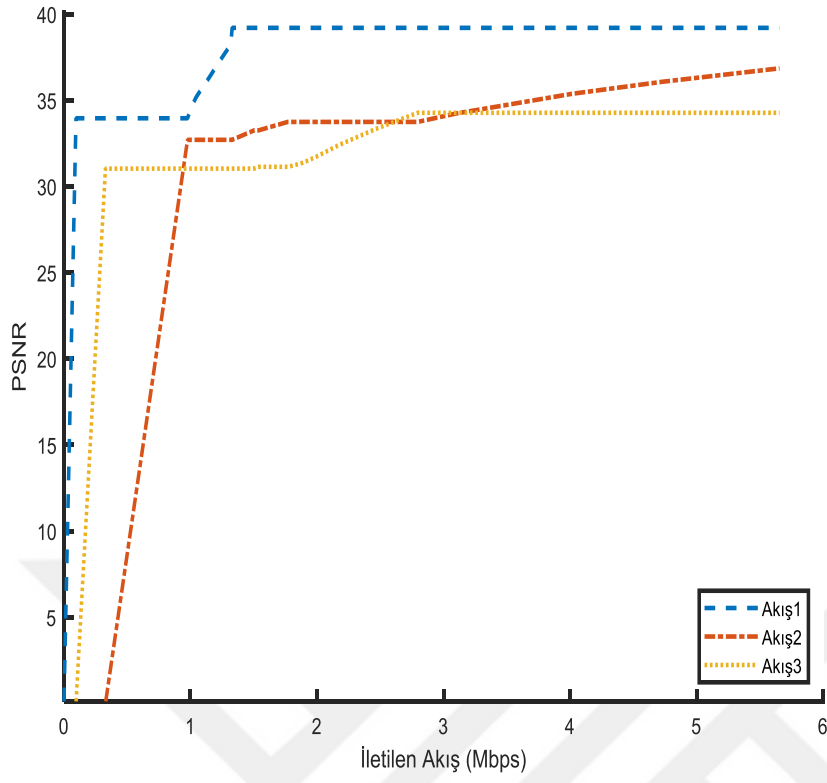
Şekil 3.7 : Yolları kısıtlanmış ağda uçtan uca gecikmenin performans üzerindeki etkisi

Şekil 3.8’de önerilen algoritmaları karşılaştırmak için performans metriği olarak diğer şekillerde kullanılan PSNR yerine SSIM metriği seçilmiştir.



Şekil 3.8: Uçtan uca gecikmenin yapılandırma 1 için performans karşılaştırmasının SSIM cinsinden gösterimi

Şekil 3.9’da artan link kapasiteleriyle orantılı olarak bireysel video kapasitelerinin artışı görülebilir. Şekilden görüldüğü üzere, Akış 1’in ve Akış 3’ün, Akış 2’ye oranla çözünürlükleri daha küçük olup iletim için daha az kaynağa ihtiyaç duymaktadır. Sezgisel algoritma bu nedenle Akış 1’in ve Akış 3’ün iletilmesine öncelik vermektedir. Bu durum ağ sistemlerinde adaletli iletim kriterlerine zarar verebilir. Bu durumun önüne geçmek için benzetimlerde Eşitlik 29 yerine farklı bir metrik kullanılabilir.



Şekil 3.9: Çözünürlüğün eniyilendirilmiş video akışına etkisi (Yapılandırma 1)

## 4. MİNİNET VE POX EMÜLASYON SONUÇLARI

### 4.1. Gerçek zamanlı uygulanması

Gerçekleştirmeler için Mininet v2.22 [29] ve POX "l2-multi.py" öğrenme anahtarına dayalı [30] SDN denetleyici modülü kullanılmıştır. Kaynak ve hedef düğümler arasındaki yolları keşfetmek için OpenFlow Discovery ve OpenFlow Spanning tree [31] protokollerinden yararlanılmıştır. Ağlarda topoloji keşfinin önemi nedeniyle, SDN denetleyicileri tarafından sıklıkla kullanılan OpenFlow Keşif Protokolünü (OFDP), kullanılmıştır. OFDP protokolü, bağlantıları keşfetmek ve anahtarların bir listesini döndürmek için kapsüllenmiş The Link Layer Discovery Protocol (LLDP) Paketlerini kullanır. Bağlantıların keşfinden sonra SDN denetleyicisi ve Mininet kullanılarak önerilen sezgisel algoritmalara göre hangi yolların kullanılacağı belirlenir.

SDN denetleyicisinin birden fazla akışı eş zamanlı yönlendirmedeki yetersizliği nedeniyle en gerçekçi senaryoyu elde etmek için aşamalı bir yaklaşım kullanıyoruz. Bu senaryoda, bir akışın katmanlarını ardışık olarak gönderiyoruz ve ilgili katmanları gönderdikten sonra yoldaki kalan bağlantı kapasitelerini güncelliyoruz. Bağlantılara kapasite tahsisleri, Mininet kullanılarak dinamik olarak güncellenir. Benzetimler için, öğrenme anahtarı ve sanal Mininet ağları birbirlerinden bağımsız çalıştırılarak kullanılmıştır. İlk önce topoloji Mininet'te emülasyon yapılmaktadır ve düğümler arası video akışları yaratılmaktadır. Akışların iletimi bittikten sonra Mininet, sunucu ve müşteri sonuçlarını değerlendirir. Mininet ile learning switch arasındaki tek etkileşim öğrenme anahtarı ve Mininet şu anda hangi katmanın iletildiği bilgisidir. Bizim senaryomuzda Mininet, POX denetleyicisine iletim başladığında bilgi verir ve bu katman için ayrılan bant genişliğine göre kalan bağlantı bant genişliğini güncellenir. Bu adımlar aşağıdakiler için gerçekleştirilir:

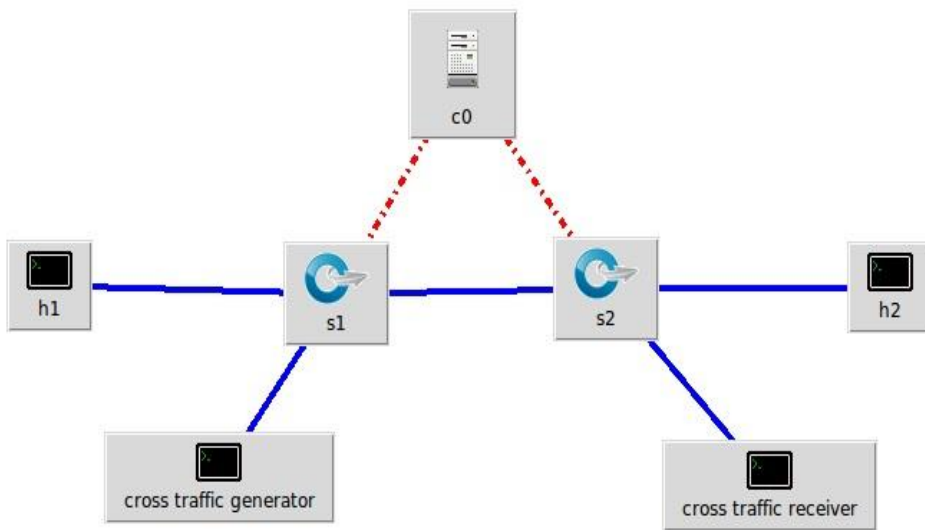
Test bant genişliği aralığındaki her bir değer için bu aşamalar tekrar edilmektedir. Mininet öykünmeyi başlattığında denetleyici, OpenFlow keşif olaylarıyla başlatılır.



LinkEvents, bağlantı bilgilerini, bitişikliği ve bağlantı noktası matrisini tutan çizelgeye başlangıç değerlerini atanır ve bağlantıları arası yönlendirmeleri başlatmak için kullanılır. Yollar bu matrise göre güncellenir.

Mininet hangi katmanın gönderileceğine karar verirken, bir paket aktarımı hakkında SDN keşfini tetikler ve SDN denetleyicisi hangi yolun kullanılacağına karar verir. Ardından denetleyici tarafından, ilgili learning swithlerin yönlendirme çizelgeleri güncellenir.

Bunlara ek olarak Mininet'te emülasyon sonuçlarında kullanılmak üzere bir gecikme ölçüm metodu geliştirmiştir. Yalnızca 2 anahtar ve 2 sunucudan oluşan bir ağ emülasyon yapılarak ve deneylerde kullanılabilecek yaklaşık bir bağlantı gecikmesi değeri ölçülmektedir [32], [33]. Kullanılan topoloji Şekil 4.1'de görülebilir. Gecikme şu şekilde hesaplanmıştır. Öncelikli denetleyici tarafından birinci anahtarın 2.Anahtara ping attıktan sonra denetleyiciye atış zamanını belirtmesi istenmiştir. 2. Anahtara ise paket gelince denetleyiciye haber vermesi talimatını verir. Bunlardan biri anahtarlar arası gecikme ve diğer ikisi anahtarlarla denetleyiciler arasındaki gecikme 3 farklı gecikmenin toplamını elde ederiz. Buna ek olarak denetleyici ile anahtarlar arasındaki ortalama gecikme port stats request paketinin gönderimi ile port stats reception paketinin alınması arasındaki sürenin yarısıdır.

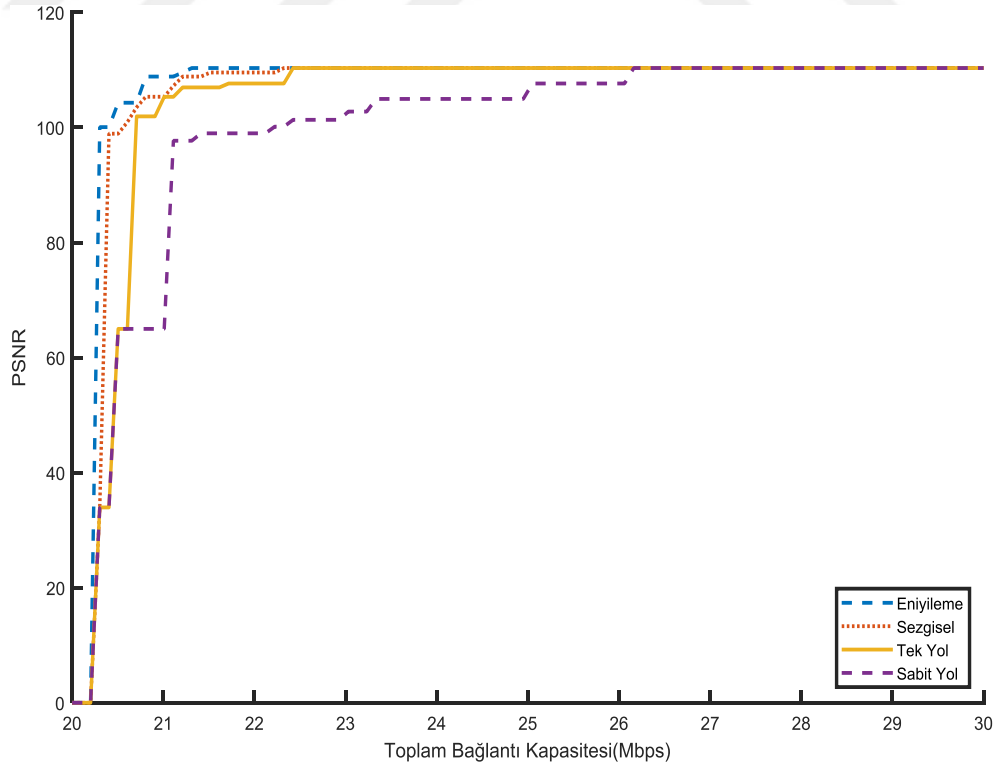


Şekil 4.1: Gecikmeyi ölçmek için kullanılan Mininet senaryosu

Önerilen çok yönlü sezgisel algoritmanın uygulama detayları yukarıda açıklanmıştı. Bu bölümde Mininet’te yapılan emülasyon sonuçları gösterilmiştir ve önerilen çok yönlü algoritmalarla kıyaslama algoritmaları karşılaştırılmıştır. Önerilen algoritmanın emülasyonu için, GAMS ile elde edilen sonuçlar, önceden işlenmiş veri olarak Mininet’e girdi olarak kullanılmıştır. Kıyaslama algoritmaları ve Sezgisel algoritma, yönlendirme yollarını hesaplamak için Mininette kullanılabilir şekilde Python modülü olarak yazılmıştır. Emülasyonları gerçekleştirirken gereken hesaplamaların sayısının çok yüksek olması ve Mininet’in dengesiz yapısından ötürü bazı hesaplamalar önceden işlenerek yapılmıştır.

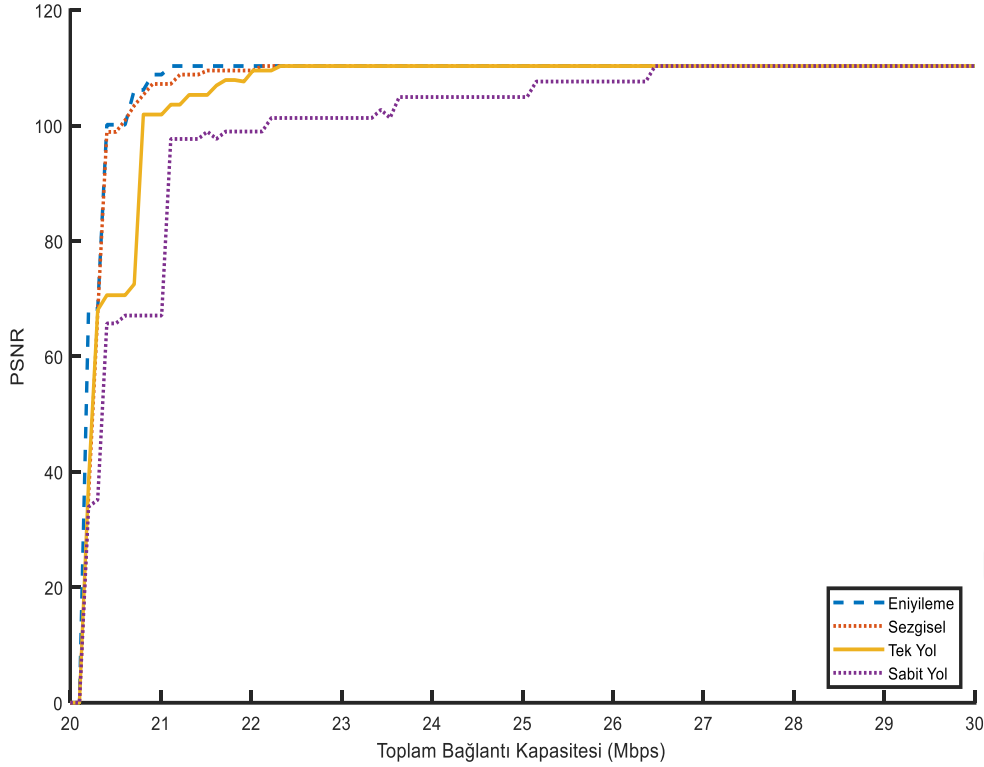
## 4.2 Emülasyon Sonuçları

Şekil 4.2’de Yapılandırma 1 (adaletli dağıtım) için algoritmaların PSNR sonuçları ve o sonuçlara karşılık gelen bağlantı kapasiteleri gösterilmiştir. Sonuçlar benzetim sonuçlarıyla emülasyon sonuçlarının uyduğunu göstermektedir. Sezgisel çok yönlü algoritma, eniyileme algoritmasına yakın sonuç ve kıyaslama algoritmalarından çok daha iyi performans göstermiştir.



Şekil 4.2: Mininet emülasyon sonuçları (Yapılandırma 1)

Şekil 4.3'te Yapılandırma 2 için algoritmaların farklı bağlantı kapasiteleri için karşılaştırma yapılmıştır. Yapılandırma 2 adaletli dağıtım yerine toplam kaliteyi yükseltmeye odaklandığı için merdiven gibi basamaklı bir şekle sahiptir ve kısıtlı kaynaklar için performansta iyileşme görülür. Bu yapılandırmada sezgisel çok yollu algoritma eniyileme algoritmasına yakın sonuç vermektedir. Ayrıca kıyaslama algoritmalarıyla karşılaştırırsak kısıtlı kaynaklar için çok daha iyi sonuç vermektedir.



Şekil 4.3: Mininet emülasyon sonuçları (Yapılandırma 2)

## 5. SONUÇ VE ÖNERİLER

Bu tezde, SDN tabanlı Ölçeklendirilebilir video akışı için eniyilenmiş çok yollu yönlendirme sorunu ele alınmıştır. Katmanlar için eniyilenmiş yönlendirme kurallarını belirlemek için, bant genişliği gereksinimi karşılanmadığında bir video akışının daha küçük parçalara bölünmesini kullanan bir ortak çoklu akış katmanı tahsis problemi formüle edilmiştir. Önerilen çözüm aynı zamanda uçtan uca gecikme ve jitter gereksinimlerini de hesaba katar. Ayrıca, SDN denetleyicisinde verimli bir şekilde kullanılabilen, karmaşıklığı düşük bir sezgisel yöntem önerilmiştir. Sonuçlar, önerilen eniyilenmiş ve sezgisel çözümlerin, benzetim değişkenlerinin tüm değerleri için kalitede önemli iyileştirme sağladığını açıkça göstermektedir. Önerilen algoritmanın iki ana avantajı olduğu görülmektedir. İlk olarak, en yüksek maksimum kaliteye ulaşmak için kendisiyle kıyaslanan algoritmalarından daha az bant genişliğine ihtiyaç duymasıdır. İkinci avantaj, adaletli dağıtım gibi çeşitli senaryolar için kolayca yapılandırılabilmesidir. Ayrıca, gerçek ağın önerilen çözüm ve kıyaslamalar üzerindeki etkilerini araştırmamıza olanak tanıyan Mininet üzerinde emülasyonlar da gerçekleştirilmiştir. Emülasyon sonuçları, önerilen çözümlerin kıyaslama yaklaşımlarından daha iyi performans gösterdiğini ve emülasyon sonuçlarının kuramsal olanlarla uyuştuğunu göstermektedir.

Gelecekteki, daha gerçekçi emülasyon sonuçları almak amacıyla bu çalışma çok yollu yönlendirme tabanlı bir SDN emülasyonu geliştirilerek tekrar edilebilir. Ayrıca bu çalışmada yapılan benzetim ve emülasyonlar, sistem modelini kablosuz bir ağ mimarisi üzerine kurarak ve düğümler arası rastgele paket kaybı gibi gerçek hayatta karşılaşılabilecek olası senaryolar modele dahil edilerek genişletilebilir.



## KAYNAKLAR

- [1] **The World Bank.** (2020). Individuals using the Internet (% of population). Retrieved December 19, 2020, from <https://data.worldbank.org/indicator/IT.NET.USER.ZS?end=2017>
- [2] **Cisco.** (2019). *Cisco visual networking index: Forecast and trends, 2017–2022* (Rep. No. <https://davidellis.ca/wp-content/uploads/2019/05/cisco-vni-feb2019.pdf>). Cisco. doi: <https://davidellis.ca/wp-content/uploads/2019/05/cisco-vni-feb2019.pdf>
- [3] **Schwarz, H., Marpe, D., & 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), 1103-1120. doi:10.1109/tcsvt.2007.905532
- [4] **Cisco.** (2019, July 10). Cisco Nexus 6000 Series NX-OS System Management Configuration Guide, Release 7.x - Configuring OpenFlow [Cisco Nexus 6000 Series Switches]. Retrieved December 19, 2020, from [https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus6000/sw/system\\_management/7x/b\\_6k\\_System\\_Mgmt\\_Config\\_7x/configuring\\_openflow.html](https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus6000/sw/system_management/7x/b_6k_System_Mgmt_Config_7x/configuring_openflow.html)
- [5] **Wu, J., Yuen, C., Cheng, B., Wang, M., & Chen, J.** (2016). Energy-Minimized Multipath Video Transport to Mobile Devices in Heterogeneous Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 34(5), 1160-1178. doi:10.1109/jsac.2016.2551483
- [6] **Wu, J., Cheng, B., Wang, M., & Chen, J.** (2017). Quality-Aware Energy Optimization in Wireless Video Communication With Multipath TCP. *IEEE/ACM Transactions on Networking*, 25(5), 2701-2718. doi:10.1109/tnet.2017.2701153
- [7] **Wu, J., Cheng, B., Wang, M., & Chen, J.** (2018). Energy-Aware Concurrent Multipath Transfer for Real-Time Video Streaming Over Heterogeneous Wireless Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(8), 2007-2023. doi:10.1109/tcsvt.2017.2695368
- [8] **Qadir, J., Ali, A., Yau, K. A., Sathiaselan, A., & Crowcroft, J.** (2015). Exploiting the Power of Multiplicity: A Holistic Survey of Network-Layer

- Multipath. *IEEE Communications Surveys & Tutorials*, 17(4), 2176-2213. doi:10.1109/comst.2015.2453941
- [9] **Nam, H., Calin, D., & Schulzrinne, H.** (2016). Towards dynamic MPTCP Path control using SDN. *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, 286-294. doi:10.1109/netsoft.2016.7502424
- [10] **Rezende, P., Kianpisheh, S., Glitho, R., & Madeira, E.** (2019). An SDN-Based Framework for Routing Multi-Streams Transport Traffic Over Multipath Networks. *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 1-6. doi:10.1109/icc.2019.8762061
- [11] **Cetinkaya, C., & Sayit, M.** (2017). Design of a layer-based video streaming system over software-defined networks. *2017 8th International Conference on the Network of the Future (NOF)*, 8-13. doi:10.1109/nof.2017.8251213
- [12] **Liu, Z.** (2018). SVC video transmission optimization algorithm in software defined network. *China Communications*, 15(10), 143-149. doi:10.1109/cc.2018.8485476
- [13] **Kalan, R. S., Bikas, S., Kheibari, B., & Sayit, M.** (2018). SDN-CPS: Selecting codec, path and server for video streaming over SDN. *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 1-4. doi:10.1109/siu.2018.8404660
- [14] **Hu, H., Zhu, X., Wang, Y., Pan, R., Zhu, J., & Bonomi, F.** (2013). Proxy-Based Multi-Stream Scalable Video Adaptation Over Wireless Networks Using Subjective Quality and Rate Models. *IEEE Transactions on Multimedia*, 15(7), 1638-1652. doi:10.1109/tmm.2013.2266092
- [15] **Quinlan, J. J., Zahran, A. H., Ramakrishnan, K., & Sreenan, C. J.** (2015). Delivery of adaptive bit rate video: Balancing fairness, efficiency and quality. *The 21st IEEE International Workshop on Local and Metropolitan Area Networks*, 1-6. doi:10.1109/lanman.2015.7114736
- [16] **Tashtarian, F., Erfanian, A., & Varasteh, A.** (2018). S2VC: An SDN-based framework for maximizing QoE in SVC-based HTTP adaptive streaming. *Computer Networks*, 146, 33-46. doi: 10.1016/j.comnet.2018.09.007
- [17] **Yue, Y., Ran, Y., Chen, S., Yang, B., Sun, L., & Yang, J.** (2015). Joint Routing and Layer Selecting for Scalable Video Transmission in SDN. *2015 IEEE Globecom Workshops (GC Wkshps)*, 1-6. doi:10.1109/glocomw.2015.7414212
- [18] **Egilmez, H. E., Civanlar, S., & Tekalp, A. M.** (2013). An Optimization Framework for QoS-Enabled Adaptive Video Streaming Over OpenFlow Networks. *IEEE Transactions on Multimedia*, 15(3), 710-715. doi:10.1109/tmm.2012.2232645

- [19] **Ongaro, F., Cerqueira, E., Foschini, L., Corradi, A., & Gerla, M.** (2015). Enhancing the quality level support for real-time multimedia applications in software-defined networks. *2015 International Conference on Computing, Networking and Communications (ICNC)*, 1-7. doi:10.1109/icnc.2015.7069395
- [20] **Grigoriou, E., Barakabitze, A. A., Atzori, L., Sun, L., & Piloni, V.** (2017). An SDN-approach for QoE management of multimedia services using resource allocation. *2017 IEEE International Conference on Communications (ICC)*, 1-7. doi:10.1109/icc.2017.7997261
- [21] **Barakabitze, A. A., Mkwawa, I., Sun, L., & Ifeachor, E.** (2018). QualitySDN: Improving Video Quality using MPTCP and Segment Routing in SDN/NFV. *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 182-186. doi:10.1109/netsoft.2018.8459917
- [22] **Nam, H., Kim, K., Kim, J. Y., & Schulzrinne, H.** (2014). Towards QoE-aware video streaming using SDN. *2014 IEEE Global Communications Conference*, 1317-1322. doi:10.1109/glocom.2014.7036990
- [23] **Yang, J., Yang, B., Chen, S., Zhang, Y., Zhang, Y., & Hanzo, L.** (2019). Dynamic Resource Allocation for Streaming Scalable Videos in SDN-Aided Dense Small-Cell Networks. *IEEE Transactions on Communications*, 67(3), 2114-2129. doi:10.1109/tcomm.2018.2883627
- [24] **Alsaeedi, M., Mohamad, M. M., & Al-Roubaiey, A. A.** (2019). Toward Adaptive and Scalable OpenFlow-SDN Flow Control: A Survey. *IEEE Access*, 7, 107346-107379. doi:10.1109/access.2019.2932422
- [25] **Afzal, S., Testoni, V., Rothenberg, C. E., Kolan, P., & Bouazizi, I.** (n.d.). A holistic survey of wireless multipath video streaming. *ArXiv Preprint*, (ArXiv:1906.06184).
- [26] **Richardson, I. E., & Richardson, I. E.** (2012). *The H.264 advanced video compression standard*. Chichester, West Sussex, UK: Wiley.
- [27] **Floriandjonckheere.** (n.d.). Floriandjonckheere/jsvm. Retrieved December 21, 2020, Github Repository, from <https://github.com/floriandjonckheere/jsvm>
- [28] **Wang, Z., Bovik, A., Sheikh, H., & Simoncelli, E.** (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4), 600-612. doi:10.1109/tip.2003.819861
- [29] **Team, M.** (n.d.). Mininet. Retrieved December 19, 2020, from <http://www.mininet.org/>
- [30] **Ke, C.** (n.d.). CPqD/RouteFlow. Retrieved December 19, 2020, from [https://github.com/CPqD/RouteFlow/blob/master/pox/pox/forwarding/l2\\_multi.py](https://github.com/CPqD/RouteFlow/blob/master/pox/pox/forwarding/l2_multi.py)



- [31] **McCauley, J.** (2015). Installing POX. Retrieved December 21, 2020, Github Repository, from <https://noxrepo.github.io/pox-doc/html/>
- [32] **Ke, C.** (n.d.). Measure the latency. Retrieved December 19, 2020, from <http://csie.nqu.edu.tw/smallko/sdn/latency.htm>
- [33] **Phemius, K., & Bouet, M.** (2013). Monitoring latency with OpenFlow. *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*, 122-125. doi:10.1109/cnsm.2013.6727820

